

Tivoli Storage Manager
for Linux
Version 6.2

Administrator's Guide



Tivoli Storage Manager
for Linux
Version 6.2

Administrator's Guide



Note:

Before using this information and the product it supports, read the information in "Notices" on page 879.

This edition applies to Version 6.2 of IBM Tivoli Storage Manager (product numbers 5608-E01, 5608-E02, 5608-E03, 5608-E07, 5608-E12), and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. This edition replaces SC23-9771-01.

© **Copyright IBM Corporation 1993, 2010.**

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

Contents

Preface	xiii
Who should read this guide	xiii
Publications	xiii
Tivoli Storage Manager publications	xiii
Related hardware publications	xv
Support information	xv
Getting technical training.	xv
Searching knowledge bases	xvi
Contacting IBM Software Support	xvii
Conventions used in this guide	xix

New for IBM Tivoli Storage Manager

Version 6.2.	xxi
New for the server in Version 6.2	xxi
Client-side data deduplication	xxi
Automatic backup-archive client deployment	xxii
Simultaneous-write operations during storage	
pool migration	xxii
In-flight data encryption using SSL	xxii
New for the Tivoli Storage Manager reporting	
and monitoring feature in version 6.2	xxiii
The Tivoli Integrated Portal GUI.	xxiii
The Administration Center not installable on	
HP-UX	xxiii
Sun StorageTek T10000B drive encryption	xxiv
MOVESIZETHRESH server option	xxiv
CHECKTAPEPOS server option to validate	
data position on tape	xxiv

Part 1. Tivoli Storage Manager basics **1**

Chapter 1. Tivoli Storage Manager overview **3**

How client data is stored	5
Data-protection options.	8
Data movement to server storage	14
Consolidation of backed-up client data	14
How the server manages storage	15
Device support	15
Data migration through the storage hierarchy	16
Removal of expired data	16

Chapter 2. Tivoli Storage Manager concepts **19**

Interfaces to Tivoli Storage Manager	19
Server options	20
Storage configuration and management	21
Disk devices	21
Removable media devices	22
Defined volumes and scratch volumes	23
Migrating data from disk to tape	24
Managing storage pools and volumes	24

Management of client operations	25
Managing client nodes.	26
Managing client data with policies	28
Schedules for client operations	30
Server maintenance.	33
Server-operation management	33
Server script automation	34
Database and recovery-log management.	34
Sources of information about the server	35
Tivoli Storage Manager server networks.	35
Exporting and importing data	36
Protecting Tivoli Storage Manager and client data	36
Protecting the server	37

Part 2. Configuring and managing server storage **39**

Chapter 3. Storage device concepts **41**

Road map for key device-related task information	41
Tivoli Storage Manager storage devices	42
Tivoli Storage Manager storage objects	42
Libraries	42
Drives	45
Device class	45
Library, drive, and device-class objects	47
Storage pools and storage-pool volumes.	48
Data movers	49
Paths	50
Server objects.	50
Tivoli Storage Manager volumes	50
Volume inventory for an automated library.	51
Device configurations	52
Devices on local area networks	52
Devices on storage area networks	52
LAN-free data movement.	54
Network-attached storage	55
Mixed device types in libraries	58
Removable media mounts and dismounts	59
How Tivoli Storage Manager uses and reuses	
removable media	60
Required definitions for storage devices	63
Example: Mapping devices to device classes	64
Example: Mapping storage pools to device	
classes and devices	64
Planning for server storage	65
Server options that affect storage operations	66

Chapter 4. Magnetic disk devices **69**

Requirements for disk subsystems.	69
Random access and sequential access disk devices	71
Configuring random access volumes on disk devices	75
Configuring FILE sequential volumes on disk	
devices	76
Varying disk volumes online or offline	77

Cache copies for files stored on disk	77
Freeing space on disk	78
Scratch FILE volumes	78
Volume history file and volume reuse	78

Chapter 5. Using devices with the server system. 79

Attaching a manual drive.	79
Attaching an automated library device	79
Setting the library mode	80
Device driver selection	80
IBM device drivers	81
Tivoli Storage Manager device drivers	81
Installing and configuring device drivers	82
Installing device drivers for IBM SCSI tape devices	82
Installing device drivers for IBM 349x libraries	83
Configuring Tivoli Storage Manager passthru drivers for tape and optical devices and libraries	83
Installing zSeries Linux fibre-channel adapter (zfcp) device drivers	84
Tivoli Storage Manager Support for multipath I/O with IBM tape devices.	85
Device special file names	87
Information about your system's SCSI devices.	87

Chapter 6. Configuring storage devices 89

Device configuration planning	89
Configuring SCSI libraries used by one server.	90
Configuration with a single drive device type	91
Configuration with multiple drive device types	93
Checking in and labeling library volumes	94
Configuring SCSI libraries shared among servers on a SAN	95
Setting up server communications.	96
Set up the device on the server systems and the SAN.	96
Setting up the library manager server	96
Setting up the library client servers	98
Storing client data on devices	99
Migrating DEFINE DRIVE and DEFINE PATH definitions to the library manager server	99
Configuring IBM 3494 libraries	99
Categories in an IBM 3494 library	99
Configuring an IBM 3494 library for use by one server	100
Sharing an IBM 3494 library among servers	105
Migrating a shared IBM 3494 library to a library manager	107
Sharing an IBM 3494 library by static partitioning of drives.	108
ACSLs-Managed libraries	112
Configuring an ACSLS-managed library	112
Configuring an ACSLS library with a single drive device type	112
Configuring an ACSLS library with multiple drive device types	114
Setting up an ACSLS library manager server	115
Setting up an ACSLS library client server	116

Checking in and labeling ACSLS library volumes	117
Removable file device configuration	118
Example of removable file support	118
Labeling requirements for removable file device types	119
Configuration for libraries controlled by media manager programs	120
Setting up Tivoli Storage Manager to work with an external media manager.	120
Externally-controlled IBM Tivoli Storage Manager media.	121
Configuring manually mounted devices	122
Defining devices as part of a manual library	122
Labeling volumes	123
Configuring IBM Tivoli Storage Manager for LAN-free data movement	124
Validating your LAN-free configuration	125
Configuring IBM Tivoli Storage Manager for NDMP operations	125
Defining devices and paths.	126
Defining libraries	126
Defining drives.	127
Defining data movers	128
Defining paths	129
Shared FILE volumes.	130
Impacts of device changes on the SAN	131

Chapter 7. Managing removable media operations. 133

Preparing removable media	133
Labeling removable media volumes	134
Checking new volumes into a library	137
Write-once, read-many tape media	141
Managing the volume inventory	143
Controlling access to volumes	143
Reusing tapes in storage pools	144
Setting up a tape rotation	144
Reusing volumes used for database backups and export operations	146
Maintaining a supply of scratch volumes	146
Managing volumes in automated libraries.	147
Changing the status of a volume	147
Removing volumes from a library	148
Managing a full library	148
Auditing a library's volume inventory	149
Maintaining a supply of scratch volumes in an automated library	150
Operations with shared libraries	150
Managing server requests for media.	151
Using the administrative client for mount messages.	152
Mount operations for manual libraries	152
Messages for automated libraries.	152
Requesting information about pending operator requests	152
Replying to operator requests	153
Canceling an operator request.	153
Responding to requests for volume check-in	153
Determining which volumes are mounted.	154
Dismounting idle volumes	154

Managing libraries	154
Requesting information about libraries	155
Updating libraries	155
Deleting libraries	156
Managing drives	156
Requesting information about drives	156
Updating drives	157
Drive encryption	158
Cleaning drives	159
Deleting drives	163
Managing paths	164
Requesting information about paths	164
Updating paths	164
Deleting paths	164
Managing data movers	165
Requesting information about data movers	165
Updating data movers	165
Deleting data movers	165
Tape alert messages	165

Chapter 8. Using NDMP for operations with NAS file servers 167

NDMP requirements	167
Interfaces for NDMP operations	169
Data formats for NDMP backup operations	170
NDMP operations management	170
Managing NAS file server nodes	170
Managing data movers used in NDMP operations	171
Dedicating a Tivoli Storage Manager drive to NDMP operations	172
Storage pool management for NDMP operations	172
Managing table of contents	172
Configuring Tivoli Storage Manager for NDMP operations	173
Configuring Tivoli Storage Manager policy for NDMP operations	174
Tape libraries and drives for NDMP operations	177
Attaching tape library robotics for NAS-attached libraries	180
Registering NAS nodes with the Tivoli Storage Manager server	185
Defining a data mover for the NAS file server	186
Defining tape drives and paths for NDMP operations	186
Labeling and checking tapes into the library	188
Scheduling NDMP operations	188
Defining virtual file spaces	188
Tape-to-tape copy to back up data	189
Tape-to-tape copy to move data	189
Backing up and restoring NAS file servers using NDMP	189
NAS file servers; backups to a single Tivoli Storage Manager server	190
Performing NDMP filer to Tivoli Storage Manager server backups	191
File-level backup and restore for NDMP operations	192
Interfaces for file-level restore	193
International characters for NetApp file servers	194
File level restore from a directory-level backup image	194

Directory-level backup and restore	195
Directory-level backup and restore for NDMP operations	195
Backing up and restoring with snapshots	195
Backup and restore using NetApp SnapMirror to Tape feature	196
NDMP backup operations using Celerra file server integrated checkpoints	197

Chapter 9. Defining device classes 199

Sequential-access device types	200
Defining tape and optical device classes	201
Specifying the estimated capacity of tape and optical volumes	201
Specifying recording formats for tape and optical media	201
Associating library objects with device classes	202
Controlling media-mount operations for tape and optical devices	202
Write-once, read-many devices	204
Defining 3592 device classes	204
Defining device classes for removable media devices	207
Defining sequential-access disk (FILE) device classes	207
Concurrent access to FILE volumes	207
Mitigating performance degradation when backing up or archiving to FILE volumes	208
Specifying directories in FILE device-class definitions	208
Controlling the size of FILE volumes	210
Controlling the number of concurrently open FILE volumes	210
Defining LTO device classes	211
Mixing LTO drives and media in a library	211
Mount limits in LTO mixed-media environments	212
Encrypting data using LTO generation 4 tape drives	213
Defining SERVER device classes	214
Controlling the size of files created on a target server	214
Controlling the number of simultaneous sessions between source and target servers	214
Controlling the amount of time a SERVER volume remains mounted	215
Defining device classes for StorageTek VolSafe devices	215
Enabling ECARTRIDGE drive encryption	216
Disabling ECARTRIDGE drive encryption	216
Obtaining information about device classes	216
How Tivoli Storage Manager fills volumes	217
Data compression	218
Tape volume capacity and data compression	218

Chapter 10. Managing storage pools and volumes. 221

Storage pools	222
Primary storage pools	222
Copy storage pools	223
Active-data pools	223

Example: Setting up server storage	225	Collocation of copy storage pools and	
Defining storage pools	227	active-data pools	316
Task tips for storage pools	232	Planning for and enabling collocation	316
Storage pool volumes	234	Reclaiming space in sequential-access storage pools	319
Random-access storage pool volumes	234	How Tivoli Storage Manager reclamation works	319
Sequential-access storage pool volumes.	234	Reclamation thresholds	321
Preparing volumes for random-access storage		Reclaiming volumes with the most reclaimable	
pools	236	space	321
Preparing volumes for sequential-access storage		Starting reclamation manually or in a schedule	322
pools	236	Optimizing drive usage using multiple	
Updating storage pool volumes	238	concurrent reclamation processes	322
Access modes for storage pool volumes	239	Reclaiming volumes in a storage pool with one	
Storage pool hierarchies	241	drive	323
Setting up a storage pool hierarchy	241	Reducing the time to reclaim tape volumes with	
How the server groups files before storing	243	high capacity	324
Where the server stores files	244	Controlling reclamation of virtual volumes	324
Example: How the server determines where to		Reclaiming copy storage pools and active-data	
store files in a hierarchy.	244	pools	325
Backing up the data in a storage hierarchy	246	How collocation affects reclamation	329
Staging client data from disk to tape	251	Estimating space needs for storage pools	330
Migrating files in a storage pool hierarchy.	252	Estimating space requirements in random-access	
Migrating disk storage pools	253	storage pools	330
Migrating sequential-access storage pools	258	Estimating space needs in sequential-access	
The effect of migration on copy storage pools		storage pools	332
and active-data pools.	263	Monitoring storage-pool and volume usage	332
Caching in disk storage pools	263	Monitoring space available in a storage pool	332
How the server removes cached files	264	Monitoring the use of storage pool volumes	335
Effect of caching on storage pool statistics.	264	Monitoring migration processes	342
Deduplicating data	264	Monitoring the use of cache space on disk	
Data deduplication location	264	storage	344
Data deduplication limitations.	269	Obtaining information about the use of storage	
Planning guidelines for data deduplication	270	space	345
Detecting possible security attacks on the server		Moving data from one volume to another volume	349
during client-side deduplication	271	Data movement within the same storage pool	350
Evaluating data deduplication in a test		Data movement to a different storage pool	350
environment.	272	Data movement from off-site volumes in copy	
Managing deduplication-enabled storage pools	274	storage pools or active-data pools	351
Controlling data deduplication	276	Moving data	351
Displaying statistics about server-side data		Moving data belonging to a client node	354
deduplication	283	Moving data in all file spaces belonging to one	
Improving performance when reading from		or more nodes	354
deduplicated storage pools	284	Moving data in selected file spaces belonging to	
Writing data simultaneously to primary, copy, and		a single node	355
active-data pools	285	Obtaining information about data-movement	
Guidelines for using the simultaneous-write		processes	356
function	286	Troubleshooting incomplete data-movement	
Limitations that apply to the simultaneous-write		operations	356
function	287	Renaming storage pools	357
Controlling the simultaneous-write function	288	Defining copy storage pools and active-data pools	357
Simultaneous-write operations: Examples	291	Example: Defining a copy storage pool	359
Planning simultaneous-write operations	305	Properties of primary, copy, and active-data	
Simultaneous-write function as part of a backup		pools	359
strategy: Example	309	Deleting storage pools	361
Keeping client files together using collocation	310	Deleting storage pool volumes	361
The effects of collocation on operations.	311	Deleting empty storage pool volumes	362
How the server selects volumes with collocation		Deleting storage pool volumes that contain data	362
enabled	313		
How the server selects volumes with collocation			
disabled	315		
Collocation on or off settings	315		

Part 3. Managing client operations 365

Chapter 11. Adding client nodes . . . 367

Overview of clients and servers as nodes	367
Installing client node software	368
Registering nodes with the server	368
Accepting default closed registration or enabling open registration	368
Registering nodes with client options sets	370
Registering a network-attached storage file server as a node	370
Registering a source server as a node on a target server	371
Registering an API to the server	371
Connecting nodes with the server	372
Required client options	372
UNIX and Linux client options	373
Creating or updating a client options file	373
Using a text editor to create or configure a client options file	373
Using the client configuration wizard to create or update a client options file	373
Comparing network-attached nodes to local nodes	374
Adding clients through the administrative command line client	375
Enabling open registration	375
Example: registering three client nodes using the administrative command line	375
Chapter 12. Managing client nodes	377
Managing nodes	377
Managing client nodes across a firewall	378
Updating client node information	380
Automatically updating backup-archive clients using the Administration Center	380
Command-line backup-archive client automatic deployment feature: overview	384
Renaming client nodes	388
Locking and unlocking client nodes	388
Deleting client nodes	389
Consolidating multiple clients under a single client node name	389
Displaying information about client nodes	392
Overview of remote access to web backup-archive clients	393
Managing client access authority levels	395
Managing file spaces	397
Defining client nodes and file spaces	398
Supporting Unicode-enabled clients	400
Displaying information about file spaces	409
Moving data for a client node	410
Deleting file spaces	410
Managing client option files	411
Creating client option sets on the server	411
Managing client option sets	413
Managing IBM Tivoli Storage Manager sessions	414
Displaying information about IBM Tivoli Storage Manager sessions	414
Canceling an IBM Tivoli Storage Manager session	415
When a client session is automatically canceled	416
Disabling or enabling access to the server	417
Managing client restartable restore sessions	417

Chapter 13. Managing IBM Tivoli Storage Manager security	421
Securing client and server communications	421
Setting up SSL	421
Securing the server console	425
Administrative authority and privilege classes	425
Managing access to the server and clients	427
Managing IBM Tivoli Storage Manager administrators	428
Registering administrators	428
Updating information about other administrators	428
Renaming an administrator	429
Removing administrators	429
Displaying information about administrators	429
Locking and unlocking administrators from the server	430
Managing levels of administrative authority	430
Granting authority to administrators	431
Extending authority for administrators	431
Reducing authority for administrators	431
Reducing privilege classes	432
Revoking authority for administrators	432
Managing passwords and login procedures	432
Modifying the default password expiration period	432
Setting a limit for invalid password attempts	433
Setting a minimum length for a password	433
Disabling the default password authentication	433

Chapter 14. Implementing policies for client data	435
Basic policy planning	436
Reviewing the standard policy	437
Getting users started	438
Changing policy	439
File expiration and expiration processing	439
Client operations controlled by policy	440
Backup and restore	440
Archive and retrieve	441
Client migration and recall	441
The parts of a policy	443
Relationships among clients, storage, and policy	444
More on management classes	446
Contents of a management class	446
Default management classes	447
The include-exclude list	448
How files and directories are associated with a management class	449
How Tivoli Storage Manager selects files for policy operations	451
Incremental backup	452
Selective backup	454
Logical volume backup	454
Archive	455
Automatic migration from a client node	455
How client migration works with backup and archive	456
Creating your own policies	456
Example: sample policy objects	457

Defining and updating a policy domain	458
Defining and updating a policy set	460
Defining and updating a management class	461
Defining and updating a backup copy group	462
Defining and updating an archive copy group	468
Assigning a default management class	470
Validating and activating a policy set	470
Assigning client nodes to a policy domain.	472
Running expiration processing to delete expired files	472
Running expiration processing automatically	473
Using commands and scheduling to control expiration processing	473
Additional expiration processing with disaster recovery manager	473
Protection and expiration of archive data	474
Data retention protection	474
Deletion hold	475
Protecting data using the NetApp SnapLock licensed feature.	476
Reclamation and the SnapLock feature	477
Set up SnapLock volumes as Tivoli Storage Manager WORM FILE volumes	481
Policy configuration scenarios	482
Configuring policy for direct-to-tape backups	482
Configuring policy for Tivoli Storage Manager application clients	483
Policy for logical volume backups	483
Configuring policy for NDMP operations	485
Configuring policy for LAN-free data movement	486
Policy for Tivoli Storage Manager servers as clients	488
Setting policy to enable point-in-time restore for clients	488
Distributing policy using enterprise configuration	489
Querying policy	489
Querying copy groups	490
Querying management classes.	490
Querying policy sets	491
Querying policy domains	491
Deleting policy	492
Deleting copy groups.	492
Deleting management classes	493
Deleting policy sets	493
Deleting policy domains.	493
Chapter 15. Managing data for client nodes.	495
Validating a node's data	495
Performance considerations for data validation	496
Validating a node's data during a client session	496
Encrypting data on tape	496
Choosing an encryption method	497
Changing your encryption method and hardware configuration	498
Securing sensitive client data	499
Setting up shredding	500
Ensuring that shredding is enforced	501
Creating and using client backup sets	502
Generating client backup sets on the server	503

Restoring backup sets from a backup-archive client	507
Moving backup sets to other servers.	508
Managing client backup sets	509
Enabling clients to use subfile backup	511
Setting up clients to use subfile backup.	512
Managing subfile backups	513
Optimizing restore operations for clients	514
Environment considerations	514
Restoring entire file systems	515
Restoring parts of file systems.	515
Restoring databases for applications.	516
Restoring files to a point-in-time	517
Concepts for client restore operations	517
Managing archive data	520
Archive operations overview	520
Managing storage usage for archives	521

Chapter 16. Scheduling operations for client nodes 525

Prerequisites to scheduling operations	525
Scheduling a client operation	526
Defining client schedules	526
Associating client nodes with schedules	527
Starting the scheduler on the clients.	527
Displaying schedule information	528
Checking the status of scheduled operations	528
Creating schedules for running command files	529
Updating the client options file to automatically generate a new password	530

Chapter 17. Managing schedules for client nodes 531

Managing IBM Tivoli Storage Manager schedules	531
Adding new schedules	531
Copying existing schedules.	532
Modifying schedules	532
Deleting schedules	532
Displaying information about schedules	532
Managing node associations with schedules	533
Adding new nodes to existing schedules	533
Moving nodes from one schedule to another	534
Displaying nodes associated with schedules	534
Removing nodes from schedules	534
Managing event records	534
Displaying information about scheduled events	535
Managing event records in the server database	536
Managing the throughput of scheduled operations	537
Modifying the default scheduling mode	537
Specifying the schedule period for incremental backup operations.	539
Balancing the scheduled workload for the server	539
Controlling how often client nodes contact the server	541
Specifying one-time actions for client nodes	543
Determining how long the one-time schedule remains active	544

Part 4. Maintaining the server . . . 545

Chapter 18. Managing servers with the Administration Center.	547	Using substitution variables in a macro.	589
Using the Administration Center	547	Running a macro	590
Starting and stopping the Administration Center	550	Command processing in a macro.	590
Functions not in the Administration Center	550	Chapter 21. Managing the database and recovery log	593
Protecting the Administration Center	552	Database and recovery log overview	593
Backing up the Administration Center	553	Database	594
Restoring the Administration Center.	553	Recovery log	595
Chapter 19. Managing server operations.	555	Disk space requirements for the server database and recovery log	599
Licensing IBM Tivoli Storage Manager	555	Estimating database space requirements	599
Registering licensed features	556	Estimating recovery log space requirements	601
Monitoring licenses	557	Active log space	601
Starting the Tivoli Storage Manager server	557	Active log mirror space	601
Starting the server on AIX, Linux, and UNIX	558	Archive log space	602
Automating server startup	558	Archive failover log space	602
Stand-alone mode for server startup.	559	Monitoring the database and recovery log.	603
Starting the server in other modes	560	Increasing the size of the database	604
Running multiple server instances on a single system	560	Reducing the size of the database	604
Halting the server	561	Increasing the size of the active log	605
Stopping the server when running as a background process	562	Backing up the database.	605
Moving the Tivoli Storage Manager server to another system.	562	Preparing the system for database backups	606
Date and time on the server	563	Scheduling database backups	606
Managing server processes	564	Backing up the database manually	607
Requesting information about server processes	565	Restoring the database	607
Canceling server processes	565	Moving the database and recovery log on a server	607
Preemption of client or server operations	566	Moving both the database and recovery log	608
Setting the server name	567	Moving only the database	608
Changing the host name for a Tivoli Storage Manager server.	568	Moving only the active log	609
Adding or updating server options	568	Moving only the archive log	609
Adding or updating a server option without restarting the server	569	Moving only the archive failover log	609
Getting help on commands and error messages	569	Adding optional logs after server initialization	610
Chapter 20. Automating server operations.	571	Transaction processing	610
Automating a basic administrative command schedule	572	Files moved as a group between client and server	610
Defining the schedule	572	Chapter 22. Monitoring the Tivoli Storage Manager server.	613
Verifying the schedule	573	Using IBM Tivoli Storage Manager queries to display information	613
Tailoring schedules	573	Requesting information about IBM Tivoli Storage Manager definitions	613
Using classic and enhanced command schedules	575	Requesting information about client sessions	614
Copying schedules	576	Requesting information about server processes	615
Deleting schedules	576	Requesting information about server settings	616
Managing scheduled event records	576	Querying server options.	616
Querying events	577	Querying the system	617
Removing event records from the database	577	Using SQL to query the IBM Tivoli Storage Manager database.	618
IBM Tivoli Storage Manager server scripts.	578	Using SELECT commands	618
Defining a server script	578	Using SELECT commands in IBM Tivoli Storage Manager scripts	621
Managing server scripts	584	Querying the SQL activity summary table	622
Running a server script	586	Creating output for use by another application	623
Using macros	587	Using the IBM Tivoli Storage Manager activity log	623
Writing commands in a macro.	588	Requesting information from the activity log	624
Writing comments in a macro	588	Setting a retention period for the activity log	625
Using continuation characters	589	Setting a size limit for the activity log	625

Logging IBM Tivoli Storage Manager events to receivers	626
Enabling and disabling events	627
Beginning and ending event logging	628
Logging events to the IBM Tivoli Storage Manager server console and activity log	628
Logging events to a file exit and a user exit	629
Logging events to the Tivoli Enterprise Console	630
Logging events to an SNMP manager	634
Enterprise event logging: logging events to another server	639
Querying event logging	640
User exit and file exit receivers	641
Monitoring IBM Tivoli Storage Manager accounting records	645
Daily monitoring scenario	646
Reporting and monitoring servers and clients	647
Client activity reports	650
Server trend reports	654
Monitoring workspaces	657
Running the Tivoli Storage Manager client and server reports	667
Monitoring Tivoli Storage Manager real-time data	668
Modifying the IBM Tivoli Monitoring environment file	669

Chapter 23. Managing a network of Tivoli Storage Manager servers 671

Concepts for managing server networks	671
Enterprise configuration	672
Command routing	673
Central monitoring for the Tivoli Storage Manager server	673
Data storage on another server	674
Examples: management of multiple Tivoli Storage Manager servers	674
Enterprise-administration planning	676
Setting up communications among servers	676
Setting up communications for enterprise configuration and enterprise event logging	676
Setting up communications for command routing	680
Updating and deleting servers	684
Setting up enterprise configurations	685
Enterprise configuration scenario	685
Creating the default profile on a configuration manager	689
Creating and changing configuration profiles	690
Getting information about profiles	697
Subscribing to a profile	699
Refreshing configuration information	703
Managing problems with configuration refresh	704
Returning managed objects to local control	704
Setting up administrators for the servers	705
Managing problems with synchronization of profiles	705
Switching a managed server to a different configuration manager	706
Deleting subscribers from a configuration manager	706
Renaming a managed server	706

Performing tasks on multiple servers	707
Working with multiple servers using the Administration Center	707
Routing commands	707
Setting up server groups	710
Querying server availability	712
Using virtual volumes to store data on another server	712
Setting up source and target servers for virtual volumes	714
Performance limitations for virtual volume operations	715
Performing operations at the source server	716
Reconciling virtual volumes and archive files	718

Chapter 24. Exporting and importing data 721

Reviewing data that can be exported and imported	721
Exporting restrictions	722
Deciding what information to export	722
Deciding when to export	723
Exporting data directly to another server	724
Options to consider before exporting	724
Preparing to export to another server for immediate import	728
Monitoring the server-to-server export process	730
Exporting administrator information to another server	730
Exporting client node information to another server	731
Exporting policy information to another server	732
Exporting server data to another server	732
Exporting and importing data using sequential media volumes	732
Using preview before exporting or importing data	732
Planning for sequential media used to export data	733
Exporting tasks	734
Importing data from sequential media volumes	737
Monitoring export and import processes	748
Exporting and importing data from virtual volumes	751

Part 5. Protecting the server 753

Chapter 25. Protecting and recovering your server 755

Levels of protection	755
Storage pool protection overview	756
Storage pool restore processing	757
Marking volumes as destroyed	757
Database and recovery log protection overview	758
Types of server database restores	759
Active log mirroring	759
Snapshot database backup	760
Secure Sockets Layer digital certificate file protection	760
Backing up storage pools	760
Scheduling storage pool backups	764

Scenario: scheduling a backup with one copy storage pool	764
Simultaneous-write operations to copy storage pools and active-data storage pools	765
Using multiple copy storage pools and active-data pools	765
Delaying reuse of volumes for recovery purposes	766
Backing up the server database	767
Defining device classes for backups	767
Estimating the size of the active log	767
Scheduling database backups	768
Saving the volume history file	768
Saving the device configuration file	770
Saving the server options and database and recovery log information	771
Running full and incremental backups	771
Running snapshot database backups	772
Recovering the server using database and storage pool backups	772
Restoring a server database to a point in time	774
Restoring a server database to its most current state	776
Updating the device configuration file	776
Restoring storage pools	777
Restoring storage pool volumes	780
Volume restoration	781
Fixing an incomplete volume restoration	781
Auditing storage pool volumes	782
Storage pool volume audit	783
Data validation during audit volume processing	784
Auditing a disk storage pool volume	788
Auditing multiple volumes in a sequential access storage pool	789
Auditing a single volume in a sequential access storage pool	789
Auditing volumes by date written	790
Auditing volumes in a specific storage pool	790
Scheduling volume audits	790
Fixing damaged files	790
Ensuring the integrity of files	791
Restoring damaged files	791
Backup and recovery scenarios	792
Protecting the database and storage pools	792
Recovering to a point-in-time from a disaster	794
Recovering a lost or damaged storage pool volume	796
Restoring a library manager database	797
Restoring a library client database	798

Chapter 26. Using disaster recovery manager 801

Querying defaults for the disaster recovery plan file	801
Specifying defaults for the disaster recovery plan file	802
Specifying defaults for offsite recovery media management	805
Specifying recovery instructions for your site	807
Specifying information about your server and client node machines	808

Specifying recovery media for client machines	810
Creating and storing the disaster recovery plan	810
Storing the disaster recovery plan locally	811
Storing the disaster recovery plan on a target server	812
Managing disaster recovery plan files stored on target servers	812
Displaying information about recovery plan files	812
Displaying the contents of a recovery plan file	813
Restoring a recovery plan file	813
Expiring recovery plan files automatically	813
Deleting recovery plan files manually	814
Moving backup media	814
Moving copy storage pool and active-data pool volumes off-site	816
Moving copy storage pool and active-data pool volumes on-site	817
Summary of disaster recovery manager daily tasks	819
Staying prepared for a disaster	821
Recovering from a disaster	822
Server recovery scenario	823
Client recovery scenario	826
Recovering with different hardware at the recovery site	828
Automated SCSI library at the original and recovery sites	828
Automated SCSI library at the original site and a manual scsi library at the recovery site	829
Managing copy storage pool volumes and active-data pool volumes at the recovery site	830
Disaster recovery manager checklist	831
The disaster recovery plan file	836
Breaking out a disaster recovery plan file	836
Structure of the disaster recovery plan file	836
Example disaster recovery plan file	838

Part 6. Appendixes 857

Appendix A. External media management interface description . . . 859

CreateProcess call	859
Processing during server initialization	860
Processing for mount requests	860
Processing for release requests	861
Processing for batch requests	861
Error handling	862
Begin batch request	862
End batch request	862
Volume query request	863
Initialization requests	864
Volume eject request	864
Volume release request	865
Volume mount request	866
Volume dismount request	869

Appendix B. User exit and file exit receivers 871

Sample user exit declarations	871
Sample user-exit program	873
Readable text file exit (FILETEXTXIT) format	874

**Appendix C. Accessibility features for
Tivoli Storage Manager 877**

Notices 879

Trademarks 881

Glossary 883

Index 905

Preface

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

Who should read this guide

This guide is intended for anyone who is registered as an administrator for Tivoli Storage Manager. 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

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

You can search all publications in the Tivoli Storage Manager Information Center: <http://publib.boulder.ibm.com/infocenter/tsminfo/v6r2>.

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

Go to Tivoli Documentation Central to find information centers that contain official product documentation for current and previous versions of Tivoli products, including Tivoli Storage Manager products at <http://www.ibm.com/developerworks/wikis/display/tivolidoccentral/Tivoli+Storage+Manager>.

You can also order some related publications from the IBM Publications Center Web site. The Web site provides information about ordering publications from countries other than the United States. In the United States, you can order publications by calling 1-800-879-2755.

Tivoli Storage Manager publications

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

Table 1. IBM Tivoli Storage Manager troubleshooting and tuning publications

Publication title	Order number
<i>IBM Tivoli Storage Manager Client Messages and Application Programming Interface Return Codes</i>	SC27-2877
<i>IBM Tivoli Storage Manager Server Messages and Error Codes</i>	SC27-2878

Table 1. IBM Tivoli Storage Manager troubleshooting and tuning publications (continued)

Publication title	Order number
<i>IBM Tivoli Storage Manager Performance Tuning Guide</i>	GC23-9788
<i>IBM Tivoli Storage Manager Problem Determination Guide</i>	GC23-9789

Table 2. Tivoli Storage Manager server publications

Publication title	Order number
<i>IBM Tivoli Storage Manager for AIX Installation Guide</i>	GC23-9781
<i>IBM Tivoli Storage Manager for AIX Administrator's Guide</i>	SC23-9769
<i>IBM Tivoli Storage Manager for AIX Administrator's Reference</i>	SC23-9775
<i>IBM Tivoli Storage Manager for HP-UX Installation Guide</i>	GC23-9782
<i>IBM Tivoli Storage Manager for HP-UX Administrator's Guide</i>	SC23-9770
<i>IBM Tivoli Storage Manager for HP-UX Administrator's Reference</i>	SC23-9776
<i>IBM Tivoli Storage Manager for Linux Installation Guide</i>	GC23-9783
<i>IBM Tivoli Storage Manager for Linux Administrator's Guide</i>	SC23-9771
<i>IBM Tivoli Storage Manager for Linux Administrator's Reference</i>	SC23-9777
<i>IBM Tivoli Storage Manager for Sun Solaris Installation Guide</i>	GC23-9784
<i>IBM Tivoli Storage Manager for Sun Solaris Administrator's Guide</i>	SC23-9772
<i>IBM Tivoli Storage Manager for Sun Solaris Administrator's Reference</i>	SC23-9778
<i>IBM Tivoli Storage Manager for Windows Installation Guide</i>	GC23-9785
<i>IBM Tivoli Storage Manager for Windows Administrator's Guide</i>	SC23-9773
<i>IBM Tivoli Storage Manager for Windows Administrator's Reference</i>	SC23-9779
<i>IBM Tivoli Storage Manager Server Upgrade Guide</i>	SC23-9554
<i>IBM Tivoli Storage Manager Integration Guide for Tivoli Storage Manager FastBack</i>	SC27-2828

Table 3. Tivoli Storage Manager storage agent publications

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

Table 4. Tivoli Storage Manager client publications

Publication title	Order number
<i>IBM Tivoli Storage Manager for UNIX and Linux: Backup-Archive Clients Installation and User's Guide</i>	SC23-9791
<i>IBM Tivoli Storage Manager for Windows: Backup-Archive Clients Installation and User's Guide</i>	SC23-9792

Table 4. Tivoli Storage Manager client publications (continued)

Publication title	Order number
IBM Tivoli Storage Manager for Space Management for UNIX and Linux: User's Guide	SC23-9794
IBM Tivoli Storage Manager Using the Application Programming Interface	SC23-9793

Table 5. Tivoli Storage Manager Data Protection publications

Publication title	Order number
IBM Tivoli Storage Manager for Enterprise Resource Planning: Data Protection for SAP Installation and User's Guide for DB2	SC33-6341
IBM Tivoli Storage Manager for Enterprise Resource Planning: Data Protection for SAP Installation and User's Guide for Oracle	SC33-6340

Related hardware publications

The following table lists related IBM hardware products publications.

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

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

Support information

You can find support information for IBM products from various sources.

Start at the IBM Support Portal: <http://www.ibm.com/support/entry/portal/>. You can select the products that you are interested in, and search for a wide variety of relevant information.

Getting technical training

Information about Tivoli technical training courses is available online.

Go to these Web sites for training information:

Tivoli software training and certification

Choose from instructor led, online classroom training, self-paced Web classes, Tivoli certification preparation, and other training options at this site: <http://www.ibm.com/software/tivoli/education/>

Tivoli Support Technical Exchange

Technical experts share their knowledge and answer your questions in these webcasts: http://www.ibm.com/software/sysmgmt/products/support/supp_tech_exch.html

Searching knowledge bases

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

Begin by searching the Tivoli Storage Manager Information Center at <http://publib.boulder.ibm.com/infocenter/tsminfo/v6r2>. From this Web site, you can search the current Tivoli Storage Manager documentation.

Searching the Internet

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

To search multiple Internet resources, go to the support Web site for Tivoli Storage Manager at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

You can search for information without signing in. Sign in using your IBM ID and password, if you want to customize the site based on your product usage and information needs. If you do not already have an IBM ID and password, click **Sign in** at the top of the page and follow the instructions to register.

From the Support Web site, you can search various resources including:

- IBM technotes
- IBM downloads
- IBM Redbooks® publications
- IBM Authorized Program Analysis Reports (APARs)

Select the product and click **Downloads** to search the APAR list.

If you still cannot find a solution to the problem, you can search forums and newsgroups on the Internet for the latest information that might help you resolve your problem.

An independent user discussion list, ADSM-L, is hosted by Marist College. You can subscribe by sending an e-mail to listserv@vm.marist.edu. The body of the message must contain the following text: `SUBSCRIBE ADSM-L your_first_name your_family_name`.

To share your experiences and learn from others in the Tivoli Storage Manager user community, go to the Tivoli Storage Manager wiki at <http://www.ibm.com/developerworks/wikis/display/tivolistoragemanager>.

Using IBM Support Assistant

IBM Support Assistant is a complimentary software product that helps you with problem determination. You can install the stand-alone IBM Support Assistant application on any workstation. You can then enhance the application by installing product-specific plug-in modules for the IBM products that you use.

IBM Support Assistant helps you gather support information when you need to open a problem management record (PMR), which you can then use to track the problem. For more information, see the IBM Support Assistant Web site at <http://www.ibm.com/software/support/isa/>.

The product-specific plug-in modules provide you with the following resources:

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

Find add-ons for specific products here: <http://www.ibm.com/support/docview.wss?&uid=swg27012689>.

Finding product fixes

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

You can determine what fixes are available by checking the IBM Software Support Web site at <http://www.ibm.com/support/entry/portal/>.

- If you previously customized the site based on your product usage:
 1. Click the link for your Tivoli Storage Manager product, or one of the other Tivoli Storage Manager components that you want to find a fix for.
 2. Click **Downloads**, and then click **Fixes by version**.
- If you have not customized the site based on your product usage, click **Downloads** and search for your product.

Receiving notification of product fixes

You can receive notifications about fixes, flashes, upgrades, and other news about IBM products.

To sign up to receive notifications about IBM products, follow these steps:

1. From the support page at <http://www.ibm.com/support/entry/portal/>, click **My notifications** in the notifications module.
2. Sign in using your IBM ID and password. If you do not have an ID and password, click **register now** above the IBM ID and password.
3. Click the **Subscribe** tab to select your product family and click **Continue**.
4. Select the type of information that you want to receive, and add your personal preferences. You can specify how you want to be notified, how often, and you can also optionally select a folder for the notifications.
5. Click **Submit**.
6. For notifications for other products, repeat steps 4 and 5.

Tip: You can also pick a product first, from the main support portal site, and then click in the **Notifications** section to create or update your subscription for that product.

Contacting IBM Software Support

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

Before you contact IBM Software Support, follow these steps:

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

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

Setting up a subscription and support contract

Set up a subscription and support contract. The type of contract that you need depends on the type of product you have.

For IBM distributed software products (including, but not limited to, IBM Tivoli, Lotus®, and Rational® products, as well as IBM DB2® and IBM WebSphere® products that run on Microsoft® Windows® or UNIX® operating systems), enroll in IBM Passport Advantage® in one of the following ways:

- **Online:** Go to the Passport Advantage Web page at <http://www.ibm.com/software/lotus/passportadvantage/>, click **How to enroll**, and follow the instructions.
- **By Phone:** You can call 1-800-IBMSERV (1-800-426-7378) in the United States, or for the phone number to call in your country, go to the IBM Software Support Handbook Web page at <http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html> and click **Contacts**.

Determining the business impact

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

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

Describing the problem and gather background information

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

To save time, know the answers to these questions:

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

Submitting the problem to IBM Software Support

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

Online

Go to the IBM Software Support Web site at [http://www.ibm.com/support/entry/portal/Open_service_request/Software/Software_support_\(general\)](http://www.ibm.com/support/entry/portal/Open_service_request/Software/Software_support_(general)). Sign in to access IBM Service Requests, and enter your information into the problem submission tool.

By phone

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

Conventions used in this guide

- Command to be entered on the Linux command line:
`> dsmadm`
- Command to be entered on the command line of an administrative client:
`query devclass`

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

New for IBM Tivoli Storage Manager Version 6.2

Many features in the Tivoli Storage Manager Version 6.2 server are new for previous Tivoli Storage Manager users.

New for the server in Version 6.2

Tivoli Storage Manager server Version 6.2 contains many new features and changes. Any updates that have been made to the information since the previous edition are marked with a vertical bar (|) in the left margin.

| **Client-side data deduplication**

| In client-side data deduplication, the Tivoli Storage Manager backup-archive client
| and the server work together to identify duplicate data.

| Data deduplication is a method of reducing storage needs by eliminating
| redundant data. In Tivoli Storage Manager V6.1, only the server could identify and
| remove redundant data. In V6.2, you have the option of identifying and removing
| redundant data during backup and archive processing before data is sent to the
| server. This method of data deduplication is called *client-side data deduplication*. It is
| available with V6.2 backup-archive clients and the V6.2 Tivoli Storage Manager
| application programming interface (API).

| Client-side data deduplication provides several advantages to server-side data
| deduplication. Client-side data deduplication reduces the amount of data sent over
| the local area network (LAN). In addition, the processing power that is required to
| identify duplicate data is offloaded from the server to client nodes. The processing
| that is required to remove duplicate data on the server is eliminated. Space savings
| occur immediately.

| If you used server-side data deduplication, V6.2 client nodes can access existing
| deduplicated data and storage pools that are already set up for data deduplication.
| When restoring or retrieving files, the client node queries for and displays files as
| it normally does. If a user selects a file that exists in a deduplicated storage pool,
| the server manages the work of reconstructing the file.

| You enable client-side data deduplication using a combination of settings on the
| client node and the server. The primary storage pool that is specified by the copy
| group of the management class associated with the client data must be a
| sequential-access disk (FILE) storage pool that is enabled for data deduplication.

| **Related tasks**

| “Deduplicating data” on page 264

Automatic backup-archive client deployment

IBM Tivoli Storage Manager V6.2 can deploy backup-archive client code to workstations that already have the backup-archive client installed.

You can now deploy backup-archive client code to candidate client workstations from the Tivoli Storage Manager V6.2 Administration Center. From the Administration Center, you can coordinate the client updates to each workstation that is at release 5.4 and later to V6.2. You are helped through the process by wizards that configure your workstation and schedule the deployments. The backup-archive client deployment feature is available for Windows backup-archive clients only.

Related concepts

“Backup-archive client automatic deployment feature: Overview” on page 381

Simultaneous-write operations during storage pool migration

With Tivoli Storage Manager, you can now write data simultaneously to copy storage pools and active-data pools during server data-migration processes.

The simultaneous-write function during migration can reduce the amount of time required to back up storage pools or copy active data. Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied again to the copy storage pools or active-data pools. For example, suppose that you migrate all the data in your primary random-access disk storage pool nightly and then back up your primary storage pools. By using the simultaneous-write function during migration, you can significantly reduce the amount of time required for backup operations.

You can also use the simultaneous-write function during migration if you have many client nodes and the number of mount points that are required to perform the simultaneous-write function during client store operations is unacceptable. If mounting and demounting tapes when writing data simultaneously during client store operations is taking too much time, consider writing data simultaneously during migration.

With Tivoli Storage Manager V6.2, you can specify the simultaneous-write function for a primary storage pool if it is the target for *any* of the eligible operations (client store sessions, server import processes, and server data-migration processes).

Related tasks

“Writing data simultaneously to primary, copy, and active-data pools” on page 285

In-flight data encryption using SSL

Support for Secure Sockets Layer (SSL) is available on HP-UX, Linux[®], Solaris, AIX[®], and Windows platforms.

With SSL industry-standard communications, you can encrypt all traffic between the backup-archive client, the administrative command-line clients, and the IBM Tivoli Storage Manager server. You can use either self-signed or vendor-acquired SSL certificates.

New for the Tivoli Storage Manager reporting and monitoring feature in version 6.2

The Tivoli Storage Manager reporting and monitoring feature, Version 6.2 has a few new changes.

The Tivoli Storage Manager reporting and monitoring feature, Version 6.2, has been integrated into a new user interface called the Tivoli Integrated Portal. This move affects the reporting and monitoring reports that are run from the Administration Center. The Administration Center moved from the Integrated Solutions Console to the Tivoli Integrated Portal. The Tivoli Integrated Portal provides all the functions that were available in the Integrated Solutions Console, but with a new look-and-feel.

The Administration Center is installed separately and is not included in the reporting and monitoring installation.

There is a new information roadmap for the Tivoli Storage Manager reporting and monitoring feature on the Tivoli Storage Manager Wiki. This roadmap has detailed information on planning, installing, configuring, customizing, and trouble shooting. Reporting and monitoring feature information roadmap

Related concepts

Chapter 18, “Managing servers with the Administration Center,” on page 547

The Tivoli Integrated Portal GUI

The IBM Tivoli Integrated Portal is a graphical user interface (GUI) that is included with Tivoli Storage Manager V6.2. The Tivoli Integrated Portal provides all the functions that were available in the Integrated Solutions Console.

The Administration Center, Tivoli Storage Manager reporting and monitoring feature, and other applications are integrated into this new graphical user interface. The Administration Center can be moved to the Tivoli Integrated Portal if the servers being managed are at version 5.5 or later. By deploying the Tivoli Integrated Portal early, you can prepare your system for an upgrade to Tivoli Storage Manager V6.2. Servers at versions earlier than 6.2 that are managed using the V6.2 Administration Center cannot use the version V6.2 features.

Related concepts

Chapter 18, “Managing servers with the Administration Center,” on page 547

The Administration Center not installable on HP-UX

The Administration Center, a Web-based interface for centrally configuring and managing Tivoli Storage Manager servers, cannot be installed on an HP-UX server.

In IBM Tivoli Storage Manager Version 6.2, the Administration Center cannot be installed on an HP-UX server. However, when installed on a supported server platform, the Administration Center can be used to manage HP-UX servers. For Administration Center system requirements, see the following Web site:
<http://www.ibm.com/support/docview.wss?uid=swg21410467>

Sun StorageTek T10000B drive encryption

You can now use tape device encryption with Sun StorageTek T10000B drives. Encryption provides security for data on individual tapes and protects sensitive information that is transported off-site. When enabled, Tivoli Storage Manager handles encrypting and decrypting data on tapes according to specifications set when defining an ECARTRIDGE device class.

Related tasks

“Enabling ECARTRIDGE drive encryption” on page 216

“Disabling ECARTRIDGE drive encryption” on page 216

MOVESIZETHRESH server option

The MOVESIZETHRESH server option default and maximum values have been increased.

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. The default value for MOVESIZETHRESH has been increased from 2048 to 4096; and the maximum value has also been increased from 2048 to 32768.

CHECKTAPEPOS server option to validate data position on tape

With the new CHECKTAPEPOS server option, you can determine the validity and consistency of the position of data blocks on tape.

The CHECKTAPEPOS option applies to only operations using tape drives. It does not apply to non-tape, sequential-access device classes such as FILE or OPTICAL. If the server information about position does not match the position detected by the drive, an error message is displayed, the transaction is rolled back, and the data is not committed to the database.

Part 1. Tivoli Storage Manager basics

Chapter 1. Tivoli Storage Manager overview

IBM Tivoli Storage Manager is an enterprise-wide storage management application. It provides automated storage management services to workstations, personal computers, and file servers from a variety of vendors, with a variety of operating systems.

Tivoli Storage Manager includes the following components:

Server

Server program

The server program provides backup, archive, and space management services to the clients.

You can set up multiple servers in your enterprise network to balance storage, processor, and network resources.

Administrative interface

The administrative interface allows administrators to control and monitor server activities, define management policies for clients, and set up schedules to provide services to clients at regular intervals.

Administrative interfaces available include a command-line administrative client and a Web browser interface called the Administration Center. Tivoli Storage Manager allows you to manage and control multiple servers from a single interface that runs in a Web browser.

Server database and recovery log

The Tivoli Storage Manager server uses a database to track information about server storage, clients, client data, policy, and schedules. The server uses the recovery log as a scratch pad for the database, recording information about client and server actions while the actions are being performed.

Server storage

The server can write data to hard disk drives, disk arrays and subsystems, stand-alone tape drives, tape libraries, and other forms of random- and sequential-access storage. The media that the server uses are grouped into *storage pools*.

The storage devices can be connected directly to the server, or connected via local area network (LAN) or storage area network (SAN).

Client Nodes

A client node can be a workstation, a personal computer, a file server, or even another Tivoli Storage Manager server. The client node has IBM Tivoli Storage Manager client software installed and is registered with the server.

Network-attached storage (NAS) file servers can also be client nodes, but when using NDMP, they do not have Tivoli Storage Manager client software installed.

Backup-archive client

The backup-archive client allows users to maintain backup versions

of files, which they can restore if the original files are lost or damaged. Users can also archive files for long-term storage and retrieve the archived files when necessary. Users themselves or administrators can register workstations and file servers as client nodes with a Tivoli Storage Manager server.

The storage agent is an optional component that may also be installed on a system that is a client node. The storage agent enables LAN-free data movement for client operations and is supported on a number of operating systems.

Network-attached storage file server (using NDMP)

The server can use the Network Data Management Protocol (NDMP) to back up and restore file systems stored on a network-attached storage (NAS) file server. The data on the NAS file server is backed up to a tape library. No Tivoli Storage Manager software needs to be installed on the NAS file server. A NAS file server can also be backed up over the LAN to a Tivoli Storage Manager server. See Chapter 8, “Using NDMP for operations with NAS file servers,” on page 167 for more information, including supported NAS file servers.

Application client

Application clients allow users to perform online backups of data for applications such as database programs. After the application program initiates a backup or restore, the application client acts as the interface to Tivoli Storage Manager. The Tivoli Storage Manager server then applies its storage management functions to the data. The application client can perform its functions while application users are working, with minimal disruption.

The following products provide application clients for use with the Tivoli Storage Manager server:

- Tivoli Storage Manager for Databases
- Tivoli Storage Manager for Enterprise Resource Planning
- Tivoli Storage Manager for Mail

Application program interface (API)

The API allows you to enhance existing applications to use the backup, archive, restore, and retrieve services that Tivoli Storage Manager provides. Tivoli Storage Manager API clients can register as client nodes with a Tivoli Storage Manager server.

Tivoli Storage Manager for Space Management

Tivoli Storage Manager for Space Management provides space management services for workstations on some platforms. The space management function is essentially a more automated version of archive. Tivoli Storage Manager for Space Management automatically migrates files that are less frequently used to server storage, freeing space on the workstation. The migrated files are also called *space-managed files*.

Users can recall space-managed files automatically simply by accessing them as they normally would from the workstation. Tivoli Storage Manager for Space Management is also known as the space manager client, or the hierarchical storage management (HSM) client.

Storage agents

The storage agent is an optional component that may be installed on a system that is also a client node. The storage agent enables LAN-free data movement for client operations.

The storage agent is available for use with backup-archive clients and application clients on a number of operating systems. The Tivoli Storage Manager for Storage Area Networks product includes the storage agent.

For information about supported operating systems for clients, see the IBM Tivoli Storage Manager Web site at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

Client programs such as the backup-archive client and the HSM client (space manager) are installed on systems that are connected through a LAN and are registered as client nodes. From these client nodes, users can back up, archive, or migrate files to the server.

The following sections present key concepts and information about IBM Tivoli Storage Manager. The sections describe how Tivoli Storage Manager manages client files based on information provided in administrator-defined policies, and manages devices and media based on information provided in administrator-defined Tivoli Storage Manager storage objects.

The final section gives an overview of tasks for the administrator of the server, including options for configuring the server and how to maintain the server.

Concepts:
"How client data is stored"
"How the server manages storage" on page 15

How client data is stored

Tivoli Storage Manager policies are rules that determine how the client data is stored and managed. The rules include where the data is initially stored, how many backup versions are kept, how long archive copies are kept, and so on.

You can have multiple policies and assign the different policies as needed to specific clients, or even to specific files. Policy assigns a location in server storage where data is initially stored. Server storage is divided into storage pools that are groups of storage volumes.

Server storage can include disk and tape volumes.

When you install Tivoli Storage Manager, you have a default policy that you can use. For details about this default policy, see "Reviewing the standard policy" on page 437. You can modify this policy and define additional policies.

Clients use Tivoli Storage Manager to store data for any of the following purposes:

Backup and restore

The backup process copies data from client workstations to server storage to ensure against loss of data that is regularly changed. The server retains versions of a file according to policy, and replaces older versions of the file with newer versions. Policy includes the number of versions and the retention time for versions.

A client can restore the most recent version of a file, or can restore earlier versions.

Archive and retrieve

The archive process copies data from client workstations to server storage for long-term storage. The process can optionally delete the archived files from the client workstations. The server retains archive copies according to the policy for archive retention time. A client can retrieve an archived copy of a file.

Instant archive and rapid recovery

Instant archive is the creation of a complete set of backed-up files for a client. The set of files is called a *backup set*. A backup set is created on the server from the most recently backed-up files that are already stored in server storage for the client. Policy for the backup set consists of the retention time that you choose when you create the backup set.

You can copy a backup set onto compatible portable media, which can then be taken directly to the client for rapid recovery without the use of a network and without having to communicate with the Tivoli Storage Manager server.

Migration and recall

Migration, a function of the Tivoli Storage Manager for Space Management program, frees up client storage space by copying files from workstations to server storage. On the client, the Tivoli Storage Manager for Space Management program replaces the original file with a stub file that points to the original in server storage. Files are recalled to the workstations when needed.

This process is also called hierarchical storage management (HSM). Once configured, the process is transparent to the users. Files are migrated and recalled automatically.

Policy determines when files are considered for automatic migration. On the UNIX or Linux systems that support the Tivoli Storage Manager for Space Management program, policies determine whether files must be backed up to the server before being migrated. Space management is also integrated with backup. If the file to be backed up is already migrated to server storage, the file is backed up from there.

Figure 1 on page 7 shows how policy is part of the Tivoli Storage Manager process for storing client data.

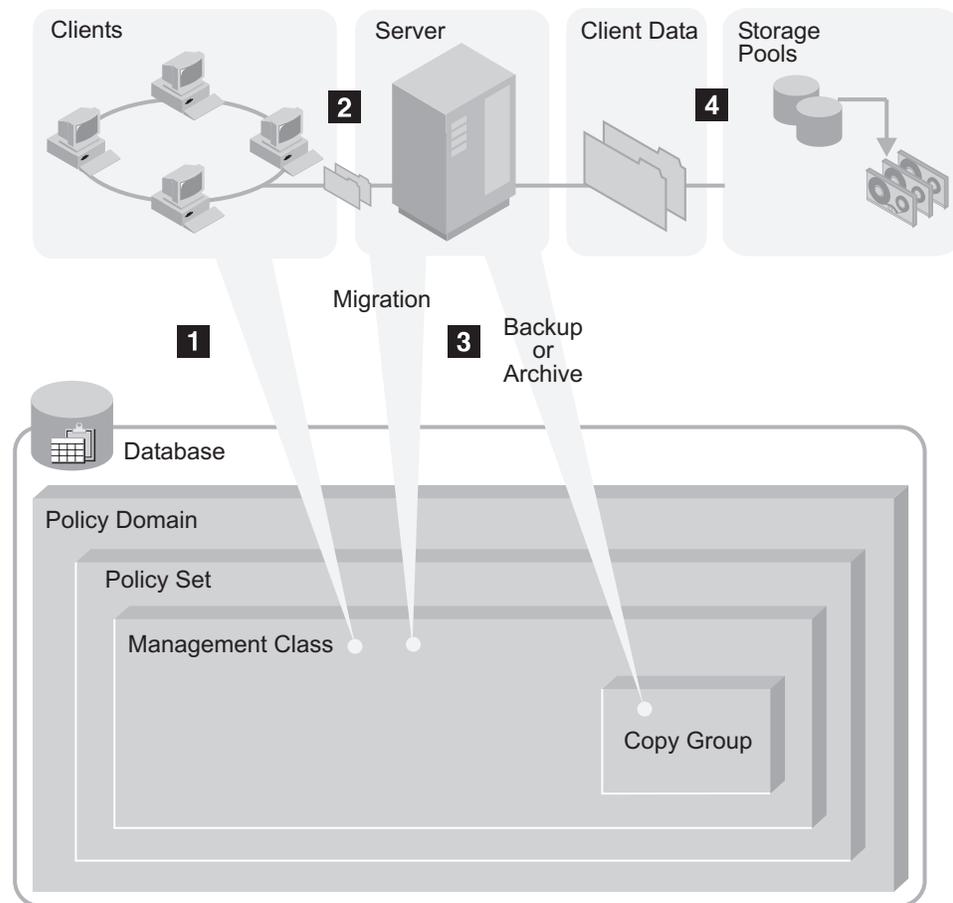


Figure 1. How IBM Tivoli Storage Manager Controls Backup, Archive, and Migration Processes

The steps in the process are as follows:

- **1** A client initiates a backup, archive, or migration operation. The file involved in the operation is bound to a management class. The management class is either the default or one specified for the file in client options (the client's include-exclude list).
- **2** If the file is a candidate for backup, archive, or migration based on information in the management class, the client sends the file and file information to the server.
- **3** The server checks the management class that is bound to the file to determine the *destination*, the name of the Tivoli Storage Manager storage pool where the server initially stores the file. For backed-up and archived files, destinations are assigned in the backup and archive copy groups, which are within management classes. For space-managed files, destinations are assigned in the management class itself.

The storage pool can be a group of disk volumes, tape volumes, or optical volumes.

- **4** The server stores the file in the storage pool that is identified as the storage destination.

The Tivoli Storage Manager server saves information in its database about each file that it backs up, archives, or migrates.

If you set up server storage in a hierarchy, Tivoli Storage Manager can later migrate the file to a storage pool different from the one where the file was

initially stored. For example, you may want to set up server storage so that Tivoli Storage Manager migrates files from a disk storage pool to tape volumes in a tape storage pool.

Files remain in server storage until they expire and expiration processing occurs, or until they are deleted from server storage. A file expires because of criteria that are set in policy. For example, the criteria include the number of versions allowed for a file and the number of days that have elapsed since a file was deleted from the client's file system. If data retention protection is activated, an archive object cannot be inadvertently deleted.

For information on assigning storage destinations in copy groups and management classes, and on binding management classes to client files, see Chapter 14, "Implementing policies for client data," on page 435.

For information on managing the database, see Chapter 21, "Managing the database and recovery log," on page 593.

For information about storage pools and storage pool volumes, see Chapter 10, "Managing storage pools and volumes," on page 221.

For information about event-based policy, deletion hold, and data retention protection, see Chapter 14, "Implementing policies for client data," on page 435.

Data-protection options

Tivoli Storage Manager provides a variety of backup and archive operations, allowing you to select the right protection for the situation.

Table 6 shows some examples of the protection options.

Table 6. Examples of meeting your goals with Tivoli Storage Manager

For this goal...	Do this...
Back up files that are on a user's workstation, and have the ability to restore individual files.	Use the backup-archive client to perform incremental backups or selective backups.
Back up a file server, and have the ability to restore individual files.	Use the backup-archive client to perform incremental backups or selective backups. If the file server is a network-attached storage file server that is supported, you can have the server use NDMP to perform image backups. This support is available in the Tivoli Storage Manager Extended Edition product.
Make restore media portable, or make restores easier to perform remotely.	Use the backup-archive client to perform incremental backups, and then generate backup sets by using the Tivoli Storage Manager server.
Provide the ability to more easily restore the entire contents of a single logical volume, instead of restoring individual files.	Use the backup-archive client to perform logical volume backups (also called image backups).
Set up records retention to meet legal or other long-term storage needs.	Use the backup-archive client to occasionally perform archiving. To ensure that the archiving occurs at the required intervals, use central scheduling.

Table 6. Examples of meeting your goals with Tivoli Storage Manager (continued)

For this goal...	Do this...
Create an archive for a backup-archive client, from data that is already stored for backup.	Use the backup-archive client to perform incremental backups, and then generate a backup set by using the Tivoli Storage Manager server. This is also called <i>instant archive</i> .
Provide the ability to restore data to a point in time.	<p>Use the backup-archive client to regularly perform incremental backups (either manually or automatically through schedules). Then do one of the following:</p> <ul style="list-style-type: none"> • Set up policy to ensure that data is preserved in server storage long enough to provide the required service level. See “Setting policy to enable point-in-time restore for clients” on page 488 for details. • Create backup sets for the backup-archive client on a regular basis. Set the retention time to provide the required service level. See “Creating and using client backup sets” on page 502 for details.
Save a set of files and directories before making significant changes to them.	<p>Use the backup-archive client to archive the set of files and directories.</p> <p>If this kind of protection is needed regularly, consider creating backup sets from backup data already stored for the client. Using backup sets instead of frequent archive operations can reduce the amount of metadata that must be stored in the server's database.</p>
Manage a set of related files, which are not in the same file system, with the same backup, restore, and server policies.	<p>Use the backup group command on the backup-archive client to create a logical grouping of a set of files, which can be from one or more physical file systems. The group backup process creates a virtual file space in server storage to manage the files, because the files might not be from one file system on the client. Actions such as policy binding, migration, expiration, and export are applied to the group as a whole. See the <i>Backup-Archive Clients Installation and User's Guide</i> for details.</p>
Back up data for an application that runs continuously, such as a database application (for example, DB2 or Oracle) or a mail application (Lotus Domino®).	Use the appropriate application client. For example, use Tivoli Storage Manager for Mail to protect the Lotus Domino application.
Exploit disk hardware capable of data snapshots.	Use the appropriate component in the Tivoli Storage Manager for Hardware product, such as System Storage™ Archive Manager for IBM Enterprise Storage Server® for DB2.
Make backups transparent to end users.	Use the backup-archive client with centrally scheduled backups that run during off-shift hours. Monitor the schedule results.

Table 6. Examples of meeting your goals with Tivoli Storage Manager (continued)

For this goal...	Do this...
Reduce the load on the LAN by moving backup data over your SAN.	Use LAN-free data movement or, for supported network-attached storage (NAS) file servers, use NDMP operations.

Schedule the backups of client data to help enforce the data management policy that you establish. If you schedule the backups, rather than rely on the clients to perform the backups, the policy that you establish is followed more consistently. See Chapter 16, "Scheduling operations for client nodes," on page 525.

The standard backup method that Tivoli Storage Manager uses is called *progressive incremental backup*. It is a unique and efficient method for backup. See "Progressive incremental backups" on page 13.

Table 7 summarizes the client operations that are available. In all cases, the server tracks the location of the backup data in its database. Policy that you set determines how the backup data is managed.

Table 7. Summary of client operations

Type of operation	Description	Usage	Restore options	For more information
Progressive incremental backup	The standard method of backup used by Tivoli Storage Manager. After the first, full backup of a client system, incremental backups are done. Incremental backup by date is also available. No additional full backups of a client are required after the first backup.	Helps ensure complete, effective, policy-based backup of data. Eliminates the need to retransmit backup data that has not been changed during successive backup operations.	The user can restore just the version of the file that is needed. Tivoli Storage Manager does <i>not</i> need to restore a base file followed by incremental backups. This means reduced time and fewer tape mounts, as well as less data transmitted over the network.	See "Incremental backup" on page 452 and the <i>Backup-Archive Clients Installation and User's Guide</i> .
Selective backup	Backup of files that are selected by the user, regardless of whether the files have changed since the last backup.	Allows users to protect a subset of their data independent of the normal incremental backup process.	The user can restore just the version of the file that is needed. Tivoli Storage Manager does <i>not</i> need to restore a base file followed by incremental backups. This means reduced time and fewer tape mounts, as well as less data transmitted over the network.	See "Selective backup" on page 454 and the <i>Backup-Archive Clients Installation and User's Guide</i> .

Table 7. Summary of client operations (continued)

Type of operation	Description	Usage	Restore options	For more information
Adaptive subfile backup	<p>A backup method that backs up only the <i>parts</i> of a file that have changed since the last backup. The server stores the base file (the complete initial backup of the file) and subsequent subfiles (the changed parts) that depend on the base file.</p> <p>The process works with either the standard progressive incremental backup or with selective backup.</p> <p>Applicable to clients on Windows systems.</p>	<p>Maintains backups of data while minimizing connect time and data transmission for the backup of mobile and remote users.</p>	<p>The base file plus a maximum of one subfile is restored to the client.</p>	<p>See “Enabling clients to use subfile backup” on page 511 and the <i>Backup-Archive Clients Installation and User’s Guide</i>.</p>
Journal-based backup	<p>Aids all types of backups (progressive incremental backup, selective backup, adaptive subfile backup) by basing the backups on a list of changed files. The list is maintained on the client by the journal engine service of IBM Tivoli Storage Manager.</p>	<p>Reduces the amount of time required for backup. The files eligible for backup are known before the backup operation begins.</p> <p>Applicable to clients on AIX and Windows systems, except Windows 2003 64-bit IA64.</p>	<p>Journal-based backup has no effect on how files are restored; this depends on the type of backup performed.</p>	<p>See the <i>Backup-Archive Clients Installation and User’s Guide</i>.</p>
Image backup	<p>Full volume backup.</p> <p>Nondisruptive, on-line backup is possible for Windows clients by using the Tivoli Storage Manager snapshot function.</p>	<p>Allows backup of an entire file system or raw volume as a single object. Can be selected by backup-archive clients on UNIX, Linux, and Windows systems.</p>	<p>The entire image is restored.</p>	<p>See “Policy for logical volume backups” on page 483 and the <i>Backup-Archive Clients Installation and User’s Guide</i>.</p>
Image backup with differential backups	<p>Full volume backup, which can be followed by subsequent differential backups.</p>	<p>Used only for the image backups of NAS file servers, performed by the server using NDMP operations.</p>	<p>The full image backup plus a maximum of one differential backup are restored.</p>	<p>See Chapter 8, “Using NDMP for operations with NAS file servers,” on page 167.</p>
Backup using hardware snapshot capabilities	<p>A method of backup that exploits the capabilities of IBM Enterprise Storage Server FlashCopy® and EMC TimeFinder to make copies of volumes used by database servers. The Tivoli Storage Manager for Hardware product then uses the volume copies to back up the database volumes.</p>	<p>Implements high-efficiency backup and recovery of business-critical applications while virtually eliminating backup-related downtime or user disruption on the database server.</p>	<p>Details depend on the hardware.</p>	<p>See the documentation for IBM Tivoli Storage Manager for hardware components.</p>

Table 7. Summary of client operations (continued)

Type of operation	Description	Usage	Restore options	For more information
Group backup	<p>A method that backs up files that you specify as a named group. The files can be from one or more file spaces. The backup can be a full or a differential backup.</p> <p>Applicable to clients on UNIX and Linux systems.</p>	<p>Creates a consistent point-in-time backup of a group of related files. The files can reside in different file spaces on the client. All objects in the group are assigned to the same management class. The server manages the group as a single logical entity, and stores the files in a virtual file space in server storage.</p> <p>A group can be included in a backup set.</p>	<p>The user can select to restore the entire group or just selected members of the group. The user can restore just the version of the file that is needed.</p>	<p>See the <i>Backup-Archive Clients Installation and User's Guide</i>.</p>
Archive	<p>The process creates a copy of files and stores them for a specific time.</p>	<p>Use for maintaining copies of vital records for legal or historical purposes.</p> <p>Note: If you need to frequently create archives for the same data, consider using instant archive (backup sets) instead. Frequent archive operations can create a large amount of metadata in the server database resulting in increased database growth and decreased performance for server operations such as expiration. Frequently, you can achieve the same objectives with incremental backup or backup sets. Although the archive function is a powerful way to store inactive data with fixed retention, it should not be used on a frequent and large scale basis as the primary backup method.</p>	<p>The selected version of the file is retrieved on request.</p>	<p>See "Archive" on page 455 and the <i>Backup-Archive Clients Installation and User's Guide</i>.</p>

Table 7. Summary of client operations (continued)

Type of operation	Description	Usage	Restore options	For more information
Instant archive	The process creates a backup set of the most recent versions of the files for the client, using files already in server storage from earlier backup operations.	Use when portability of the recovery media or rapid recovery of a backup-archive client is important. Also use for efficient archiving.	The files are restored directly from the backup set. The backup set resides on media that can be mounted on the client system, such as a CD, a tape drive, or a file system. The Tivoli Storage Manager server does not have to be contacted for the restore process, so the process does not use the network or the server.	See “Creating and using client backup sets” on page 502.

Progressive incremental backups

The terms *differential* and *incremental* are often used to describe backups. The standard method of backup used by Tivoli Storage Manager is progressive incremental.

The terms *differential* and *incremental* have the following meanings:

- A differential backup backs up files that have changed since the last full backup.
 - If a file changes after the full backup, the changed file is backed up again by *every* subsequent differential backup.
 - All files are backed up at the next full backup.
- An incremental backup backs up only files that have changed since the last backup, whether that backup was a full backup or another incremental backup.
 - If a file changes after the full backup, the changed file is backed up *only* by the next incremental backup, not by all subsequent incremental backups.
 - If a file has not changed since the last backup, the file is not backed up.

Tivoli Storage Manager takes incremental backup one step further. After the initial full backup of a client, no additional full backups are necessary because the server, using its database, keeps track of whether files need to be backed up. Only files that change are backed up, and then entire files are backed up, so that the server does not need to reference base versions of the files. This means savings in resources, including the network and storage.

If you choose, you can force full backup by using the selective backup function of a client in addition to the incremental backup function. You can also choose to use adaptive subfile backup, in which the server stores the base file (the complete initial backup of the file) and subsequent subfiles (the changed parts) that depend on the base file.

Backup methods are summarized in Table 7 on page 10.

Storage-pool and server-database backups

Tivoli Storage Manager protects client data by letting you back up storage pools and the database.

You can back up client backup, archive, and space-managed data in primary storage pools to copy storage pools. You can also copy active versions of client backup data from primary storage pools to active-data pools. The server can automatically access copy storage pools and active-data pools to retrieve data. See “Storage pool protection overview” on page 756.

You can also back up the server's database. The database is key to the server's ability to track client data in server storage. See “Database and recovery log protection overview” on page 758.

These backups can become part of a disaster recovery plan, created automatically by the disaster recovery manager. See:

Chapter 26, “Using disaster recovery manager,” on page 801

Data movement to server storage

Tivoli Storage Manager provides several methods for sending client data to server storage.

In many configurations, the Tivoli Storage Manager client sends its data to the server over the LAN. The server then transfers the data to a device that is attached to the server. You can also use storage agents that are installed on client nodes to send data over a SAN. This minimizes use of the LAN and the use of the computing resources of both the client and the server. For details, see:

“LAN-free data movement” on page 54

For network-attached storage, use NDMP operations to avoid data movement over the LAN. For details, see “NDMP backup operations” on page 57.

Consolidation of backed-up client data

By grouping the backed-up data for a client, you can minimize the number of media mounts required for client recovery.

The server offers you methods for doing this:

Collocation

The server can keep each client's files on a minimal number of volumes within a storage pool. Because client files are consolidated, restoring collocated files requires fewer media mounts. However, backing up files from different clients requires more mounts.

You can have the server collocate client data when the data is initially stored in server storage. If you have a storage hierarchy, you can also have the data collocated when the server migrates the data from the initial storage pool to the next storage pool in the storage hierarchy.

Another choice you have is the level of collocation. You can collocate by client, by file space per client, or by group. Your selection depends on the size of the file spaces being stored and the restore requirements.

See “Keeping client files together using collocation” on page 310.

Active-data pools

Active-data pools are storage pools that contain only the active versions of

client backup data. Archive data and data migrated by Hierarchical Space Management (HSM) clients are not allowed in active-data pools.

Active-data pools can be associated with three types of devices: sequential-access disk (FILE), removable media (tape or optical), or sequential-access volumes on another Tivoli Storage Manager server. There are three types of active-data pool, each of which has distinct advantages. For example, an active-data pool associated with sequential-access disk is particularly well-suited for fast restores of client data because tapes do not have to be mounted and because the server does not have to position past inactive files.

For more information, see “Backing up storage pools” on page 760.

Backup set creation

You can generate a backup set for each backup-archive client. A backup set contains all active backed-up files that currently exist for that client in server storage. The process is also called instant archive.

The backup set is portable and is retained for the time that you specify. Creation of the backup set consumes more media because it is a copy in addition to the backups that are already stored.

See “Creating and using client backup sets” on page 502.

Moving data for a client node

You can consolidate data for a client node by moving the data within server storage. You can move it to a different storage pool, or to other volumes in the same storage pool.

See “Moving data belonging to a client node” on page 354.

How the server manages storage

Through the server, you manage the devices and media used to store client data. The server integrates the management of storage with the policies that you define for managing client data.

Device support

With Tivoli Storage Manager, you can use of a variety of devices for server storage.

Tivoli Storage Manager can use direct-attached storage devices as well as network-attached storage devices.

See the current list on the IBM Tivoli Storage Manager Web site at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

Tivoli Storage Manager represents physical storage devices and media with the following administrator-defined objects:

Library

A library is one or more drives (and possibly robotic devices) with similar media mounting requirements.

Drive

Each drive represents a drive mechanism in a tape device.

Data mover

A data mover represents a device that accepts requests from Tivoli Storage Manager to transfer data on behalf of the server. Data movers transfer data between storage devices.

Path A path represents how a source accesses a destination. For example, the source can be a server, and the destination can be a tape drive. A path defines the one-to-one relationship between a source and a destination. Data may flow from the source to the destination, and back.

Device class

Each device is associated with a device class that specifies the device type and how the device manages its media.

Storage pools and volumes

A storage pool is a named collection of volumes that have the same media type. A storage pool is associated with a device class. A storage pool volume is associated with a specific storage pool.

For example, an LTO tape storage pool contains only LTO tape volumes.

For details about device concepts, see Chapter 3, “Storage device concepts,” on page 41.

Data migration through the storage hierarchy

You can organize the server's storage pools into one or more hierarchical structures. This storage hierarchy allows flexibility in a number of ways. For example, you can set policy to have clients send their backup data to disks for faster backup operations, then later have the server automatically migrate the data to tape.

See “Storage pool hierarchies” on page 241.

Removal of expired data

A policy that you define controls when client data automatically expires from the Tivoli Storage Manager server. The expiration process is how the server implements the policy.

For example, you have a backup policy that specifies that three versions of a file be kept. File A is created on the client, and backed up. Over time, the user changes file A, and three versions of the file are backed up to the server. Then the user changes file A again. When the next incremental backup occurs, a fourth version of file A is stored, and the oldest of the four versions is eligible for expiration.

To remove data that is eligible for expiration, a server expiration process marks data as expired and deletes metadata for the expired data from the database. The space occupied by the expired data is then available for new data.

You control the frequency of the expiration process by using a server option, or you can start the expiration processing by command or scheduled command.

See “Running expiration processing to delete expired files” on page 472.

Media reuse by reclamation

As server policies automatically expire data, the media where the data is stored accumulates unused space. The Tivoli Storage Manager server implements a process, called *reclamation*, which allows you to reuse media without traditional tape rotation.

Reclamation is a server process that automatically defragments media by consolidating unexpired data onto other media when the free space on media reaches a defined level. The reclaimed media can then be used again by the server. Reclaiming media allows the automated circulation of media through the storage management process. Use of reclamation can help minimize the number of media that you need to have available.

Chapter 2. Tivoli Storage Manager concepts

The server comes with many defaults so that you can begin using its services immediately. The amount and importance of the data protected by Tivoli Storage Manager, your business process requirements, and other factors make it likely that you need to adjust and customize the server's behavior.

Your changing storage needs and client requirements can mean on-going configuration changes and monitoring. The server's capabilities are extensively described in this guide. To get an introduction to the tasks available to an administrator of Tivoli Storage Manager, read the following sections:

Administrative Tasks:
"Interfaces to Tivoli Storage Manager"
"Storage configuration and management" on page 21
"Management of client operations" on page 25
"Server maintenance" on page 33
"Protecting the server" on page 37
Chapter 18, "Managing servers with the Administration Center," on page 547

Note: When you start the Tivoli Storage Manager server, the server attempts to change the ulimit values to unlimited. In general, this helps to ensure optimal performance and to assist in debugging. If you are a non-root user when you start the server, attempts to change the ulimits might fail. To ensure proper server operation if you are running as a non-root user, make sure that you set the ulimits as high as possible, preferably to unlimited, before starting the server.

This includes setting DB2 user limits as high as possible. DB2 relies on private data memory for sort memory allocations during SQL processing. Insufficient shared heap memory can lead to Tivoli Storage Manager server failures when interacting with DB2. For more information on setting the appropriate platform values, see the following Technote: <http://www.ibm.com/support/docview.wss?uid=swg21212174>

Interfaces to Tivoli Storage Manager

Tivoli Storage Manager has several types of interfaces that allow you to work with many different applications.

The following interfaces are provided:

- Graphical user interfaces

For the clients, there are graphical user interfaces for the backup-archive client and the space manager client (if installed, on supported operating systems). For information about using the interfaces, see the online information or the *Installation Guide*.

- The IBM Tivoli Storage Manager for Windows program folder.
- The IBM Tivoli Storage Manager Management Console, selected from the IBM Tivoli Storage Manager program folder or the desktop. The IBM Tivoli Storage Manager Console is a Microsoft Management Console snap-in that provides:

- Wizards to assist with Tivoli Storage Manager administration and configuration tasks
- A Windows-style tree view of the storage management resource network
- Network scan utilities that can be used to locate Tivoli Storage Manager client nodes and server nodes for remote management
- A net send feature that can be used to notify operators of Tivoli Storage Manager mount requests and status messages
- Web interfaces for server administration and for the backup-archive client

The Administration Center allows you to access Tivoli Storage Manager server functions from any workstation using a supported Web browser. The interface also allows Web access to the command line. See Chapter 18, “Managing servers with the Administration Center,” on page 547 for more information.

The Web backup-archive client (Web client) allows an authorized user to remotely access a client to run backup, archive, restore, and retrieve processes. The Web browser must have the appropriate support for Java™. See the *Backup-Archive Clients Installation and User’s Guide* for requirements.
- The command-line interface

For information about using the command-line interface of the administrative client, see the *Administrator’s Reference*. For information about using the command-line interface of the backup-archive client or other clients, see the user’s guide for that client.
- The application program interface

For more information, see the *IBM Tivoli Storage Manager Using the Application Program Interface*.
- Access to information in the server’s database using standard SQL SELECT statements. Tivoli Storage Manager Versions 6.1 and later use the DB2 open database connectivity (ODBC) driver to query the database and display the results. For more information, see “Using SQL to query the IBM Tivoli Storage Manager database” on page 618.

Server options

Server options let you customize the server and its operations.

Server options can affect the following:

- Server communications
- Storage
- Database and recovery log operations
- Client transaction performance

Server options are in the server options file. Some options can be changed and made active immediately by using the command, SETOPT. Most server options are changed by editing the server options file and then halting and restarting the server to make the changes active. See the *Administrator’s Reference* for details about the server options file and reference information for all server options.

Storage configuration and management

Configuring and managing storage for efficiency and capacity are important tasks for an administrator.

The server uses its storage for the data it manages for clients. The storage can be a combination of devices.

- Disk
- Tape drives that are either manually operated or automated
- Other drives that use removable media

Devices can be locally attached, or accessible through a SAN. Key decisions in configuring and managing the storage include:

- Selecting the devices and media that will form the server storage. This includes deciding whether library devices will be shared among Tivoli Storage Manager servers.
- Designing the storage hierarchy for efficient backups and optimal storage usage.
- Using product features that allow the server to provide services to clients while minimizing traffic on the communications network:
 - LAN-free data movement
 - Data movement using NDMP to protect data on network-attached storage (NAS) file servers when backing up to libraries directly attached to the NAS file servers
- Using the Tivoli Storage Manager product to help you to manage the drives and media, or using an external media manager to do the management outside of the Tivoli Storage Manager product.

For an introduction to key storage concepts, see Chapter 3, “Storage device concepts,” on page 41.

Disk devices

Disk devices can be used with Tivoli Storage Manager for storing the database and recovery log or client data that is backed up, archived, or migrated from client nodes.

The server can store data on disk by using random-access volumes (device type of DISK) or sequential-access volumes (device type of FILE).

The Tivoli Storage Manager product allows you to exploit disk storage in ways that other products do not. You can have multiple client nodes back up to the same disk storage pool at the same time, and still keep the data for the different client nodes separate. Other products also allow you to back up different systems at the same time, but only by interleaving the data for the systems, leading to slower restore processes.

If you have enough disk storage space, data can remain on disk permanently or temporarily, depending on the amount of disk storage space that you have. Restore process performance from disk can be very fast compared to tape.

You can have the server later move the data from disk to tape; this is called migration through the storage hierarchy. Other advantages to this later move to tape include:

- Ability to collocate data for clients as the data is moved to tape

- Streaming operation of tape drives, leading to better tape drive performance
- More efficient use of tape drives by spreading out the times when the drives are in use

For information about storage hierarchy and setting up storage pools on disk devices, see:

Chapter 10, “Managing storage pools and volumes,” on page 221

Removable media devices

Removable media devices can be used with Tivoli Storage Manager for storage of client data that is backed up, archived, or migrated from client nodes; storage of database backups; and the exporting, that is, moving, of data to another server.

The following topics provide an overview of how to use removable media devices with Tivoli Storage Manager.

For guidance and scenarios on configuring your tape devices, see:

Chapter 6, “Configuring storage devices,” on page 89

Device classes

A device class represents a set of storage devices with similar availability, performance, and storage characteristics.

You must define device classes for the drives available to the Tivoli Storage Manager server. You specify a device class when you define a storage pool so that the storage pool is associated with drives.

For more information about defining device classes, see Chapter 9, “Defining device classes,” on page 199.

Removable media operations

Routine removable media operations include preparing and controlling media for reuse, ensuring that sufficient media are available, and mounting volumes in response to server requests, for manually operated drives. Removable media operations also include managing libraries and drives.

For information about removable media operations, see:

Chapter 7, “Managing removable media operations,” on page 133

Attaching and configuring devices

Before you can use tape and optical devices with Tivoli Storage Manager, you must attach the device to the server and install the appropriate device driver on your system.

1. Attach the devices to your system, which includes physically attaching the devices, configuring the device drivers, and determining the device names.
2. Define the following: a library for the drives, the drives, paths from the server to library and drives, a device class, and a storage pool associated with the device class.
3. Include the storage pool in your storage hierarchy. To perform these tasks you can use the administrative client command line or the Administration Center.

See Chapter 5, “Using devices with the server system,” on page 79 for more information.

Defined volumes and scratch volumes

A defined volume is a *private volume* and is assigned to a specific storage pool. A volume that is not defined is a *scratch volume*. If you want a volume to be used only when it is requested by name, you must also define it to Tivoli Storage Manager.

You can use tapes as scratch volumes, up to the number of scratch volumes you specified for the storage pool. Using scratch volumes allows Tivoli Storage Manager to acquire volumes as needed. A storage pool can request available scratch volumes up to the number specified for that storage pool.

You must define private volumes to Tivoli Storage Manager, assigning each to a specific storage pool. However, if a storage pool contains only private volumes and runs out of them, storage operations to that pool stop until more volumes are defined.

All tape volumes must have standard tape labels before Tivoli Storage Manager can use them.

Preparing tape volumes with a manual drive

Tape media must be prepared before it can be used. Preparation includes labeling volumes and, if necessary, defining private volumes. If you are using only one tape drive, consider labeling several tapes at a time.

Complete the following steps:

1. From an administrative client command line, use a drive in the library named MANUAL to label a volume as DSM001:

```
label libvolume manual dsm001
```
2. Define any volumes that are to be used as private volumes. For example, define the volume you just labeled:

```
define volume tapepool dsm001
```

Preparing tape volumes with an automated library

All tape volumes must have standard tape labels before Tivoli Storage Manager can use them.

To label tapes with an automated library:

1. Remove any tapes that you do not want to use with IBM Tivoli Storage Manager, and load the tapes to be labeled.
2. Use the LABEL LIBVOLUME command to label and check in the volumes. From an administrative client command line, begin a search of an automated library named AUTOLIB for unlabeled volumes and label them based on their barcodes: For example:

```
label libvolume autolib search=yes labelsource=barcode  
checkin=scratch
```
3. Define any volumes that are to be used as private volumes: For example:

```
define volume autopool dsm001
```

Migrating data from disk to tape

After you set up disk and tape storage pools, you can configure the server so that client data can be migrated to tape. By migrating data to tape from a disk storage pool, you can verify that tape devices are properly set up.

Migration requires tape mounts. The mount messages are directed to the console message queue and to any administrative client that has been started with either the mount mode or console mode option. To have the server migrate data from BACKUPPOOL to AUTOPOOL and from ARCHIVEPOOL to TAPEPOOL do the following:

```
update stgpool backuppool nextstgpool=autopool
update stgpool archivepool nextstgpool=tapepool
```

The server can perform migration as needed, based on migration thresholds that you set for the storage pools. Because migration from a disk to a tape storage pool uses resources such as drives and operators, you might want to control when migration occurs. To do so, you can use the MIGRATE STGPOOL command:

```
migrate stgpool backuppool
```

To migrate from a disk storage pool to a tape storage pool, devices must be allocated and tapes must be mounted. For these reasons, you may want to ensure that migration occurs at a time that is best for your situation. You can control when migration occurs by using migration thresholds.

You might not want to empty the disk storage pool every time migration occurs by setting the low migration threshold to 0. Normally, you might want to keep the low threshold at 40%, and vary the high threshold from as high as 90% to as low as 50%.

See “Migrating disk storage pools” on page 253 and the *Administrator’s Reference* for more information.

Managing storage pools and volumes

Backed-up, archived, and space-managed files are stored in groups of volumes that are called storage pools. Because each storage pool is assigned to a device class, you can logically group your storage devices to meet your storage-management needs.

The following are other examples of what you can control for a storage pool:

Collocation

The server can keep each client's files on a minimal number of volumes within a storage pool. Because client files are consolidated, restoring collocated files requires fewer media mounts. However, backing up files from different clients requires more mounts.

Reclamation

Files on sequential access volumes may expire, move, or be deleted. The reclamation process consolidates the active, unexpired data on many volumes onto fewer volumes. The original volumes can then be reused for new data, making more efficient use of media.

Storage pool backup

Client backup, archive, and space-managed data in primary storage pools can be backed up to copy storage pools for disaster recovery purposes. As

client data is written to the primary storage pools, it can also be simultaneously written to copy storage pools.

Copy active data

The active versions of client backup data can be copied to active-data pools. Active-data pools provide a number of benefits. For example, if the device type associated with an active-data pool is sequential-access disk (FILE), you can eliminate the need for disk staging pools. Restoring client data is faster because FILE volumes are not physically mounted, and the server does not need to position past inactive files that do not need to be restored.

An active-data pool that uses removable media, such as tape or optical, lets you reduce the number of volumes for onsite and offsite storage. (Like volumes in copy storage pools, volumes in active-data pools can be moved offsite for protection in case of disaster.) If you vault data electronically to a remote location, a SERVER-type active-data pool lets you save bandwidth by copying and restoring only active data.

As backup client data is written to primary storage pools, the active versions can be simultaneously written to active-data pools.

Cache When the server migrates files from disk storage pools, duplicate copies of the files can remain in cache (disk storage) for faster retrieval. Cached files are deleted only when space is needed. However, client backup operations that use the disk storage pool may have poorer performance.

You can establish a hierarchy of storage pools. The hierarchy can be based on the speed or the cost of the devices associated with the pools. Tivoli Storage Manager migrates client files through this hierarchy to ensure the most efficient use of a server's storage devices.

You manage storage volumes by defining, updating, and deleting volumes, and by monitoring the use of server storage. You can also move files within and across storage pools to optimize the use of server storage.

For more information about storage pools and volumes and taking advantage of storage pool features, see Chapter 10, "Managing storage pools and volumes," on page 221.

Increasing disk storage-pool sizes

When the server is installed, Tivoli Storage Manager creates storage pool volumes. You can increase disk storage-pool sizes if the volumes created at server installation are not adequate.

Management of client operations

Because the key task of the server is to provide services to clients, many of the server administrator's tasks deal with client operations.

Tasks include the following:

- Registering clients and customizing client operations
- Ensuring that client operations meet security requirements
- Providing required levels of service by customizing policies
- Automating protection by using schedules

After you have created schedules, you manage and coordinate those schedules. Your tasks include the following:

- Verify that the schedules ran successfully.
- Determine how long Tivoli Storage Manager retains information about schedule results (*event records*) in the database.
- Balance the workload on the server so that all scheduled operations complete.

For more information about client operations, see the following sections:

- For setting up an include-exclude list for clients, see “Getting users started” on page 438.
- For automating client operations, see Chapter 16, “Scheduling operations for client nodes,” on page 525.
- For running the scheduler on a client system, see the user's guide for the client.
- For setting up policy domains and management classes, see Chapter 14, “Implementing policies for client data,” on page 435.

For more information about these tasks, see Chapter 17, “Managing schedules for client nodes,” on page 531

Managing client nodes

A basic administrative task is adding client nodes and giving the systems that the nodes represent access to the services and resources of the Tivoli Storage Manager server.

The Tivoli Storage Manager server supports a variety of client nodes. You can register the following types of clients and servers as client nodes:

- Tivoli Storage Manager backup-archive client
- Application clients that provide data protection through one of the following products: Tivoli Storage Manager for Application Servers, Tivoli Storage Manager for Databases, Tivoli Storage Manager for Enterprise Resource Planning, or Tivoli Storage Manager for Mail.
- Tivoli Storage Manager for Space Management client (called space manager client or HSM client)
- A NAS file server for which the Tivoli Storage Manager server uses NDMP for backup and restore operations
- Tivoli Storage Manager source server (registered as a node on a target server)

When you register clients, you have choices to make about the following:

- Whether the client should compress files before sending them to the server for backup
- Whether the client node ID has the authority to delete its files from server storage
- Whether an administrator ID that matches the client ID is created, for remote client operations

Other important tasks include the following:

Controlling client options from the server

Client options on client systems allow users to customize backup, archive, and space management operations, as well as schedules for these operations. On most client systems, the options are in a file called *dsm.opt*. In some cases, you may need or want to provide the clients with options to use. To help users get started, or to control what users back up, you can

define sets of client options for clients to use. Client options sets are defined in the server database and are used by the clients that you designate.

Among the options that can be in a client option set are the include and exclude options. These options control which files are considered for the client operations.

For more information, see:

- Chapter 11, “Adding client nodes,” on page 367
- Chapter 12, “Managing client nodes,” on page 377

Allowing subfile backups

For mobile and remote users, you want to minimize the data sent over the network, as well as the time that they are connected to the network. You can set the server to allow a client node to back up changed portions of files that have been previously backed up, rather than entire files. The portion of the file that is backed up is called a *subfile*.

For more information, see Chapter 15, “Managing data for client nodes,” on page 495.

Creating backup sets for client nodes

You can perform an instant archive for a client by creating a backup set. A backup set copies a client node's active, backed-up files from server storage onto sequential media. If the sequential media can be read by a device available to the client system, you can restore the backup set directly to the client system without using the network. The server tracks backup sets that you create and retains the backup sets for the time you specify.

For more information, see Chapter 15, “Managing data for client nodes,” on page 495.

For more information on managing client nodes, see the *Backup-Archive Clients Installation and User's Guide*.

Security management

Tivoli Storage Manager includes security features for user registration and passwords. Also included are features that can help ensure security when clients connect to the server across a firewall.

Registration for clients can be closed or open. With closed registration, a user with administrator authority must register all clients. With open registration, clients can register themselves at first contact with the server. See “Registering nodes with the server” on page 368.

You can ensure that only authorized administrators and client nodes are communicating with the server by requiring the use of passwords. You can also set the following requirements for passwords:

- Number of characters in a password.
- Expiration time.
- A limit on the number of consecutive, invalid password attempts. When the client exceeds the limit, Tivoli Storage Manager locks the client node from access to the server.

See “Managing passwords and login procedures” on page 432.

You can control the authority of administrators. An organization may name a single administrator or may distribute the workload among a number of administrators and grant them different levels of authority. For details, see “Managing levels of administrative authority” on page 430.

For better security when clients connect across a firewall, you can control whether clients can initiate contact with the server for scheduled operations. See the “Managing client nodes across a firewall” on page 378 for details.

Several server options allow you to keep client and administrative traffic on separate server ports.

For additional ways to manage security, see Chapter 13, “Managing IBM Tivoli Storage Manager security,” on page 421.

Adding administrators

If you have installed any additional administrative clients, you should register them and grant an authority level to each.

For example, register an administrator with a user ID of MARK and a password of MISSOURI and grant system authority:

```
register admin mark missouri  
grant authority mark classes=system
```

Attention: The user ID SERVER_CONSOLE cannot be used by another administrator.

See “Managing IBM Tivoli Storage Manager administrators” on page 428 and Chapter 12, “Managing client nodes,” on page 377 for more information.

Adding backup-archive clients

If you installed additional backup-archive clients you should register them.

For example, register a node named MERCEDES with the password MONTANA:

```
register node mercedes montana userid=none
```

Two commands, DEFINE CLOPTSET and DEFINE CLIENTOPT, allow you to define client options affecting backup, archive, restore, and retrieval operations.

Managing client data with policies

As the administrator, you define the rules for client backup, archive, and migration operations, based on user or business requirements.

The rules are called *policies*. Policies identify:

- The criteria for backup, archive, and migration of client data
- Where the client data is initially stored
- How the data is managed by the server (how many backup versions are kept, for how long)

In Tivoli Storage Manager, you define policies by defining policy domains, policy sets, management classes, and backup and archive copy groups. When you install Tivoli Storage Manager, you have a default policy that consists of a single policy domain named STANDARD.

The default policy provides basic backup protection for end-user workstations. To provide different levels of service for different clients, you can add to the default policy or create new policy. For example, because of business needs, file servers are likely to require a policy different from policy for users' workstations. Protecting data for applications such as Lotus Domino also may require a unique policy.

For more information about the default policy and establishing and managing new policies, see Chapter 14, "Implementing policies for client data," on page 435.

Creating new policies

The Tivoli Storage Manager default policy specifies how Tivoli Storage Manager manages client files.

For example, it specifies that Tivoli Storage Manager retains up to two backup versions of any file that exists on the client (see Chapter 14, "Implementing policies for client data," on page 435 for details). Two versions may be enough for most clients. However, if some clients need the last ten versions to be kept, you can do either of the following:

- Create a new policy domain and assign these clients to that domain (described in this section).
- Create a new management class within the default policy domain. The include-exclude lists for all the affected clients must now be updated.

Remember: Under the default policy, client files are stored directly to disk. You can also define policies for storing client files directly to tape. In a copy group, simply name a tape pool as the destination. However, if you store directly to tape, the number of available tape drives limits the number of client nodes that can store data at the same time.

To create a new policy, you can start by copying the policy domain, STANDARD. This operation also copies the associated policy set, management class, and copy groups. You then assign clients to the new domain.

1. Copy the default policy domain, STANDARD, to the new policy domain, NEWDOMAIN:

```
copy domain standard newdomain
```

This operation copies the policy domain, and all associated policy sets, management classes, and copy groups. Within the policy domain named NEWDOMAIN and the policy set named STANDARD, you have:

- Management class named STANDARD
- Backup copy group named STANDARD
- Archive copy group named STANDARD

In this example, you update only the backup copy group.

2. Update the backup copy group by specifying that ten versions of backed up files are to be kept:

```
update copygroup newdomain standard standard standard -  
type=backup verexists=10
```

3. Validate and activate the STANDARD policy set in NEWDOMAIN:

```
validate policysset newdomain standard  
activate policysset newdomain standard
```

Important: The following conditions result in warning messages during validation:

- A destination storage pool is not a defined storage pool.

- The default management class does not contain a backup or archive copy group.
 - The current ACTIVE policy set names a management class that is not defined in the policy set being validated.
 - The current ACTIVE policy set contains copy groups that are not defined in the named policy set.
 - A management class specifies that a backup version must exist before a file can be migrated from a client node, but the management class does not contain a backup copy group.
4. Assign client nodes to the NEWDOMAIN policy domain by either updating existing client nodes or registering new nodes. For example, to update client node **mercedes**:

```
update node mercedes domain=newdomain
```

Because it is assigned to the NEWDOMAIN policy domain, Tivoli Storage Manager keeps up to ten versions of backed up files for client node **mercedes**.

Remember: If you associate a client that is currently running with a new domain, the client must be stopped and restarted before the new association will take effect.

For more information about the default policy and establishing and managing new policies, see Chapter 14, “Implementing policies for client data,” on page 435.

Schedules for client operations

Scheduling client operations can mean better protection for data, because operations can occur consistently without user intervention.

Scheduling also can mean better utilization of resources such as the network. Client backups that are scheduled at times of lower usage can minimize the impact on user operations on a network.

You can automate operations for clients by using schedules. Tivoli Storage Manager provides a central scheduling facility. You can also use operating system utilities or other scheduling tools to schedule Tivoli Storage Manager operations.

With Tivoli Storage Manager schedules, you can perform the operations for a client immediately or schedule the operations to occur at regular intervals.

The key objects that interact are:

Include-exclude options on each client

The include-exclude options determines which files are backed up, archived, or space-managed, and determines management classes, encryption, and type of backup for files.

The client can specify a management class for a file or group of files, or can use the default management class for the policy domain. The client specifies a management class by using an INCLUDE option in the client's include-exclude list or file. You can have central control of client options such as INCLUDE and EXCLUDE by defining client option sets on the server. When you register a client, you can specify a client option set for that client to use. See “Managing client option files” on page 411 for details.

Association defined between client and schedule

Associations determine which schedules are run for a client.

Clients are assigned to a policy domain when they are registered. To automate client operations, you define schedules for a domain. Then you define associations between schedules and clients in the same domain.

Schedule

The schedule determines when a client operation automatically occurs.

Schedules that can automate client operations are associated with a policy domain.

The scheduled client operations are called *events*. The Tivoli Storage Manager server stores information about events in its database. For example, you can query the server to determine which scheduled events completed successfully and which failed.

Management class

The management class determines where client files are initially stored and how they are managed.

The management class contains information that determines how Tivoli Storage Manager handles files that clients backup, archive, or migrate. For example, the management class contains the backup copy group and the archive copy group. Each copy group points to a *destination*, a storage pool where files are first stored when they are backed up or archived.

For a schedule to work on a particular client, the client machine must be turned on. The client either must be running the client scheduler or must allow the client acceptor daemon to start the scheduler when needed.

Client include-exclude lists

Any client can exclude some files from some policy operations and include other files in other operations.

This is done with statements in an include-exclude list or, on UNIX and Linux clients, in an include-exclude file. For example, an include-exclude file should exclude system files that, if recovered, could corrupt the operating system. Tivoli Storage Manager server and client directories should also be excluded. See the appropriate Tivoli Storage Manager client user's guide for details.

You can define include-exclude statements for your installation. Users can add these statements in their client options file (*dsm.sys*). You can also enter the statements in a set of options and assign that set to client nodes when you register or update the nodes. For details about the `DEFINE CLOPTSET` and `DEFINE CLIENTOPT` commands, see Chapter 12, "Managing client nodes," on page 377 and the *Administrator's Reference*.

Here are a few examples of include-exclude statements:

- A user wants all **.sct* and **.drw* files in the */eng/spec/* directory included for backup but all other files in that directory excluded. The user adds the following include-exclude statements:

```
exclude /eng/spec/*.*
include /eng/spec/*.drw
include /eng/spec/*.sct
```

Tivoli Storage Manager reads the statements from the bottom up until a match is found. In the preceding example, no match would be found on the include

statements for the file `/eng/spec/proto.obj`. Tivoli Storage Manager reads the exclude statement, finds a match, and excludes the file.

- For a file or group of files, the user can also override the default management class:

```
exclude /eng/spec/*.  
include /eng/spec/*.drw monthly  
include /eng/spec/*.sct
```

In this example,

- `*.sct` files are bound to the default management class.
- `*.drw` files are bound to the management class **monthly**.
- All other files in the **spec** directory are excluded from backup or archive.

For more information, see “The include-exclude list” on page 448.

Automating client operations

You can schedule most client operations, such as backup, to begin automatically.

You can schedule the following on most clients:

- Tivoli Storage Manager backup, archive, restore, and retrieve operations
- Operating system commands
- Macros (files that can contain operating system and Tivoli Storage Manager commands)

This section guides you through scheduling client backups for three registered client nodes assigned to the STANDARD policy domain: **bill**, **mark**, and **mercedes**.

1. Schedule an incremental backup and associate the schedule with the clients.

```
define schedule standard daily_incr action=incremental -  
    starttime=23:00  
  
define association standard daily_incr bill,mark,mercedes
```

The schedule, named DAILY_INCR, is for the Tivoli Storage Manager default policy domain, named STANDARD. The default specifies backup to the disk storage pool BACKUPPOOL. This schedule calls for a schedule window that:

- Begins on the date the schedule is defined (the default) at 11:00 p.m.
 - Lasts for 1 hour (the default)
 - Is repeated daily (the default)
 - Stays in effect indefinitely (the default)
2. Start the client scheduler. For the schedules to become active for a workstation, a user must start the scheduler from the node.

```
dsmc schedule
```

To help ensure that the scheduler is running on the clients, start the client acceptor daemon (CAD) or client acceptor service.

The include-exclude list (file on UNIX and Linux clients) on each client also affects which files are backed up or archived by the two schedules defined in the preceding steps. For example, if a file is excluded from backup with an EXCLUDE statement, the file will not be backed up when the DAILY_INCR schedule runs.

3. Because the DAILY_INCR schedule is to run daily, you can verify that it is working as it should on the day after you define the schedule and associate it with clients. If the schedule has run successfully, the status will be *Completed*.
query event standard daily_incr begindate=today-1

You can limit the query of events to display only schedules that did not run as expected. For example, you can use the following command daily to see which clients did not run the DAILY_INCR schedule the previous day:

```
query event standard daily_incr begindate=today-1 -  
exceptionsonly=yes
```

Schedules that did not complete successfully have a status of *Failed*, *Missed*, or *Severed*.

4. Check the results of the schedule DAILY_INCR on one of the clients that was associated with that schedule. For most clients, information about what happens when a schedule runs is stored in the file dsmsched.log. See the Backup-Archive Clients Installation and User's Guide.

Server maintenance

If you manage more than one server, you can ensure that the multiple servers are consistently managed by using the enterprise management functions of Tivoli Storage Manager.

You can set up one server as the configuration manager and have other servers obtain configuration information from it.

To keep the server running well, you can perform these tasks:

- Managing server operations, such as controlling client access to the server
- Automating repetitive administrative tasks
- Monitoring and adjusting space for the database and the recovery log
- Monitoring the status of the server, server storage, and clients

Server-operation management

When managing your server operations, you can choose from a variety of associated tasks.

Some of the more common tasks that you can perform to manage your server operations are shown in the following list:

- Start and stop the server.
- Allow and suspend client sessions with the server.
- Query, cancel, and preempt server processes such as backing up the server database.
- Customize server options.

Other tasks that are needed less frequently include:

- Maintain compliance with the license agreement.
- Move the server.

See "Licensing IBM Tivoli Storage Manager" on page 555. For suggestions about the day-to-day tasks required to administer the server, see Chapter 19, "Managing server operations," on page 555.

Server script automation

Repetitive, manual tasks associated with managing the server can be automated through Tivoli Storage Manager schedules and scripts. Using schedules and scripts can minimize the daily tasks for administrators.

You can define schedules for the automatic processing of most administrative commands. For example, a schedule can run the command to back up the server's database every day.

Tivoli Storage Manager server scripts allow you to combine administrative commands with return code checking and processing. The server comes with scripts that you can use to do routine tasks, or you can define your own. The scripts typically combine several administrative commands with return code checking, or run a complex SQL SELECT command.

For more information about automating Tivoli Storage Manager operations, see Chapter 20, "Automating server operations," on page 571.

Modifying a maintenance script

You can modify your maintenance script to add, subtract, or reposition commands.

If you have a predefined maintenance script, you can add or subtract commands using the maintenance script wizard. You can add, subtract, or reposition commands if you have a custom maintenance script. Both methods can be accessed through the same process. If you want to convert your predefined maintenance script to a custom maintenance script, select a server with the predefined script, click **Select Action** → **Convert to Custom Maintenance Script**.

Perform the following tasks to modify a maintenance script:

1. Click **Server Maintenance** in the navigation tree.
2. Select a server that has either **Predefined** or **Custom** designated in the **Maintenance Script** column.
3. Click **Select Action** → **Modify Maintenance Script**. If you are modifying a predefined maintenance script, the maintenance script wizard opens your script for you to modify. If you are modifying a custom maintenance script, the maintenance script editor opens your script so that you can modify it.

Database and recovery-log management

The Tivoli Storage Manager database contains information about registered client nodes, policies, schedules, and the client data in storage pools. The database is key to the operation of the server.

The information about the client data, also called *metadata*, includes the file name, file size, file owner, management class, copy group, and location of the file in server storage. The server records changes made to the database (database transactions) in its recovery log. The recovery log is used to maintain the database in a transactionally consistent state, and to maintain consistency across server startup operations.

For more information about the Tivoli Storage Manager database and recovery log and about the tasks associated with them, see Chapter 21, "Managing the database and recovery log," on page 593.

Extending the database size

The database size is largely determined by the number of client files to be stored on server storage. As you add clients, you may need to increase the database size.

You can increase the size of the database by creating new directories and adding them to the database space.

Sources of information about the server

Tivoli Storage Manager provides you with many sources of information about server and client status and activity, the state of the server's database and storage, and resource usage. By monitoring selected information, you can provide reliable services to users while making the best use of available resources.

The Administration Center includes a health monitor, which presents a view of the overall status of multiple servers and their storage devices. From the health monitor, you can link to details for a server, including a summary of the results of client schedules and a summary of the availability of storage devices. See Chapter 18, "Managing servers with the Administration Center," on page 547.

You can use Tivoli Storage Manager queries and SQL queries to get information about the server. You can also set up automatic logging of information about Tivoli Storage Manager clients and server events. Daily checks of some indicators are suggested.

See the following sections for more information about these tasks.

- Chapter 22, "Monitoring the Tivoli Storage Manager server," on page 613
- "Using SQL to query the IBM Tivoli Storage Manager database" on page 618
- "Logging IBM Tivoli Storage Manager events to receivers" on page 626
- "Daily monitoring scenario" on page 646

Tivoli Storage Manager server networks

You might have a number of Tivoli Storage Manager servers in your network, at the same or different locations.

Some examples of different configurations are:

- Your users are scattered across many locations, so you have located Tivoli Storage Manager servers close to the users to manage network bandwidth limitations.
- You have set up multiple servers to provide services to different organizations at one location.
- You have multiple servers on your network to make disaster recovery easier.

Servers connected to a network can be centrally managed. Tivoli Storage Manager provides functions to help you configure, manage, and monitor the servers. An administrator working at one Tivoli Storage Manager server can work with servers at other locations around the world.

When you have a network of Tivoli Storage Manager servers, you can simplify configuration and management of the servers by using enterprise administration functions. You can do the following:

- Designate one server as a configuration manager that distributes configuration information such as policy to other servers. See "Setting up enterprise configurations" on page 685.

- Route commands to multiple servers while logged on to one server. See “Routing commands” on page 707.
- Log events such as error messages to one server. This allows you to monitor many servers and clients from a single server. See “Enterprise event logging: logging events to another server” on page 639.
- Store data for one Tivoli Storage Manager server in the storage of another Tivoli Storage Manager server. The storage is called server-to-server virtual volumes. See “Using virtual volumes to store data on another server” on page 712 for details.
- Share an automated library among Tivoli Storage Manager servers. See “Devices on storage area networks” on page 52.
- Store a recovery plan file for one server on another server, when using disaster recovery manager. You can also back up the server database and storage pools to another server. See Chapter 26, “Using disaster recovery manager,” on page 801 for details.
- Back up the server database and storage pools to another server. See “Using virtual volumes to store data on another server” on page 712 for details.

Exporting and importing data

As conditions change, you can move data from one server to another by using export and import processes.

For example, you may need to balance workload among servers by moving client nodes from one server to another. The following methods are available:

- You can export part or all of a server's data to sequential media, such as tape or a file on hard disk. You can then take the media to another server and import the data to that server
- You can export part or all of a server's data and import the data directly to another server, if server-to-server communications are set up.

For more information about moving data between servers, see Chapter 24, “Exporting and importing data,” on page 721.

Protecting Tivoli Storage Manager and client data

The database, recovery log, and storage pools are critical to the operation of the server and must be properly protected.

Attention: If the database is unusable, the entire Tivoli Storage Manager server is unavailable. If a database is lost and cannot be recovered, it might be difficult or impossible to recover data managed by that server. Therefore, It is critically important to back up the database. However, even without the database, fragments of data or complete files might easily be read from storage pool volumes that are not encrypted. Even if data is not completely recovered, security can be compromised. For this reason, sensitive data should always be encrypted by the Tivoli Storage Manager client or the storage device, unless the storage media is physically secured. See Part 5, “Protecting the server,” on page 753 for steps that you can take to protect your database.

IBM Tivoli Storage Manager provides a number of ways to protect your data, including backing up your storage pools and database. For example, you can define schedules so that the following operations occur:

- After the initial full backup of your storage pools, incremental storage pool backups are done nightly.
- Full database backups are done weekly.
- Incremental database backups are done nightly.

In addition, disaster recovery manager (DRM), an optional feature of Tivoli Storage Manager, can assist you in many of the tasks that are associated with protecting and recovering your data. For details, see:

Chapter 26, “Using disaster recovery manager,” on page 801

Protecting the server

Tivoli Storage Manager provides a number of ways to protect and recover your server from media failure or from the loss of the Tivoli Storage Manager database or storage pools.

Recovery is based on the following preventive measures:

- Mirroring, by which the server maintains a copy of the active log
- Periodic backup of the database
- Periodic backup of the storage pools
- Audit of storage pools for damaged files, and recovery of damaged files when necessary
- Backup of the device configuration and volume history files
- Validation of the data in storage pools, using cyclic redundancy checking

For information about protecting the server with these measures, see Chapter 25, “Protecting and recovering your server,” on page 755.

You can also create a maintenance script to perform database and storage pool backups through the Server Maintenance work item in the Administration Center. See Chapter 18, “Managing servers with the Administration Center,” on page 547 for details.

In addition to taking these actions, you can prepare a disaster recovery plan to guide you through the recovery process by using the disaster recovery manager, which is available with Tivoli Storage Manager Extended Edition. The disaster recovery manager (DRM) assists you in the automatic preparation of a disaster recovery plan. You can use the disaster recovery plan as a guide for disaster recovery as well as for audit purposes to certify the recoverability of the Tivoli Storage Manager server.

The disaster recovery methods of DRM are based on taking the following measures:

- Sending server backup volumes offsite or to another Tivoli Storage Manager server
- Creating the disaster recovery plan file for the Tivoli Storage Manager server
- Storing client machine information
- Defining and tracking client recovery media

For more information about protecting your server and for details about recovering from a disaster, see Chapter 25, “Protecting and recovering your server,” on page 755.

Part 2. Configuring and managing server storage

Initially, you must attach devices to the server and then create objects that represent those devices. You also create objects representing storage resources, such as storage pools and storage-pool volumes. A wide variety of Tivoli Storage Manager functions, such as tape reclamation and simultaneous-write operations, are available to manage client data and to control and optimize server storage.

Chapter 3. Storage device concepts

To work with storage devices, you must be familiar with Tivoli Storage Manager storage objects and other basic concepts.

"Tivoli Storage Manager storage devices" on page 42
"Tivoli Storage Manager storage objects" on page 42
"Tivoli Storage Manager volumes" on page 50
"Planning for server storage" on page 65
"Device configurations" on page 52
"Removable media mounts and dismounts" on page 59
"How Tivoli Storage Manager uses and reuses removable media" on page 60
"Required definitions for storage devices" on page 63

The examples in topics show how to perform tasks using the Tivoli Storage Manager command-line interface. For information about the commands, see the *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

You can also perform Tivoli Storage Manager tasks from the Administration Center. For more information about using the Administration Center, see Chapter 18, "Managing servers with the Administration Center," on page 547.

Road map for key device-related task information

Key tasks include configuring and managing disk devices, physically attaching storage devices to your system, and so on. In this document, information about tasks is organized into linked topics.

Use the following table to identify key tasks and the topics that describe how to perform those tasks.

Task	Topic
Configure and manage magnetic disk devices, which Tivoli Storage Manager uses to store client data, the database, database backups, recovery log, and export data.	Chapter 4, "Magnetic disk devices," on page 69
Physically attach storage devices to your system. Install and configure the required device drivers.	Chapter 5, "Using devices with the server system," on page 79
Configure devices to use with Tivoli Storage Manager, using detailed scenarios of representative device configurations.	Chapter 6, "Configuring storage devices," on page 89
Plan, configure, and manage an environment for NDMP operations	Chapter 8, "Using NDMP for operations with NAS file servers," on page 167
Perform routine operations such as labeling volumes, checking volumes into automated libraries, and maintaining storage volumes and devices.	Chapter 7, "Managing removable media operations," on page 133

Task	Topic
Define and manage device classes.	Chapter 9, "Defining device classes," on page 199

Tivoli Storage Manager storage devices

With Tivoli Storage Manager, you can use a range of manual and automated devices for server storage. Both direct and network-attached storage provide options for storing data. Tivoli Storage Manager devices can be physical, such as disk drives and tape drives, or logical, such as files on disk or storage on another server.

Tivoli Storage Manager supports the following types of devices:

- Tape devices
 - Optical disk devices
 - Storage area network (SAN) devices
- Devices in a SAN environment must be supported by the Tivoli Storage Manager server.

For a summary of supported devices, see Table 8 on page 63. For details and updates, see the Tivoli Storage Manager device support Web site:

http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_Linux.html

Tivoli Storage Manager storage objects

Devices and media are represented by objects that you define. Information about these objects is stored in the Tivoli Storage Manager database.

You can query, update, and delete the following objects:

- Library
- Drive
- Device class
- Storage pool
- Storage pool volume
- Data mover
- Path
- Server

Libraries

A physical library is a collection of one or more drives that share similar media-mounting requirements. That is, the drive can be mounted by an operator or by an automated mounting mechanism.

A library object definition specifies the library type, for example, SCSI or 349X, and other characteristics associated with the library type, for example, the category numbers used by an IBM TotalStorage® 3494 Tape Library for private, scratch volumes, and scratch, write-once, read-many (WORM) volumes.

Tivoli Storage Manager supports a variety of library types.

Shared libraries

Shared libraries are logical libraries that are represented physically by SCSI, 349X, or ACSLS libraries. The physical library is controlled by the Tivoli Storage Manager server configured as a library manager. Tivoli Storage Manager servers using the SHARED library type are library clients to the library manager server. Shared libraries reference a library manager.

Optical devices are not supported for library sharing.

Automated cartridge system library software libraries

An automated cartridge system library software (ACSL) library is a type of external library that is controlled by the Sun StorageTek ACSLS media-management software. The server can act as a client application to the ACSLS software to use the drives.

The Sun StorageTek software performs the following functions:

- Volume mounts (specific and scratch)
- Volume dismounts
- Freeing of library volumes (return to scratch)

The ACSLS software selects the appropriate drive for media-access operations. You do not define the drives, check in media, or label the volumes in an external library.

For additional information regarding ACSLS libraries, refer to the Sun StorageTek documentation.

Manual libraries

In manual libraries, operators mount the volumes in response to mount-request messages issued by the server.

The server sends these messages to the server console and to administrative clients that were started by using the special **MOUNTMODE** or **CONSOLEMODE** parameter.

You can also use manual libraries as logical entities for sharing sequential-access disk (FILE) volumes with other servers.

You cannot combine drives of different types or formats, such as Digital Linear Tape (DLT) and 8MM, in a single manual library. Instead, you must create a separate manual library for each device type.

For information about configuring a manual library, see:

Chapter 6, "Configuring storage devices," on page 89

For information about monitoring mount messages for a manual library, see:

"Mount operations for manual libraries" on page 152

SCSI libraries

A SCSI library is controlled through a SCSI interface, attached either directly to the server's host using SCSI cabling or by a storage area network. A robot or other mechanism automatically handles volume mounts and dismounts.

The drives in a SCSI library can be of different types. A SCSI library can contain drives of mixed technologies, for example LTO Ultrium and DLT drives. Some examples of this library type are:

- The Sun StorageTek L700 library
- The IBM 3590 tape device, with its Automatic Cartridge Facility (ACF)

Remember: Although it has a SCSI interface, the IBM 3494 Tape Library Dataserver is defined as a 349X library type.

For information about configuring a SCSI library, see:

Chapter 6, "Configuring storage devices," on page 89

349X libraries

A 349X library is a collection of drives in an IBM 3494. Volume mounts and demounts are handled automatically by the library. A 349X library has one or more library management control points (LMCP) that the server uses to mount and dismount volumes in a drive. Each LMCP provides an independent interface to the robot mechanism in the library.

The drives in a 3494 library must be of one type only (either IBM 3490, 3590, or 3592).

For information about configuring a 349X library, see:

Chapter 6, "Configuring storage devices," on page 89

External libraries

An external library is a collection of drives managed by an external media-management system that is not part of Tivoli Storage Manager. The server provides an interface that allows external media management systems to operate with the server.

The external media-management system performs the following functions:

- Volume mounts (specific and scratch)
- Volume dismounts
- Freeing of library volumes (return to scratch)

The external media manager selects the appropriate drive for media-access operations. You do not define the drives, check in media, or label the volumes in an external library.

An external library allows flexibility in grouping drives into libraries and storage pools. The library can have one drive, a collection of drives, or even a part of an automated library.

An ACSLS or LibraryStation-controlled Sun StorageTek library used in conjunction with an external library manager (ELM), like Gresham's EDT-DistribuTAPE, is a type of external library.

For a definition of the interface that Tivoli Storage Manager provides to the external media management system, see Appendix A, “External media management interface description,” on page 859.

Drives

A drive object represents a drive mechanism within a library that uses removable media. For devices with multiple drives, including automated libraries, you must define each drive separately and associate it with a library.

Drive definitions can include such information as the element address (for drives in SCSI libraries), how often the drive is cleaned (for tape drives), and whether or not the drive is online.

Tivoli Storage Manager drives include tape drives that can stand alone or that can be part of an automated library.

Device class

Each device that is defined to Tivoli Storage Manager is associated with one device class, which specifies the device type and media management information, such as recording format, estimated capacity, and labeling prefixes.

A device type identifies a device as a member of a group of devices that share similar media characteristics. For example, the 8MM device type applies to 8-mm tape drives.

Device types include a variety of removable media types as well as FILE and SERVER.

A device class for a tape or optical drive must also specify a library.

Disk devices

Using Tivoli Storage Manager, you can define random-access disk (DISK device type) volumes using a single command. You can also use space triggers to automatically create preassigned private volumes when predetermined space-utilization thresholds are exceeded.

For important disk-related information, see “Requirements for disk subsystems” on page 69.

Removable media

Tivoli Storage Manager provides a set of specified removable-media device types, such as 8MM for 8 mm tape devices, or REMOVABLEFILE for Jaz or DVD-RAM drives.

For more information about supported removable media device types, see Chapter 9, “Defining device classes,” on page 199 and the *Administrator’s Reference*.

Files on disk as sequential volumes (FILE)

The FILE device type lets you create sequential volumes by creating files on disk storage. To the server, these files have the characteristics of a tape volume. FILE volumes can also be useful when transferring data for purposes such as electronic vaulting or for taking advantage of relatively inexpensive disk storage devices.

FILE volumes are a convenient way to use sequential-access disk storage for the following reasons:

- You do not need to explicitly define scratch volumes. The server can automatically acquire and define scratch FILE volumes as needed.
- You can create and format FILE volumes using a single command. The advantage of private FILE volumes is that they can reduce disk fragmentation and maintenance overhead.
- Using a single device class definition that specifies two or more directories, you can create large, FILE-type storage pools. Volumes are created in the directories you specify in the device class definition. For optimal performance, volumes should be associated with file systems.
- When predetermined space-utilization thresholds have been exceeded, space trigger functionality can automatically allocate space for private volumes in FILE-type storage pools.
- The Tivoli Storage Manager server allows concurrent read-access and write-access to a volume in a storage pool associated with the FILE device type. Concurrent access improves restore performance by allowing two or more clients to access the same volume at the same time. Multiple client sessions (archive, retrieve, backup, and restore) or server processes (for example, storage pool backup) can read the volume concurrently. In addition, one client session can write to the volume while it is being read.

The following server processes are allowed shared read access to FILE volumes:

- BACKUP DB
- BACKUP STGPOOL
- COPY ACTIVATEDATA
- EXPORT/IMPORT NODE
- EXPORT/IMPORT SERVER
- GENERATE BACKUPSET
- RESTORE STGPOOL
- RESTORE VOLUME

The following server processes are not allowed shared read access to FILE volumes:

- AUDIT VOLUME
- DELETE VOLUME
- MIGRATION
- MOVE DATA
- MOVE NODEDATA
- RECLAMATION

Unless sharing with storage agents is specified, the FILE device type does not require you to define library or drive objects. The only required object is a device class.

For important disk-related information, see “Requirements for disk subsystems” on page 69.

Sequential volumes on another Tivoli Storage Manager server (SERVER)

The SERVER device type lets you create volumes for one Tivoli Storage Manager server that exist as archived files in the storage hierarchy of another server. These virtual volumes have the characteristics of sequential-access volumes such as tape. No library or drive definition is required.

You can use virtual volumes for the following:

- Device-sharing between servers. One server is attached to a large tape library device. Other servers can use that library device indirectly through a SERVER device class.
- Data-sharing between servers. By using a SERVER device class to export and import data, physical media remains at the original location instead having to be transported.
- Immediate offsite storage. Storage pools and databases can be backed up without physically moving media to other locations.
- Offsite storage of the disaster recovery manager (DRM) recovery plan file.
- Electronic vaulting.

See “Using virtual volumes to store data on another server” on page 712.

Library, drive, and device-class objects

Library objects, drive objects, and device-class objects taken together represent physical storage entities.

These three objects are shown in Figure 2.

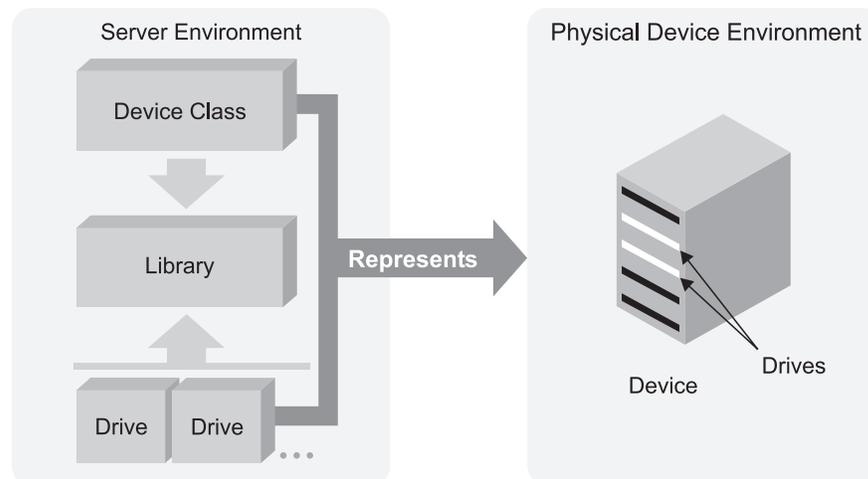


Figure 2. Removable media devices are represented by a library, drive, and device class

- For more information about the drive object, see:
 - “Managing drives” on page 156
 - “Defining drives” on page 127
- For more information about the library object, see:
 - “Managing libraries” on page 154
 - “Defining libraries” on page 126
- For more information about the device class object, see Chapter 9, “Defining device classes,” on page 199.

Storage pools and storage-pool volumes

A *storage pool* is a collection of volumes that are associated with one device class and one media type. For example, a storage pool that is associated with a device class for 8-mm tape volumes contains only 8 mm tape volumes.

You can control the characteristics of storage pools, such as whether scratch volumes are used.

Tivoli Storage Manager supplies default disk storage pools. .

Figure 3 shows storage pool volumes grouped into a storage pool. Each storage pool represents only one type of media. For example, a storage pool for 8-mm devices represents collections of only 8-mm tapes.

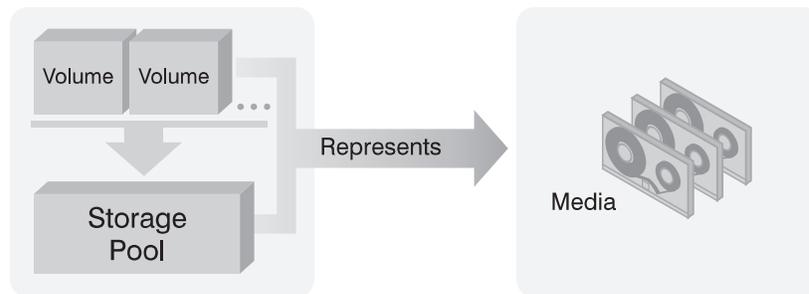


Figure 3. Relationships of storage pool volumes, storage pools, and media

For DISK device classes, you must define volumes. For other device classes, such as tape and FILE, you can allow the server to dynamically acquire scratch volumes and define those volumes as needed. For details, see:

“Preparing volumes for random-access storage pools” on page 236

“Preparing volumes for sequential-access storage pools” on page 236

One or more device classes are associated with one *library*, which can contain multiple drives. When you define a storage pool, you associate the pool with a device class. Volumes are associated with pools. Figure 4 on page 49 shows these relationships.

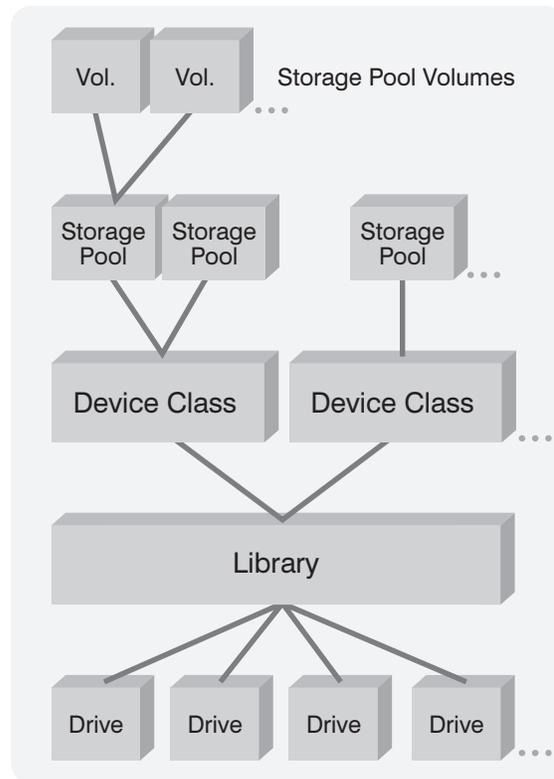


Figure 4. Relationships between storage and device objects

For information about defining storage pool and volume objects, see Chapter 10, “Managing storage pools and volumes,” on page 221.

For information about configuring volumes for random access see “Configuring random access volumes on disk devices” on page 75.

Data movers

Data movers are devices that accept requests from Tivoli Storage Manager to transfer data on behalf of the server. Data movers transfer data between storage devices without using significant server, client, or network resources.

For NDMP operations, data movers are NAS file servers. The definition for a NAS data mover contains the network address, authorization, and data formats required for NDMP operations. A data mover enables communication and ensures authority for NDMP operations between the Tivoli Storage Manager server and the NAS file server.

Tivoli Storage Manager supports two types of data movers:

- For NDMP operations, data movers are NAS file servers. The definition for a NAS data mover contains the network address, authorization, and data formats required for NDMP operations. A data mover enables communication and ensures authority for NDMP operations between the Tivoli Storage Manager server and the NAS file server.
- For server-free data movement, data movers are devices such as the IBM SAN Data Gateway, that move data between disk devices and tape devices on the SAN.

Paths

Paths allow access to drives, disks, and libraries. A path definition specifies a source and a destination. The source accesses the destination, but data can flow in either direction between the source and destination.

Here are a few examples of paths:

- Between a server and a drive or a library
- Between a storage agent and a drive
- Between a data mover and a drive, a disk, or a library

For more information about the path object, see:

“Defining paths” on page 129

“Managing paths” on page 164

Server objects

Server objects are defined to use a library that is on a SAN and that is managed by another Tivoli Storage Manager server, to use LAN-free data movement, or to store data in virtual volumes on a remote server.

Among other characteristics, you must specify the server TCP/IP address.

For more information, see:

- “Setting up the library client servers” on page 98
- “Using virtual volumes to store data on another server” on page 712
- *Storage Agent User’s Guide*

Tivoli Storage Manager volumes

A *volume* is the basic unit of storage for Tivoli Storage Manager storage pools. Tivoli Storage Manager volumes are classified according to status: private, scratch, and scratch write-once, read-many (WORM). Scratch WORM status applies to 349X libraries only when the volumes are IBM 3592 WORM volumes.

The following definitions apply:

- A private volume is a labeled volume that is in use or owned by an application, and may contain valid data. You must define each private volume. Alternatively, for storage pools associated with sequential access disk (FILE) device classes, you can use space triggers to create private, preassigned volumes when predetermined space-utilization thresholds have been exceeded. Private FILE volumes are allocated as a whole. The result is less risk of severe fragmentation than with space dynamically acquired for scratch FILE volumes.

A request to mount a private volume must include the name of that volume. Defined private volumes do not return to scratch when they become empty. For information about defining private volumes, see “Defining storage pool volumes” on page 237. For information about changing the status of a volume (for example, from private to scratch) in an automated library, see the following:

- “Changing the status of a volume” on page 147
- A scratch volume is a labeled volume that is empty or contains no valid data and that can be used to satisfy any request to mount a scratch volume. When data is written to a scratch volume, its status is changed to private, and it is defined as part of the storage pool for which the mount request was made.

When valid data is moved from the volume and the volume is reclaimed, the volume returns to scratch status and can be reused by any storage pool associated with the library.

- A WORM scratch volume is similar to a conventional scratch volume. However, WORM volumes cannot be reclaimed by Tivoli Storage Manager reclamation processing. WORM volumes can be returned to scratch status only if they have empty space in which data can be written. Empty space is space that does not contain valid, expired or deleted data. (Deleted and expired data on WORM volumes cannot be overwritten.) If a WORM volume does not have any empty space in which data can be written (for example, if the volume is entirely full of deleted or expired data), the volume remains private.

For each storage pool, you must decide whether to use scratch volumes. If you do not use scratch volumes, you must define private volumes, or you can use space-triggers if the volume is assigned to a storage pool with a FILE device type. Tivoli Storage Manager keeps an inventory of volumes in each automated library it manages and tracks whether the volumes are in scratch or private status. When a volume mount is requested, Tivoli Storage Manager selects a scratch volume only if scratch volumes are allowed in the storage pool. The server can choose any scratch volume that has been checked into the library.

You do not need to allocate volumes to different storage pools associated with the same automated library. Each storage pool associated with the library can dynamically acquire volumes from the library's inventory of scratch volumes. Even if only one storage pool is associated with a library, you do not need to explicitly define all the volumes for the storage pool. The server automatically adds volumes to and deletes volumes from the storage pool.

Tip: A disadvantage of using scratch volumes is that volume usage information, which you can use to determine when the media has reached its end of life, is deleted when a private volume is returned to the scratch volume pool.

Volume inventory for an automated library

A library's volume inventory includes only those volumes that have been checked into that library.

This inventory is not necessarily identical to the list of volumes in the storage pools associated with the library. For example:

- A volume can be checked into the library but not be in a storage pool (a scratch volume, a database backup volume, or a backup set volume).
- A volume can be defined to a storage pool associated with the library (a private volume), but not checked into the library.

For more information on how to check in volumes, see the following:

- “Checking new volumes into a library” on page 137

Device configurations

You can configure devices on a local area network, on a storage area network, for LAN-free data movement, and as network-attached storage. Tivoli Storage Manager provides methods for configuring storage devices.

For information about supported devices and Fibre Channel hardware and configurations, see http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager

Devices on local area networks

In the conventional local area network (LAN) configuration, one or more tape or optical libraries are associated with a single Tivoli Storage Manager server.

In a LAN configuration, client data, electronic mail, terminal connection, application program, and device control information must all be handled by the same network. Device control information and client backup and restore data flow across the LAN.

Libraries cannot be partitioned or shared in a LAN environment. However, the 349X library has a limited ability to share 3590 drives or 3592 drives between more than one Tivoli Storage Manager server. For details, see “Sharing an IBM 3494 library by static partitioning of drives” on page 108 and “Sharing an IBM 3494 library among servers” on page 105.

For information on the categories of libraries supported by Tivoli Storage Manager, see “Libraries” on page 42.

Devices on storage area networks

A SAN is a dedicated storage network that can improve system performance. On a SAN you can consolidate storage and relieve the distance, scalability, and bandwidth limitations of LANs and wide area networks (WANs).

Using Tivoli Storage Manager in a SAN allows the following functions:

- Sharing storage devices among multiple Tivoli Storage Manager servers. For more information on sharing storage devices, see
 - “Configuring SCSI libraries shared among servers on a SAN” on page 95
- Allowing Tivoli Storage Manager clients, through a storage agent on the client machine, to move data directly to storage devices (LAN-free data movement).

In a SAN you can share tape drives, optical drives, and libraries that are supported by the Tivoli Storage Manager server, including most SCSI devices.

For information about device driver setup information, see Chapter 5, “Using devices with the server system,” on page 79.

Figure 5 on page 53 shows a SAN configuration in which two Tivoli Storage Manager servers share a library.

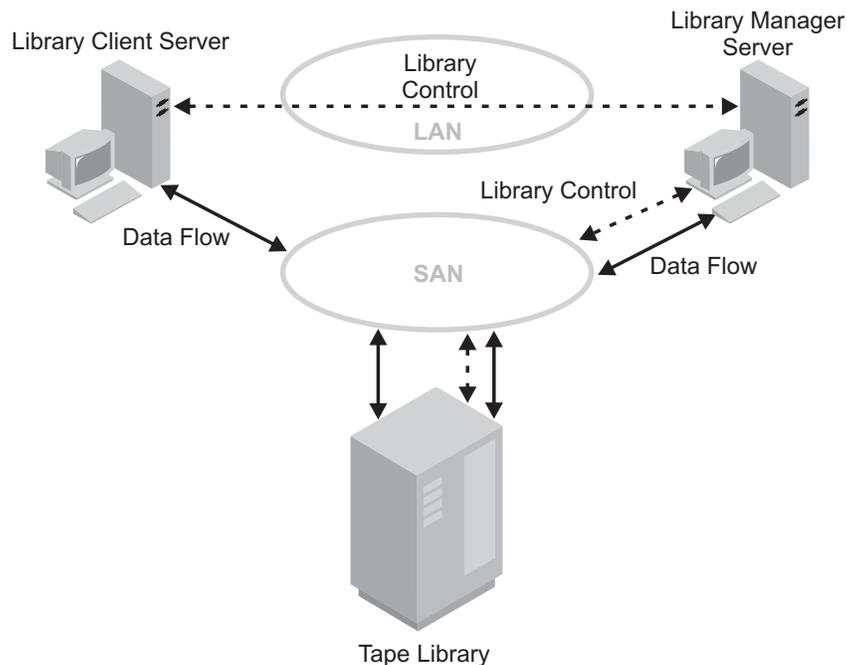


Figure 5. Library sharing in a storage area network (SAN) configuration. The servers communicate over the LAN. The library manager controls the library over the SAN. The library client stores data to the library devices over the SAN.

When Tivoli Storage Manager servers share a library, one server, the *library manager*, controls device operations. These operations include mount, dismount, volume ownership, and library inventory. Other Tivoli Storage Manager servers, *library clients*, use server-to-server communications to contact the library manager and request device service. Data moves over the SAN between each server and the storage device.

Tivoli Storage Manager servers use the following features when sharing an automated library:

Partitioning of the Volume Inventory

The inventory of media volumes in the shared library is partitioned among servers. Either one server owns a particular volume, or the volume is in the global scratch pool. No server owns the scratch pool at any given time.

Serialized Drive Access

Only one server accesses each tape drive at a time. Drive access is serialized and controlled so that servers do not dismount other servers' volumes or write to drives where other servers mount their volumes.

Serialized Mount Access

The library autochanger performs a single mount or dismount operation at a time. A single server (library manager) performs all mount operations to provide this serialization.

LAN-free data movement

Tivoli Storage Manager allows a client, through a storage agent, to directly back up and restore data to a tape library on a SAN.

Figure 6 shows a SAN configuration in which a client directly accesses a tape or FILE library to read or write data.

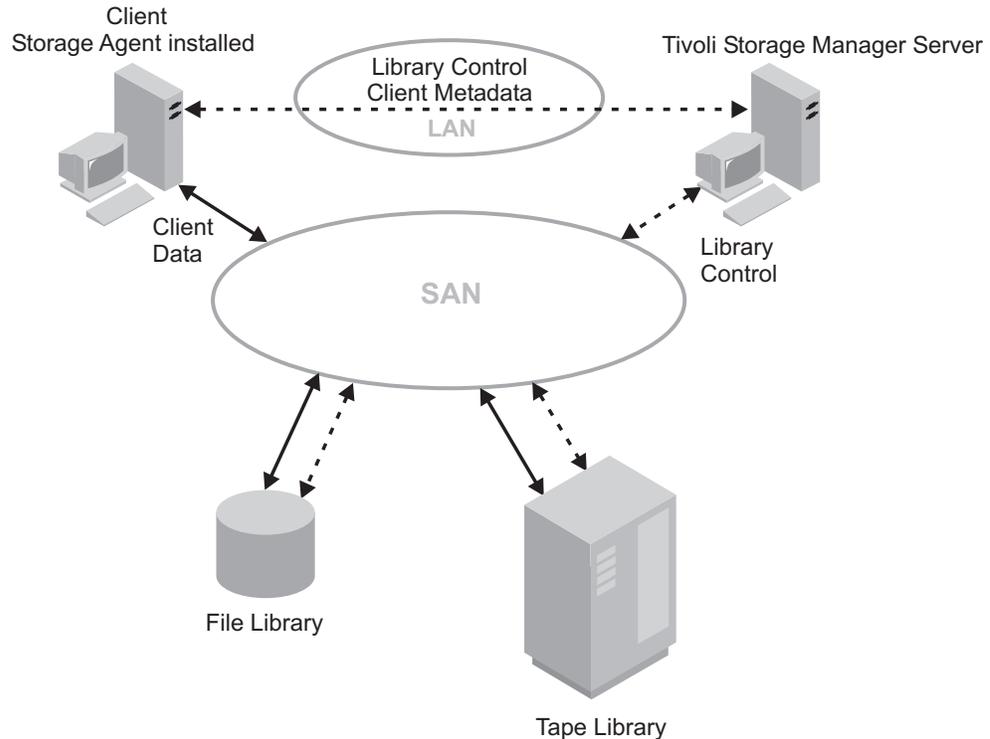


Figure 6. LAN-Free data movement. Client and server communicate over the LAN. The server controls the device on the SAN. Client data moves over the SAN to the device.

LAN-free data movement requires the installation of a storage agent on the client machine. The server maintains the database and recovery log, and acts as the library manager to control device operations. The storage agent on the client handles the data transfer to the device on the SAN. This implementation frees up bandwidth on the LAN that would otherwise be used for client data movement.

The following outlines a typical backup scenario for a client that uses LAN-free data movement:

1. The client begins a backup operation. The client and the server exchange policy information over the LAN to determine the destination of the backed up data. For a client using LAN-free data movement, the destination is a storage pool that uses a device on the SAN.
2. Because the destination is on the SAN, the client contacts the storage agent, which will handle the data transfer. The storage agent sends a request for a volume mount to the server.
3. The server contacts the storage device and, in the case of a tape library, mounts the appropriate media.
4. The server notifies the client of the location of the mounted media.

5. The client, through the storage agent, writes the backup data directly to the device over the SAN.
6. The storage agent sends file attribute information to the server, and the server stores the information in its database.

If a failure occurs on the SAN path, failover occurs. The client uses its LAN connection to the Tivoli Storage Manager server and moves the client data over the LAN.

Remember:

- For the latest information about clients that support the feature, see the IBM Tivoli Storage Manager support page at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.
- Optical devices cannot be targets for LAN-free operations.

Network-attached storage

Network-attached storage (NAS) file servers are dedicated storage machines whose operating systems are optimized for file-serving functions. NAS file servers typically do not run software acquired from another vendor. Instead, they interact with programs like Tivoli Storage Manager through industry-standard network protocols, such as network data management protocol (NDMP).

Tivoli Storage Manager provides two basic types of configurations that use NDMP for backing up and managing NAS file servers. In one type of configuration, Tivoli Storage Manager uses NDMP to back up a NAS file server to a library device directly attached to the NAS file server. (See Figure 7.) The NAS file server, which can be distant from the Tivoli Storage Manager server, transfers backup data directly to a drive in a SCSI-attached tape library. Data is stored in special, NDMP-formatted storage pools, which can be backed up to storage media that can be moved offsite for protection in case of an on-site disaster.

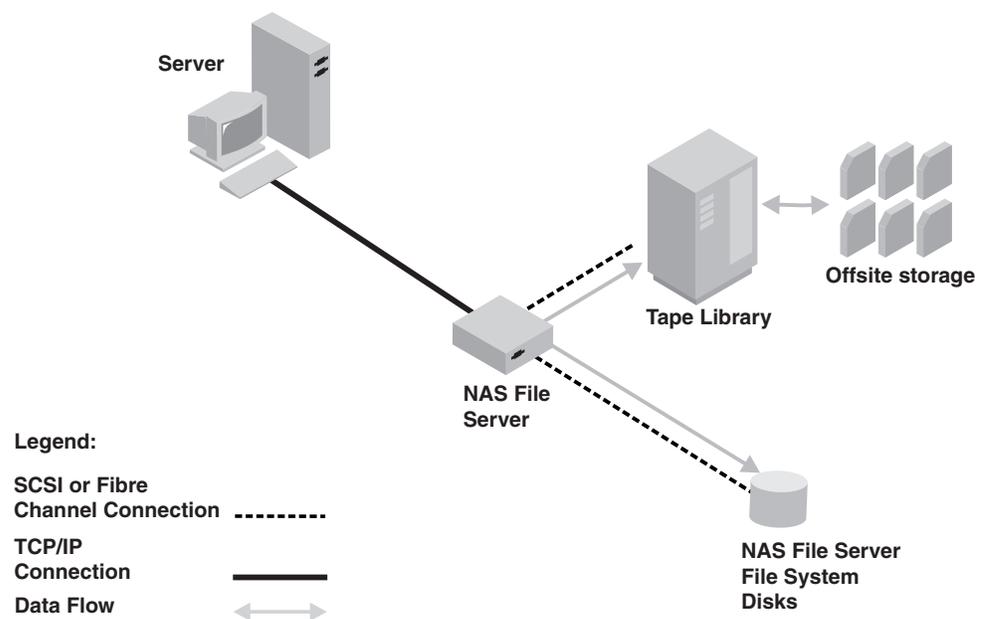


Figure 7. Library device directly attached to a NAS file server

In the other type of NDMP-based configuration, Tivoli Storage Manager uses NDMP to back up a NAS file server to a Tivoli Storage Manager storage-pool hierarchy. (See Figure 8.) With this type of configuration you can store NAS data directly to disk (either random access or sequential access) and then migrate the data to tape. Data can also be backed up to storage media that can then be moved offsite. The advantage of this type of configuration is that it gives you all the backend-data management features associated with a conventional Tivoli Storage Manager storage-pool hierarchy, including migration and reclamation.

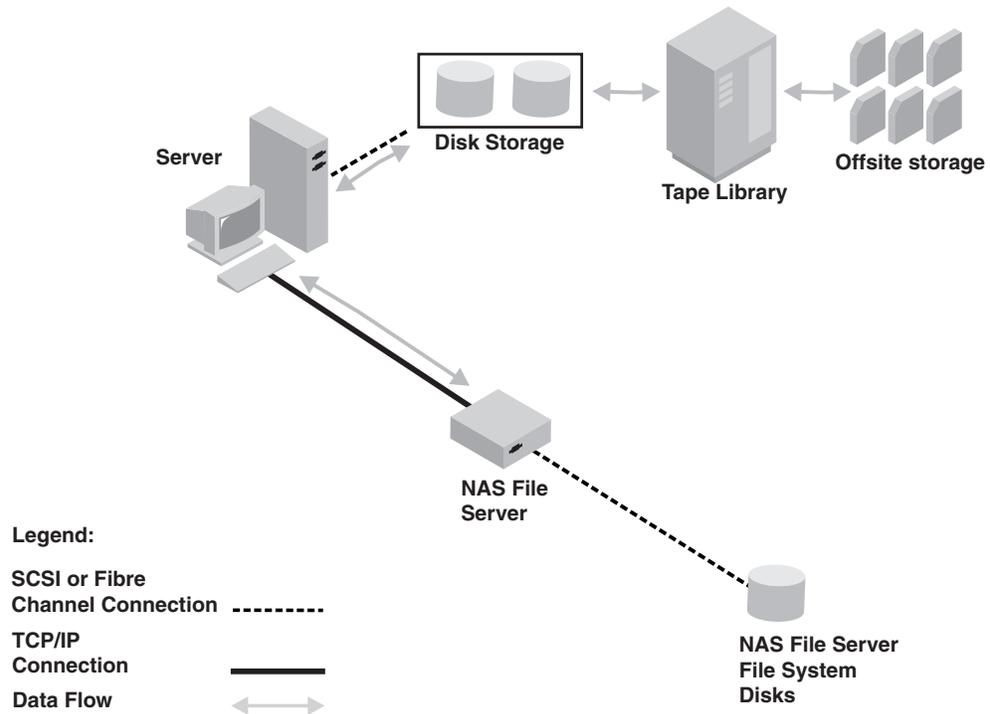


Figure 8. NAS file server to Tivoli Storage Manager storage-pool hierarchy

In both types of configurations, Tivoli Storage Manager tracks file-system image backups and has the capability to perform NDMP file-level restores. For more information regarding NDMP file-level restores, see “NDMP file-level restoration” on page 57.

Note:

- Support for filer-to-server data transfer is only available for NAS devices that support NDMP version 4.
- For a comparison of NAS backup methods, including using a backup-archive client to back up a NAS file server, see “Determining the location of NAS backup” on page 175.

NDMP backup operations

In backup images produced by network data management protocol (NDMP) operations for a NAS file server, Tivoli Storage Manager creates NAS file-system-level or directory-level image backups.

The image backups are different from traditional Tivoli Storage Manager backups because the NAS file server transfers the data to the drives in the library or directly to the Tivoli Storage Manager server. NAS file system image backups can be either full or differential image backups. The first backup of a file system on a NAS file server is always a full image backup. By default, subsequent backups are differential image backups containing only data that has changed in the file system since the last full image backup. If a full image backup does not already exist, a full image backup is performed.

If you restore a differential image, Tivoli Storage Manager automatically restores the full backup image first, followed by the differential image.

NDMP file-level restoration

Tivoli Storage Manager provides a way to restore data from backup images produced by NDMP operations. To assist users in restoring selected files, you can create a table of contents (TOC) of file-level information for each backup image.

Using the Web backup-archive client, users can then browse the TOC and select the files that they want to restore. If you do not create a TOC, users must be able to specify the name of the backup image that contains the file to be restored and the fully qualified name of the file.

You can create a TOC using one of the following commands:

- BACKUP NODE server command. For details, see the *Administrator's Reference*.
- BACKUP NAS client command, with `include.fs.nas` specified in the client options file or specified in the client options set. For details, see the *Backup-Archive Clients Installation and User's Guide*.

Directory-level backup and restore

If you have a large NAS file system, initiating a backup on a directory level reduces backup and restore times, and provides more flexibility in configuring your NAS backups.

By defining virtual file spaces, a file system backup can be partitioned among several NDMP backup operations and multiple tape drives. You can also use different backup schedules to back up sub-trees of a file system.

The virtual file space name cannot be identical to any file system on the NAS node. If a file system is created on the NAS device with the same name as a virtual file system, a name conflict will occur on the Tivoli Storage Manager server when the new file space is backed up. See the *Administrator's Reference* for more information about virtual file space mapping commands.

Remember: Virtual file space mappings are only supported for NAS nodes.

Mixed device types in libraries

Tivoli Storage Manager supports mixing different device types within a single automated library, as long as the library itself can distinguish among the different media for the different device types.

Libraries with this capability are those models supplied from the manufacturer already containing mixed drives, or capable of supporting the addition of mixed drives. Check with the manufacturer, and also check the Tivoli Storage Manager Web site for specific libraries that have been tested on Tivoli Storage Manager with mixed device types.

For example, you can have Quantum SuperDLT drives, LTO Ultrium drives, and StorageTek 9940 drives in a single library defined to the Tivoli Storage Manager server. For examples of how to set this up, see:

“Configuration with multiple drive device types” on page 93

“Configuring a 3494 library with multiple drive device types” on page 102

Different media generations in a library

While the Tivoli Storage Manager server now allows mixed device types in an automated library, the mixing of different generations of the same type of drive is still not supported. New drives cannot write the older media formats, and old drives cannot read new formats.

If the new drive technology cannot write to media formatted by older generation drives, the older media must be marked read-only to avoid problems for server operations. Also, the older drives must be removed from the library. Some examples of combinations that the Tivoli Storage Manager server does not support in a single library are:

- SDLT 220 drives with SDLT 320 drives
- DLT 7000 drives with DLT 8000 drives
- StorageTek 9940A drives with 9940B drives
- UDO1 drives with UDO2 drives

There are exceptions to the rule against mixing generations of LTO Ultrium drives and media. The Tivoli Storage Manager server does support mixtures of the following types:

- LTO Ultrium Generation 1 (LTO1) and LTO Ultrium Generation 2 (LTO2)
- LTO Ultrium Generation 2 (LTO2) with LTO Ultrium Generation 3 (LTO3)
- LTO Ultrium Generation 3 (LTO3) with LTO Ultrium Generation 4 (LTO4)
- LTO Ultrium Generation 4 (LTO4) with LTO Ultrium Generation 5 (LTO5)

The server supports these mixtures because the different drives can read and write to the different media. If you plan to upgrade all drives to Generation 2 (or Generation 3, Generation 4, or Generation 5), first delete all existing Ultrium drive definitions and the paths associated with them. Then you can define the new Generation 2 (or Generation 3, Generation 4, or Generation 5) drives and paths.

Note:

1. LTO Ultrium Generation 3 drives can only read Generation 1 media. If you are mixing Ultrium Generation 1 with Ultrium Generation 3 drives and media in a single library, you must mark the Generation 1 media as read-only, and all Generation 1 scratch volumes must be checked out.

2. LTO Ultrium Generation 4 drives can only read Generation 2 media. If you are mixing Ultrium Generation 2 with Ultrium Generation 4 drives and media in a single library, you must mark the Generation 2 media as read-only, and all Generation 2 scratch volumes must be checked out.
3. LTO Ultrium Generation 5 drives can only read Generation 3 media. If you are mixing Ultrium Generation 3 with Ultrium Generation 5 drives and media in a single library, you must mark the Generation 3 media as read-only, and all Generation 3 scratch volumes must be checked out.

To learn more about additional considerations when mixing LTO Ultrium generations, see “Defining LTO device classes” on page 211.

When using Tivoli Storage Manager you cannot mix 3592 generation 1, generation 2, and generation 3 drives. Use one of three special configurations. For details, see “Defining 3592 device classes” on page 204.

If you plan to encrypt volumes in a library, do not mix media generations in the library.

Mixed media and storage pools

You cannot mix media formats in a storage pool. Each unique media format must be mapped to a separate storage pool through its own device class.

This includes LTO1, LTO2, LTO3, and LTO4 formats. Multiple storage pools and their device classes of different types can point to the same library which can support them as explained in “Different media generations in a library” on page 58.

You can migrate to a new generation of a media type within the same storage pool by following these steps:

1. ALL older drives are replaced with the newer generation drives within the library (they cannot be mixed).
2. The existing volumes with the older formats are marked R/O if the new drive cannot append those tapes in the old format. If the new drive can write to the existing media in their old format, this is not necessary, but Step 1 is still required. If it is necessary to keep both LTO1 and LTO2 drives within the same library, separate storage pools for each must be used.

Removable media mounts and dismounts

When data is to be stored in or retrieved from a storage pool, the server selects the storage-pool volume and determines the name of the library that contains the drives to be used for the operation. When it has finished accessing the volume and the mount retention period has elapsed, the server dismounts the volume.

When data is to be stored in or retrieved from a storage pool, the server does the following:

1. The server selects a volume from the storage pool. The selection is based on the type of operation:

Retrieval

The name of the volume that contains the data to be retrieved is stored in the database.

Store If a defined volume in the storage pool can be used, the server selects that volume.

- If no defined volumes in the storage pool can be used, and if the storage pool allows it, the server selects a scratch volume.
2. The server checks the device class associated with the storage pool to determine the name of the library that contains the drives to be used for the operation.
 - The server searches the library for an available drive or until all drives have been checked. A drive status can be:
 - Offline.
 - Busy and not available for the mount.
 - In an error state and not available for the mount.
 - Online and available for the mount.
 3. The server mounts the volume:
 - For a manual library, the server displays a mount message for a private or a scratch volume to be mounted in the selected drive.
 - For an automated library, the server directs the library to move the volume from a storage slot into the selected drive. No manual intervention is required.

If a scratch mount is requested, the server checks the library's volume inventory for a scratch volume. If one is found, its status is changed to private, it is mounted in the drive, and it is automatically defined as part of the original storage pool. However, if the library's volume inventory does not contain any scratch volumes, the mount request fails.
 4. The server dismounts the volume when it has finished accessing the volume and the mount retention period has elapsed.
 - For a manual library, the server ejects the volume from the drive so that an operator can place it in its storage location.
 - For an automated library, the server directs the library to move the volume from the drive back to its original storage slot in the library.

How Tivoli Storage Manager uses and reuses removable media

Using Tivoli Storage Manager, you can control how removable media are used and reused. After Tivoli Storage Manager selects an available medium, that medium is used and eventually reclaimed according to its associated policy.

Tivoli Storage Manager manages the data on the media, but you manage the media itself, or you can use a removable media manager. Regardless of the method used, managing media involves creating a policy to expire data after a certain period of time or under certain conditions, move valid data onto new media, and reuse the empty media.

In addition to information about storage pool volumes, the volume history contains information about tapes used for database backups and exports (for disaster recovery purposes). The process for reusing these tapes is slightly different from the process for reusing tapes containing client data backups.

Figure 9 on page 61 shows a typical life cycle for removable media. The numbers (such as 1) refer to numbers in the figure.

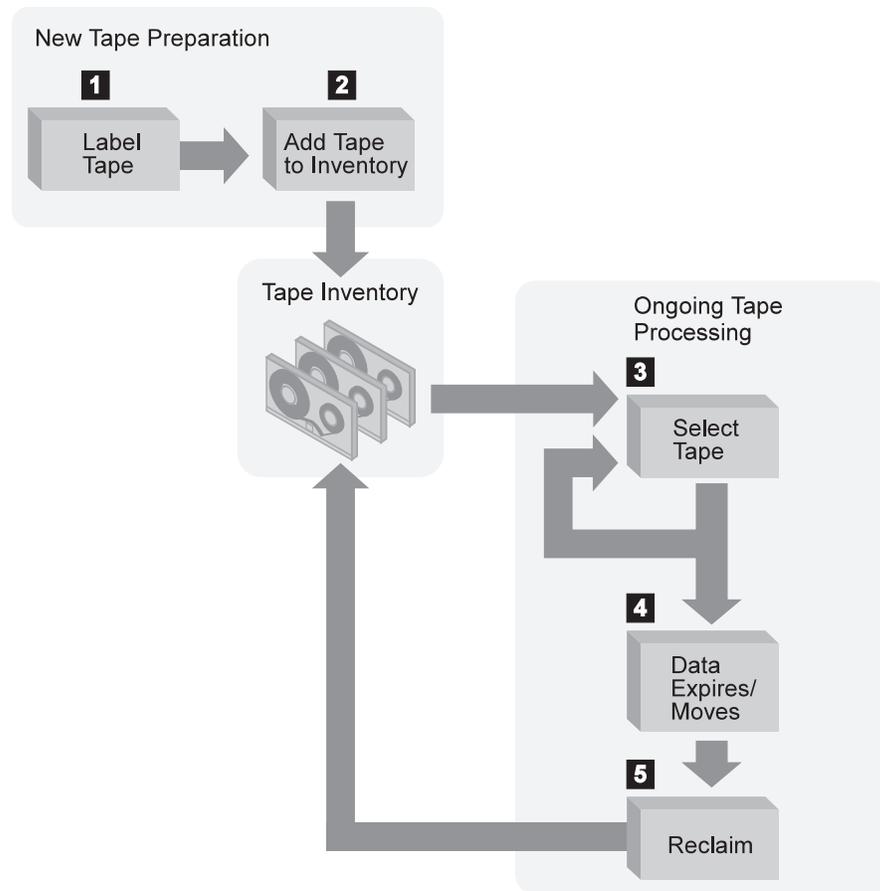


Figure 9. Simplified view of the life cycle of a tape

1. You label 1 and check in 2 the media. Checking media into a manual library simply means storing them (for example, on shelves). Checking media into an automated library involves adding them to the library volume inventory. See
 - “Labeling removable media volumes” on page 134
2. If you plan to define volumes to a storage pool associated with a device, you should check in the volume with its status specified as private. Use of scratch volumes is more convenient in most cases.
3. A client sends data to the server for backup, archive, or space management. The server stores the client data on the volume. Which volume the server selects 3 depends on:
 - The policy domain to which the client is assigned.
 - The management class for the data (either the default management class for the policy set, or the class specified by the client in the client's include/exclude list or file).
 - The storage pool specified as the destination in either the management class (for space-managed data) or copy group (for backup or archive data). The storage pool is associated with a device class, which determines which device and which type of media is used.
 - Whether the maximum number of scratch volumes that a server can request from the storage pool has been reached when the scratch volumes are selected.

- Whether collocation is enabled for that storage pool. When collocation is enabled, the server attempts to place data for different client nodes, groups of client nodes, or client file spaces on separate volumes. For more information, see “Keeping client files together using collocation” on page 310.

Figure 10 shows more detail about the policies and storage pool specifications which govern the volume selection described in step 3.

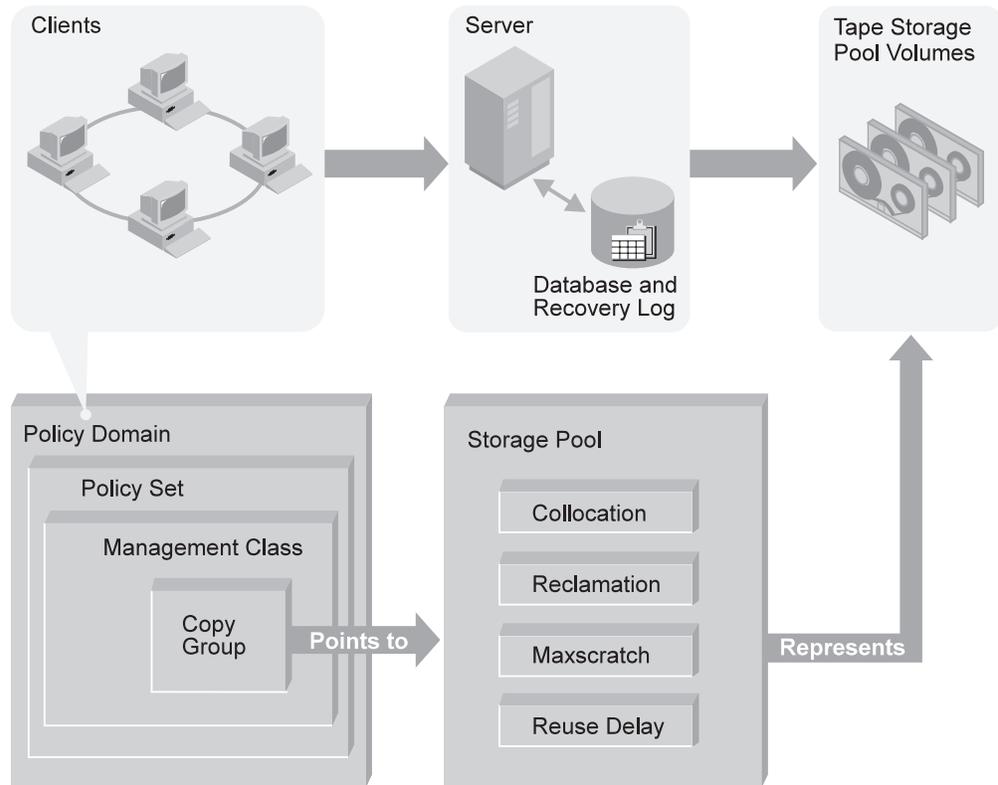


Figure 10. How Tivoli Storage Manager affects media use

4. The data on a volume changes over time as a result of:
 - Expiration of files 4 (affected by management class and copy group attributes, and the frequency of expiration processing). See “Basic policy planning” on page 436.
 - Movement and deletion of file spaces by an administrator.
 - Automatic reclamation of media 5

The amount of data on the volume and the reclamation threshold set for the storage pool affects when the volume is reclaimed. When the volume is reclaimed, any valid, unexpired data is moved to other volumes or possibly to another storage pool (for storage pools with single-drive libraries).

- Collocation, by which Tivoli Storage Manager attempts to keep data belonging to a single client node, group of client nodes, or client file space on a minimal number of removable media in a storage pool.

If the volume becomes empty because all valid data either expires or is moved to another volume, the volume is available for reuse (unless a time delay has been specified for the storage pool). The empty volume becomes a scratch volume if it was initially a scratch volume. The volume starts again at step 3 on page 61.

5. You determine when the media has reached its end of life.

For volumes that you defined (private volumes), check the statistics on the volumes by querying the database. The statistics include the number of write passes on a volume (compare with the number of write passes recommended by the manufacturer) and the number of errors on the volume.

You must move any valid data off a volume that has reached end of life. Then, if the volume is in an automated library, check out the volume from the library. If the volume is not a scratch volume, delete the volume from the database.

Required definitions for storage devices

Before the Tivoli Storage Manager server can use a device, the device must be configured to the operating system as well as to the server.

The Device Configuration Wizard, available in the Administration Center, automatically detects storage devices attached to the Tivoli Storage Manager server. You can use this wizard to select the devices you want to use with Tivoli Storage Manager, and to configure device sharing if required.

Table 8 summarizes the definitions that are required for different device types.

Table 8. Required definitions for storage devices

Device	Device Types	Required Definitions			
		Library	Drive	Path	Device Class
Magnetic disk	DISK	—	—	—	Yes ^{See note}
	FILE ^{See note}	—	—	—	Yes
Tape	3590 3592 4MM 8MM DLT LTO NAS QIC VOLSAFE ECARTRIDGE ^{See note}	Yes	Yes	Yes	Yes
Optical	OPTICAL WORM WORM12 ^{See note} WORM14 ^{See note}	Yes	Yes	Yes	Yes
Removable media (file system)	REMOVABLEFILE	Yes	Yes	Yes	Yes
Virtual volumes	SERVER	—	—	—	Yes

Notes:

- The DISK device class exists at installation and cannot be changed.
- FILE libraries, drives and paths are required for sharing with storage agents.
- The ECARTRIDGE device type is for StorageTek's cartridge tape drives such as
 - SD-3, 9480, 9890, and 9940 drives

Example: Mapping devices to device classes

You have internal disk drives, an automated tape library with 8 mm drives, and a manual DLT tape drive. You create a device class for each type of storage.

To map storage devices to device classes, use the information shown in Table 9.

Table 9. Mapping storage devices to device classes

Device Class	Description
DISK	Storage volumes that reside on the internal disk drive Tivoli Storage Manager provides one DISK device class that is already defined. You do not need and cannot define another device class for disk storage.
8MM_CLASS	Storage volumes that are 8 mm tapes, used with the drives in the automated library
DLT_CLASS	Storage volumes that are DLT tapes, used on the DLT drive

You must define any device classes that you need for your removable media devices such as tape drives. See Chapter 9, “Defining device classes,” on page 199 for information on defining device classes to support your physical storage environment.

Example: Mapping storage pools to device classes and devices

After you categorize your storage devices, you can identify availability, space, and performance requirements for client data that is stored in server storage. These requirements help you determine where to store data for different groups of clients and different types of data. You can then create storage pools that are storage destinations for backed-up, archived, or space-managed files to match requirements.

For example, you determine that users in the business department have three requirements:

- Immediate access to certain backed-up files, such as accounts receivable and payroll accounts.
These files should be stored on disk. However, you need to ensure that data is moved from the disk to prevent it from becoming full. You can set up a storage hierarchy so that files can migrate automatically from disk to the automated tape library.
- Periodic access to some archived files, such as monthly sales and inventory reports.
These files can be stored on 8-mm tapes, using the automated library.
- Occasional access to backed-up or archived files that are rarely modified, such as yearly revenue reports.
These files can be stored using the DLT drive.

To match user requirements to storage devices, you define storage pools, device classes, and, for device types that require them, libraries and drives. For example, to set up the storage hierarchy so that data migrates from the BACKUPPOOL to 8 mm tapes, you specify BACKTAPE1 as the next storage pool for BACKUPPOOL. See Table 10 on page 65.

Table 10. Mapping storage pools to device classes, libraries, and drives

Storage Pool	Device Class	Library (Hardware)	Drives	Volume Type	Storage Destination
BACKUPPOOL	DISK	—	—	Storage volumes on the internal disk drive	For a backup copy group for files requiring immediate access
BACKTAPE1	8MM_CLASS	AUTO_8MM (Exabyte EXB-210)	DRIVE01, DRIVE02	8-mm tapes	For overflow from the BACKUPPOOL and for archived data that is periodically accessed
BACKTAPE2	DLT_CLASS	MANUAL_LIB (Manually mounted)	DRIVE03	DLT tapes	For backup copy groups for files that are occasionally accessed

Note: Tivoli Storage Manager has the following default disk storage pools:

- BACKUPPOOL
- ARCHIVEPOOL
- SPACEMGPOOL

For more information, see

“Configuring random access volumes on disk devices” on page 75

Planning for server storage

To determine the device classes and storage pools that you need for your server storage, you must evaluate the devices in your storage environment.

1. Determine which drives and libraries are supported by the server. For more information on device support, see “Tivoli Storage Manager storage devices” on page 42.
2. Determine which storage devices may be selected for use by the server. For example, determine how many tape drives you have that you will allow the server to use. For more information about selecting a device configuration, see “Device configurations” on page 52

The servers can share devices in libraries that are attached through a SAN. If the devices are not on a SAN, the server expects to have exclusive use of the drives defined to it. If another application (including another Tivoli Storage Manager server) tries to use a drive while the server to which the drive is defined is running, some server functions may fail. For more information about specific drives and libraries, see http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

3. Determine the device driver that supports the devices. For more information on device driver support, see:
 - “Device driver selection” on page 80
4. Determine how to attach the devices to the server. . For more information about attaching devices, see:
 - “Attaching an automated library device” on page 79
5. Determine whether to back up client data directly to tape or to a storage hierarchy.
6. Determine which client data is backed up to which device, if you have multiple device types.

7. Determine the device type and device class for each of the available devices. Group together similar devices and identify their device classes. For example, create separate categories for 4 mm and 8 mm devices.

Tip: For sequential access devices, you can categorize the type of removable media based on their capacity. For example, standard length cartridge tapes and longer length cartridge tapes require different device classes.

8. Determine how the mounting of volumes is accomplished for the devices:
 - Devices that require operators to load volumes must be part of a defined MANUAL library.
 - Devices that are automatically loaded must be part of a defined SCSI or 349X. Each automated library device is a separate library.
 - Devices that are controlled by Sun StorageTek Automated Cartridge System Library Software (ACSL) must be part of a defined ACSL library.
 - Devices that are managed by an external media management system must be part of a defined EXTERNAL library.
9. If you are considering storing data for one Tivoli Storage Manager server using the storage of another Tivoli Storage Manager server, consider network bandwidth and network traffic. If your network resources constrain your environment, you may have problems using the SERVER device type efficiently.

Also consider the storage resources available on the target server. Ensure that the target server has enough storage space and drives to handle the load from the source server.
10. Determine the storage pools to set up, based on the devices you have and on user requirements. Gather users' requirements for data availability. Determine which data needs quick access and which does not.
11. Be prepared to label removable media. You may want to create a new labeling convention for media so that you can distinguish them from media used for other purposes.

Server options that affect storage operations

Tivoli Storage Manager provides a number of options that you can specify in the server options file (`dsmserv.opt`) to configure certain server storage operations.

Table 11 provides brief descriptions of these options. See the *Administrator's Reference* for details.

Table 11. Server storage options

Option	Description
3494SHARED	Enables sharing of an IBM TotalStorage 3494 Tape Library between a Tivoli Storage Manager server and server applications other than a Tivoli Storage Manager server. This configuration is not recommended, because this configuration can cause drive contention.
ACSACCESSID	Specifies the ID for the Automatic Cartridge System (ACS) access control.
ACSLOCKDRIVE	Allows the drives within ACSL libraries to be locked.
ACSQUICKINIT	Allows a quick or full initialization of the ACSL library.
ACSTIMEOUTX	Specifies the multiple for the built-in timeout value for ACSL API.

Table 11. Server storage options (continued)

Option	Description
ASSISTVCRRECOVERY	Specifies whether the server assists an IBM 3590 drive in recovering from a lost or corrupted Vital Cartridge Records (VCR) condition.
DRIVEACQUIRERETRY	Specifies how many times the server retries the acquisition of a drive in a library when there are no drives available after acquiring a mount point.
NOPREEMPT	Specifies whether the server allows certain operations to preempt other operations for access to volumes and devices. See "Preemption of client or server operations" on page 566 for details.
RESOURCETIMEOUT	Specifies how long the server waits for a resource before canceling the pending acquisition of a resource. Note: For proper management of shared library resources, consider setting the RESOURCETIMEOUT option at the same time limit for all servers in a shared configuration. In the case of error recovery, Tivoli Storage Manager always defers to the longest time limit.
SEARCHMPQUEUE	Specifies the order in which the server satisfies requests in the mount queue.

Chapter 4. Magnetic disk devices

Using magnetic disk devices, Tivoli Storage Manager can store essential data for the server and client environments.

Tivoli Storage Manager stores data on magnetic disks in random access volumes, as data is normally stored on disk, and in files on the disk that are treated as sequential access volumes.

Magnetic disk devices allow you to:

- Store the database and the recovery log.
- Store client data that has been backed up, archived, or migrated from client nodes. The client data is stored in storage pools. Procedures for configuring disk storage of client data are described in this chapter.
- Store backups of the database and export and import data.

See the following sections:

Tasks:
“Configuring random access volumes on disk devices” on page 75
“Configuring FILE sequential volumes on disk devices” on page 76
“Varying disk volumes online or offline” on page 77
“Cache copies for files stored on disk” on page 77
“Freeing space on disk” on page 78
“Scratch FILE volumes” on page 78
“Volume history file and volume reuse” on page 78

Note: Some of the tasks described in this chapter require an understanding of storage objects. For an introduction to these storage objects, see “Tivoli Storage Manager storage objects” on page 42.

Requirements for disk subsystems

Tivoli Storage Manager requires certain behaviors of disk storage subsystems for the database, the active and archive logs, and storage pool volumes of the DISK device class and of FILE device types.

I/O operation results must be reported synchronously and accurately. For the database and the active and archive logs, unreported or asynchronously reported write errors that result in data not being permanently committed to the storage subsystem can cause failures that range from internal processing errors to the inability to restart the server. Depending upon the error, the result could be the loss of some or all stored data.

Data in Tivoli Storage Manager storage pools, database volumes, and log volumes are interdependent. Tivoli Storage Manager requires that the data written to these entities can be retrieved exactly as it was written. Also data in these entities must be consistent with one another. There cannot be timing windows in which data being retrieved varies depending on the way that an I/O subsystem manages

writes. Generally, this means that replicated Tivoli Storage Manager environments must use features such as maintenance of write-order between the source and replication targets. It also requires that the database, log, and disk storage pool volumes be part of a consistency group in which any I/O to the members of the target consistency group are written in the same order as the source and maintain the same volatility characteristics. Requirements for I/O to disk storage subsystems at the remote site must also be met.

For the database, the active and archive logs, and DISK device class storage pool volumes, write operations must be nonvolatile. Data must be permanently committed to the storage known to Tivoli Storage Manager. Tivoli Storage Manager has many of the attributes of a database system, and data relationships that are maintained require that data written as a group be permanently resident as a group or not resident as a group. Intermediate states produce data integrity issues. Data must be permanently resident following each operating-system write API invocation.

For FILE device type storage pool volumes, data must be permanently resident following an operating system flush API invocation. This API is used at key processing points in the Tivoli Storage Manager application. The API is used when data is to be permanently committed to storage and synchronized with database and log records that have already been permanently committed to disk storage.

For subsystems that use caches of various types, the data must be permanently committed by the write APIs (for the database, the active and archive logs, and DISK device class storage pool volumes) and by the flush API (for FILE device class storage pool volumes). Tivoli Storage Manager uses write-through flags internally when using storage for the database, the active and archive logs, and DISK device class storage pool volumes. If nonvolatile cache is used to safeguard I/O writes to a device, if the nonvolatile cache is battery protected, and if the power is not restored before the battery is exhausted, data for the I/O operation can be lost. This would be the same as having uncommitted storage resulting in data integrity issues.

To write properly to the Tivoli Storage Manager database, to active and archive logs, and to DISK device class storage pool volumes, the operating system API write invocation must synchronously and accurately report the operation results. Similarly, the operating system API flush invocation for FILE device type storage pool volumes must also synchronously and accurately report the operation results. A successful result from the API for either write or flush must guarantee that the data is permanently committed to the storage subsystem.

These requirements extend to replicated environments such that the remote site must maintain consistency with the source site in terms of the order of writes; I/O must be committed to storage at the remote site in the same order that it was written at the source site. The ordering applies to the set of files that Tivoli Storage Manager is writing, whether the files belong to the database, recovery log, or storage pool volumes. Tivoli Storage Manager can recover from incomplete I/O scenarios as long as the ordering of writes is consistent between the source and target site.

To avoid having the Tivoli Storage Manager server at the local and remote site losing synchronization, the server at the remote site should not be started except in a fail-over situation. If there is a possibility that data at the source and target locations can lose synchronization, there must be a mechanism to recognize this situation. If synchronization has been lost, the Tivoli Storage Manager server at the

remote location must be restored by conventional means using Tivoli Storage Manager database and storage pool restores.

Contact the vendor for the disk subsystem if you have questions or concerns about whether the stated requirements for Tivoli Storage Manager are supported. The vendor should be able to provide the configuration settings to meet these requirements.

Tivoli Storage Manager supports the use of remote file systems or drives for reading and writing storage pool data, database backups, and other data operations. Remote file systems in particular might report successful writes, even after being configured for synchronous operations. This mode of operation causes data integrity issues if the file system can fail after reporting a successful write. Check with the vendor of your file system to ensure that flushes are performed to nonvolatile storage in a synchronous manner.

Random access and sequential access disk devices

Before configuring your disk device, you should consider the differences between the two methods of storing data on disks and the advantages and disadvantages of each. The particular advantages provided by either device type will depend on the operating system on which your Tivoli Storage Manager server is running.

Table 12 provides some general information about the characteristics of DISK devices (random access) and FILE devices (sequential access) and the benefits of each.

Table 12. Comparing random access and sequential access disk devices

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
Storage space allocation and tracking	Disk blocks.	Volumes.	Space allocation and tracking by blocks incurs higher overhead (more database storage space, and more processing power) than space allocation and tracking by volume.
Concurrent volume access	A volume can be accessed concurrently by different operations.	A volume can be accessed concurrently by different operations.	Concurrent volume access means that two or more different operations can access the same volume at the same time.

Table 12. Comparing random access and sequential access disk devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
Client restore operations	One session per restore.	Multiple concurrent sessions accessing different volumes simultaneously on both the server and the storage agent. Active versions of client backup data collocated in active-data pools.	Multi-session restore enables backup-archive clients to perform multiple restore sessions for no-query restore operations, increasing the speed of restores. Active-data pools defined using sequential-access disk (FILE) enable fast client restore because the server does not have to physically mount tapes and does not have to position past inactive files. For more information, see "Concepts for client restore operations" on page 517 and "Backing up storage pools" on page 760.
Available for use in LAN-free backup	Not available.	Available for LAN-free backup using Tivoli SANergy®, a separate product, licensed to users through the Tivoli Storage Manager product. Tivoli SANergy is included with some versions of Tivoli Storage Manager.	Using LAN-free backup, data moves over a dedicated storage area network (SAN) to the sequential-access storage device, freeing up bandwidth on the LAN. For more information, see "LAN-free data movement" on page 54.
Volume configuration	Operators need to define volumes and specify their sizes, or define space triggers to automatically allocate space when a threshold is reached.	The Tivoli Storage Manager server acquires and defines scratch volumes as needed if storage administrators set the MAXSCRATCH parameter to a value greater than zero. Operators can also define space triggers to automatically allocate space when a threshold is reached.	For more information about volumes on random-access media, see "Configuring random access volumes on disk devices" on page 75. For more information about volumes on FILE devices, see "Configuring FILE sequential volumes on disk devices" on page 76.
Tivoli Storage Manager server caching (after files have been migrated to the next storage pool in the storage pool hierarchy)	Server caching is available, but overhead is incurred in freeing the cached space. For example, as part of a backup operation, the server must erase cached files to make room for storing new files.	Server caching is not necessary because access times are comparable to random access (DISK) access times.	Caching can improve how quickly the Tivoli Storage Manager server retrieves files during client restore or retrieve operations. For more information, see "Caching in disk storage pools" on page 263.

Table 12. Comparing random access and sequential access disk devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
Recovery of disk space	<p>When caching is enabled, the space occupied by cached files is reclaimed on demand by the server.</p> <p>When caching is disabled, the server recovers disk space immediately after all physical files are migrated or deleted from within an aggregate.</p>	<p>The server recovers disk space in a process called <i>reclamation</i>, which involves copying physical files to another volume, making the reclaimed volume available for reuse. This minimizes the amount of overhead because there is no mount time required.</p>	<p>For more information about reclamation, see “Reclaiming space in sequential-access storage pools” on page 319.</p>
Aggregate reconstruction	<p>Not available; the result is wasted space.</p>	<p>Aggregate reconstruction occurs as part of the reclamation process. It is also available using the RECONSTRUCT parameter on the MOVE DATA and MOVE NODEDATA commands.</p>	<p>An <i>aggregate</i> is two or more files grouped together for storage purposes. Most data from backup-archive clients is stored in aggregates. Aggregates accumulate empty space as files are deleted, expire, or as they are deactivated in active-data pools. For more information, see “How Tivoli Storage Manager reclamation works” on page 319.</p>
Available for use as copy storage pools or active-data pools	<p>Not available.</p>	<p>Available.</p>	<p>Copy storage pools and active-data pools provide additional levels of protection for client data. For more information, see “Backing up storage pools” on page 760.</p>
File location	<p>Volume location is limited by the trigger prefix or by manual specification.</p>	<p>FILE volumes use directories. A list of directories may be specified. If directories correspond with file systems, performance is optimized.</p>	

Table 12. Comparing random access and sequential access disk devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
Restoring the database to an earlier level	See comments.	Use the REUSEDELAY parameter to retain volumes in a pending state; volumes are not rewritten until the specified number of days have elapsed. During database restoration, if the data is physically present, it can be accessed after DSMSEV RESTORE DB.	Use the AUDIT VOLUME command to identify inconsistencies between information about a volume in the database and the actual content of the volume. You can specify whether the Tivoli Storage Manager server resolves the database inconsistencies it finds. For more information about auditing volumes, see "Auditing storage pool volumes" on page 782. For more information about reuse delay, see "Delaying reuse of volumes for recovery purposes" on page 766. For command syntax, refer to the <i>Administrator's Reference</i> .
Migration	Performed by node. Migration from random-access pools can use multiple processes.	Performed by volume. Files are not migrated from a volume until all files on the volume have met the threshold for migration delay as specified for the storage pool. Migration from sequential-access pools can use multiple processes.	For more information, see "Migrating disk storage pools" on page 253.
Storage pool backup	Performed by node and filespace. Every storage pool backup operation must check every file in the primary pool to determine whether the file must be backed up.	Performed by volume. For a primary pool, there is no need to scan every object in the primary pool every time the pool is backed up to a copy storage pool.	For more information, see "Storage pools" on page 222.
Copying active data	Performed by node and filespace. Every storage pool copy operation must check every file in the primary pool to determine whether the file must be copied.	Performed by volume. For a primary pool, there is no need to scan every object in the primary pool every time the active data in the pool is copied to an active-data pool.	For more information, see "Storage pools" on page 222.
Transferring data from non-collocated to collocated storage	Major benefits by moving data from non-collocated storage to DISK storage, and then allowing data to migrate to collocated storage. See "Restoring files to a storage pool with collocation enabled" on page 779 for more information.	Some benefit by moving data from non-collocated storage to FILE storage, and then moving data to collocated storage.	For more information, see "Keeping client files together using collocation" on page 310.

Table 12. Comparing random access and sequential access disk devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
Shredding data	If shredding is enabled, sensitive data is shredded (destroyed) after it is deleted from a storage pool. Write caching on a random access device should be disabled if shredding is enforced.	Shredding is not supported on sequential access disk devices.	For more information, see “Securing sensitive client data” on page 499.
Data deduplication	Not available	Duplicate data in primary, copy, and active-data pools can be identified and removed, reducing the overall amount of time that is required to retrieve data from disk.	For more information, see “Deduplicating data” on page 264.

Configuring random access volumes on disk devices

Tivoli Storage Manager provides a predefined DISK device class that is used with all disk devices.

Define storage pool volumes on disk drives that reside on the server machine, not on remotely mounted file systems. Network attached drives can compromise the integrity of the data that you are writing. See “Disk devices” on page 45 for more information on requirements for local and remote file systems.

Complete the following steps to use random access volumes on a disk device:

1. Define a storage pool that is associated with the DISK device class, or use one of the default storage pools that Tivoli Storage Manager provides (ARCHIVEPOOL, BACKUPPOOL, and SPACEMGPOOL).

For example, enter the following command on the command line of an administrative client:

```
define stgpool engback1 disk maxsize=5G highmig=85 lowmig=40
```

This command defines storage pool ENGBACK1.

See “Example: Defining storage pools” on page 231 for details.

2. Prepare a volume for use in a random access storage pool by defining the volume. If you do not specify a full path name, the command uses the current path. See “Defining storage pool volumes” on page 237 for details. See the following example:

You want to define a 21 MB volume for the ENGBACK1 storage pool. You want the volume to be located in the path `/opt/tivoli/tsm/server/bin` and named `stgvol.002`. Enter the following command:

```
define volume engback1 /opt/tivoli/tsm/server/bin/stgvol.002 formatsize=21
```

Tivoli Storage Manager does not support raw logical volumes on Linux.

3. Do one of the following:
 - Specify the new storage pool as the destination for client files that are backed up, archived, or migrated, by modifying existing policy or creating new policy. See Chapter 14, “Implementing policies for client data,” on page 435 for details.

- Place the new storage pool in the storage pool migration hierarchy by updating an already defined storage pool. See “Example: Updating storage pools” on page 232.

Configuring FILE sequential volumes on disk devices

Magnetic disk storage uses files as volumes that store data sequentially (as on tape volumes). The space for FILE volumes is managed by the operating system rather than by Tivoli Storage Manager.

To use files as volumes that store data sequentially, do the following:

1. Define a device class with device type FILE.

For example, enter the following command on the command line of an administrative client:

```
define devclass fileclass devtype=file mountlimit=2 maxcapacity=2G
```

This command defines device class FILECLASS with a device type of FILE.

See:

“Defining sequential-access disk (FILE) device classes” on page 207

To store database backups or exports on FILE volumes, this step is all you need to do to prepare the volumes. You can use FILE sequential volumes to transfer data for purposes such as electronic vaulting. For example, you can send the results of an export operation or a database backup operation to another location. At the receiving site, the files can be placed on tape or disk. You can define a device class with a device type of FILE. For more information, see “Defining device classes for backups” on page 767 and “Planning for sequential media used to export data” on page 733.

2. Define a storage pool that is associated with the new FILE device class.

For example, enter the following command on the command line of an administrative client:

```
define stgpool engback2 fileclass maxscratch=100 mountlimit=2
```

This command defines storage pool ENGBACK2 with device class FILECLASS.

See “Defining storage pools” on page 227 for details.

To allow Tivoli Storage Manager to use scratch volumes for this device class, specify a value greater than zero for the number of maximum scratch volumes when you define the device class. If you do set MAXSCRATCH=0 to not allow scratch volumes, you must define each volume to be used in this device class. See “Preparing volumes for sequential-access storage pools” on page 236 for details.

3. Do one of the following:

- Specify the new storage pool as the destination for client files that are backed up, archived, or migrated, by modifying existing policy or creating new policy. See Chapter 14, “Implementing policies for client data,” on page 435 for details.
- Place the new storage pool in the storage pool migration hierarchy by updating an already defined storage pool. See “Example: Updating storage pools” on page 232.

You can also set up predefined sequential volumes with the DEFINE VOLUME command using:

```
define volume poolname prefix numberofvolumes=x
```

where x specifies the number of volumes that can be created at once with a size taken from the device class' maximum capacity. The advantage to this method is that a space is pre-allocated and not subject to additional fragmentation in the file system as scratch volumes are.

For storage pools associated with the FILE device class, you can also use the DEFINE SPACETRIGGER and UPDATE SPACETRIGGER commands to create volumes and assign them to a specified storage pool when predetermined space-utilization thresholds have been exceeded. For more information, see the *Administrator's Reference*.

1. From the **Tivoli Storage Manager Console**, expand the tree for the server instance you are configuring.
2. Click **Wizards**, then double-click **Device Configuration** in the right pane.
3. Navigate to the **Tivoli Storage Manager Device Selection** page and click **New**. The **Properties** dialog appears.
4. Select **File Device** from the drop down list.
5. Enter or browse for the directory you want to allocate as a FILE volume.
6. Click **OK**. Tivoli Storage Manager configures the FILE volume.
7. Click **Next** to complete the wizard.

Varying disk volumes online or offline

To perform maintenance on a disk volume or to upgrade disk hardware, you can vary a disk volume offline. If Tivoli Storage Manager encounters a problem with a disk volume, the server automatically varies the volume offline.

Task	Required Privilege Class
Vary a disk volume online or offline	System or operator

For example, to vary the disk volume named `/storage/pool001` offline, enter:

```
vary offline /storage/pool001
```

You can make the disk volume available to the server again by varying the volume online. For example, to make the disk volume named `/storage/pool001` available to the server, enter:

```
vary online /storage/pool001
```

Cache copies for files stored on disk

When you define a storage pool that uses disk random access volumes, you can choose to enable or disable cache. When you use cache, a copy of the file remains on disk storage even after the file has been migrated to the next pool in the storage hierarchy (for example, to tape). The file remains in cache until the space it occupies is needed to store new files.

Using cache can improve how fast a frequently accessed file is retrieved. Faster retrieval can be important for clients storing space-managed files. If the file needs to be accessed, the copy in cache can be used rather than the copy on tape. However, using cache can degrade the performance of client backup operations and increase the space needed for the database. For more information, see "Caching in disk storage pools" on page 263.

Freeing space on disk

As client files expire, the space they occupy is not freed for other uses until you run expiration processing on the server.

Expiration processing deletes from the database information about any client files that are no longer valid according to the policies you have set. For example, suppose four backup versions of a file exist in server storage, and only three versions are allowed in the backup policy (the management class) for the file. Expiration processing deletes information about the oldest of the four versions of the file. The space that the file occupied in the storage pool becomes available for reuse.

You can run expiration processing by using one or both of the following methods:

- Use the EXPIRE INVENTORY command. See “Running expiration processing to delete expired files” on page 472.
- Set the server option for the expiration interval, so that expiration processing runs periodically. See the *Administrator's Reference* for information on how to set the options.

Shredding occurs only after a data deletion commits, but it is not necessarily completed immediately after the deletion. The space occupied by the data to be shredded remains occupied while the shredding takes place, and is not available as free space for new data until the shredding is complete. When sensitive data is written to server storage and the write operation fails, the data that was already written is shredded. For more information, see “Securing sensitive client data” on page 499.

Scratch FILE volumes

When the server needs a new volume, the server automatically creates a file that is a scratch volume, up to the number you specify.

You can specify a maximum number of scratch volumes for a storage pool that has a FILE device type.

When scratch volumes used in storage pools become empty, the files are deleted. Scratch volumes can be located in multiple directories on multiple file systems.

Volume history file and volume reuse

When you back up the database or export server information, Tivoli Storage Manager records information about the volumes used for these operations in the *volume history*. Tivoli Storage Manager will not allow you to reuse these volumes until you delete the volume information from the volume history.

To reuse volumes that have previously been used for database backup or export, use the DELETE VOLHISTORY command. For information about the volume history and volume history files, see “Saving the volume history file” on page 768.

Note: If your server is licensed for the disaster recovery manager (DRM) function, the volume information is automatically deleted during MOVE DRMEDIA command processing. For additional information about DRM, see Chapter 26, “Using disaster recovery manager,” on page 801.

Chapter 5. Using devices with the server system

For Tivoli Storage Manager to use a device, you must attach the device to your server system and install the appropriate device driver.

Attached devices should be on their own Host Bus Adapter (HBA) and should not share with other devices types (disk, CDROM, and so on). IBM tape drives have some special requirements for HBAs and associated drivers.

Tasks:
"Attaching a manual drive"
"Attaching an automated library device"
"Device driver selection" on page 80

Attaching a manual drive

Attaching manual drives to your system allows you to utilize storage.

Perform the following steps to attach a manual drive:

1. Install the SCSI or FC adapter card in your system and associated drivers of the adapter card, if not already installed.
2. Determine the SCSI IDs available on the SCSI or FC adapter card to which you are attaching the device. Find one unused SCSI ID for each drive.
3. Follow the manufacturer's instructions to set the SCSI ID for the drive to the unused SCSI IDs that you found. You may have to set switches on the back of the device or set the IDs on the operator's panel.

Each device that is connected in a chain to a single SCSI bus through a fibre channel adapter card must be set to a unique SCSI ID. If devices are not set sequentially, and there is a gap in the sequence, the system will only see the first device.

4. Power off your system before attaching a device to prevent damage to the hardware.
5. Attach a terminator to the last device in the chain of devices connected on one SCSI adapter card.
6. Follow the manufacturer's instructions to attach the device to your server system hardware.
7. Install the appropriate device drivers for attached tape devices.

See "Device driver selection" on page 80.

Attaching an automated library device

Perform the following steps to attach an automated library device.

1. Install the SCSI or FC adapter card in your system and associated drivers of the adapter card, if not already installed. The attached devices should be on their own Host Bus Adapter (HBA) and should not share which other devices types (disk, CDROM, etc). The IBM tape drives have some special requirements on HBA and associated drivers. Check the support Web site for details.

2. Determine the SCSI IDs available on the SCSI adapter card to which you are attaching the device. Find one unused SCSI ID for each drive, and one unused SCSI ID for the library or autochanger controller.

Note: In some automated libraries, the drives and the autochanger share a single SCSI ID, but have different LUNs. For these libraries, only a single SCSI ID is required. Check the documentation for your device.

3. Follow the manufacturer's instructions to set the SCSI ID for the drives to the unused SCSI IDs that you found. You may have to set switches on the back of the device or set the IDs on the operator's panel. Each device that is connected in a chain to a single SCSI bus must be set to a unique SCSI ID. If each device does not have a unique SCSI ID, serious system problems can arise.
4. Follow the manufacturer's instructions to attach the device to your server system hardware.

Attention:

- a. Power off your system before attaching a device to prevent damage to the hardware.
 - b. Attach a terminator to the last device in the chain of devices connected on one SCSI adapter card. Detailed instructions should be in the documentation that came with your hardware.
5. Install the appropriate device drivers for attached medium changer devices. See "Device driver selection."

Setting the library mode

For the Tivoli Storage Manager server to access a SCSI library, the device must be set for the appropriate mode.

The appropriate mode is usually called *random* mode; however, terminology may vary from one device to another. Refer to the documentation for your device to determine how to set it to the appropriate mode.

Note:

1. Some libraries have front panel menus and displays that can be used for explicit operator requests. However, if you set the device to respond to such requests, it typically will not respond to Tivoli Storage Manager requests.
2. Some libraries can be placed in *sequential* mode, in which volumes are automatically mounted in drives by using a sequential approach. This mode conflicts with how Tivoli Storage Manager accesses the device.
3. Some libraries can support both SCSI 2 and SCSI 3 modes. Normally, the SCSI 3 mode should be set. Check the device manual for more information.

Device driver selection

To use devices, you must install the appropriate device driver. IBM device drivers are available for most IBM labeled devices.

The Tivoli Storage Manager Passthru driver, which is provided with IBM Tivoli Storage Manager, is available for both non-IBM tape and automated library and optical devices.

Note: IBM device drivers only support some specific Linux kernel levels. For information about supported kernel levels, see <ftp://ftp.software.ibm.com/>

storage/devdrv/Linux/archive/ .

IBM device drivers

Tivoli Storage Manager supports IBM device drivers for some devices.

IBM device drivers support some specific Linux kernel levels. For details, check the support Web site at <ftp://ftp.software.ibm.com/storage/devdrv/Linux/archive/>. For the most up-to-date list of devices and operating-system levels supported by IBM device drivers, see the Tivoli Storage Manager Supported Devices Web site at http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_Linux.html. For installing devices on zSeries® systems, please see “Installing zSeries Linux fibre-channel adapter (zfcp) device drivers” on page 84.

For device driver installation information, see the *IBM Tape Device Drivers Installation and User's Guide*. You can download the guide from the Doc folder on <ftp://ftp.software.ibm.com/storage/devdrv/>. Tivoli Storage Manager supports all devices that are supported by IBM device drivers. However, Tivoli Storage Manager does not support all the operating-system levels that are supported by IBM device drivers.

Tivoli Storage Manager device drivers

Tivoli Storage Manager provides device drivers to work with a variety of devices.

- For optical and WORM devices, you must install the Tivoli Storage Manager device driver.
- All SCSI-attached libraries that contain optical and tape drives from the list above must use the Tivoli Storage Manager changer driver.

Tivoli Storage Manager Version 5.3 and later supports the Passthru driver, which replaces the Tivoli Storage Manager Linux SCSI device driver. The Tivoli Storage Manager Passthru drivers require the Linux SCSI generic (sg) driver along with the Linux operating system to install the kernels.

- For the following tape devices, install the Tivoli Storage Manager Passthru driver:
 - All DLT and SDLT (including IBM 7337)
 - 4MM
 - 8MM
 - ECART
 - Ultrium (Non IBM LTO)
 - QIC
 - StorageTek SD3
 - StorageTek 9490
 - StorageTek 9840
 - StorageTek 9940
- All SCSI-attached libraries that contain tape drives from the list above must use the Tivoli Storage Manager Passthru driver.

For more information, see the following topics:

- “Configuring Tivoli Storage Manager passthru drivers for tape and optical devices and libraries” on page 83

Drivers for optical devices

The Passthru driver supports optical devices that include both magneto-optic and WORM-optic disc devices.

Passthru drivers are available at <ftp://ftp.software.ibm.com/storage/tivoli-storage-management/maintenance/server>. See the *Installation Guide* for installation instructions and refer to “Configuring Tivoli Storage Manager passthru drivers for tape and optical devices and libraries” on page 83 for more information.

For the following optical devices, install the Tivoli Storage Manager Passthru driver:

- Optical Drives
- Optical Jukeboxes

All SCSI-attached libraries that contain optical drives or jukeboxes must also use the Tivoli Storage Manager Passthru driver.

Installing and configuring device drivers

In order to use devices with Tivoli Storage Manager, you must install the correct device driver.

For IBM device driver installation information, see the *IBM Tape Device Drivers Installation and User's Guide*. You can download the guide from the Doc folder on <ftp://ftp.software.ibm.com/storage/devdrv/>. Tivoli Storage Manager supports all devices that are supported by IBM device drivers. However, Tivoli Storage Manager does not support all the operating-system levels that are supported by IBM device drivers.

Installing device drivers for IBM SCSI tape devices

You must install the correct driver to use IBM SCSI tape devices.

For information on how to install device drivers for IBM SCSI devices, see the appropriate document:

- *IBM Tape Device Drivers Installation and User's Guide*
- *IBM Ultrium Device Drivers: Installation and User's Guide*

The guides can be downloaded from the FTP site at <ftp://ftp.software.ibm.com/storage/devdrv/>. They are located in the Doc folder.

Depending on the device driver you are installing, a message is issued after completing the procedure in the manual.

- If you are installing the device driver for an IBM 3480 or 3490 tape device, you receive:

IBMtapex Available

where IBMtapex is the logical filename for the tape device.

- If you are installing the device driver for an IBM 3581, 3583, 3584, or 3590 Model B11, you receive:

IBMtapex Available

or

IBMChangerx Available

Note the value of *x*, which is assigned automatically by the system. To determine the special file name of your device, use the */dev/IBM* prefix with the name provided by the system. For example:

- If the message is *IBMtape0 Available*, the special file name for the drive is */dev/IBMtape0*.
- If the message is *IBMChanger2 Available*, the special file name for the media changer device is */dev/IBMChanger2*.

The file name may have additional characters at the end to indicate different operating characteristics, but these are not needed by Tivoli Storage Manager. Use the base file name in the *Device=* parameter of the *DEFINE PATH* command to assign a device to a drive (*/dev/IBMtape*x**) or a library (*/dev/IBMChanger*x**).

Note: This applies to the IBM device driver only and the device type of this class must NOT be *GENERICTAPE*.

Installing device drivers for IBM 349x libraries

Complete the following procedure to install device drivers for IBM 349X libraries.

For an IBM 349X Tape Library Dataserver, refer to *IBM Tape Device Drivers Installation and User's Guide* to install the driver. The driver is the IBM Tape Library Driver, which consists of a daemon and other components.

To determine the name to use to define the path for the library, check what you entered in the */etc/ibmatl.conf* file for the library's symbolic name.

See “Installing device drivers for IBM SCSI tape devices” on page 82.

Configuring Tivoli Storage Manager passthru drivers for tape and optical devices and libraries

To use the Tivoli Storage Manager Linux Passthru driver, you must perform the following steps.

1. Verify that the device is connected to your system, and is powered on and active.
2. Verify that the device is correctly detected by your system by issuing this command:

```
cat /proc/scsi/scsi
```
3. Ensure that both the Tivoli Storage Manager device driver package (*tsmscsi*) and the storage server package are installed.
4. There are two driver configuration methods available in the Tivoli Storage Manager device driver package: *autoconf* and *tsmscsi*. Both of these methods perform the following tasks:
 - Load the Linux SCSI generic driver (*sg*) to the kernel.
 - Create necessary special files for the Passthru driver.
 - Create device information files for tape devices (*/dev/tsmscsi/mtinfo*), optical devices (*/dev/tsmscsi/opinfo*), and libraries (*/dev/tsmscsi/lbinfo*).
5. Run the configuration method you prefer (*autoconf* or *tsmscsi*) for the Tivoli Storage Manager Passthru driver.
 - To run the *autoconf* configuration method, issue the following command:

```
autoconf
```
 - To run the *tsmscsi* configuration method, perform the following steps:

- a. Copy the two sample configuration files located in the installation directory from *mt.conf.smp*, *op.conf.smp*, and *lb.conf.smp* to *mt.conf*, *op.conf*, and *lb.conf*, respectively.
- b. Edit the *mt.conf*, *op.conf*, and *lb.conf* files. Add one stanza (as shown in the example at the top of the file) for each SCSI target, ID, and LUN combination. Each combination of SCSI target, ID, and LUN entries correspond to a tape drive or library you want configured. Make sure the files meet these requirements:
 - Remove the example located at the top of the files.
 - There must be a new line between each stanza.
 - There must be one new line after the last stanza.
 - There must be no pound sign (#) in either file.
- c. Run the *tsmscsi* script from the device driver installation directory.

Note: For Tivoli Storage Manager Version 5.3 and later, you need to re-execute the *autoconf* or *tsmscsi* script to reconfigure Tivoli Storage Manager devices following a reboot. In general the Linux SCSI generic driver is pre-loaded to the kernel. To verify that the driver is in the kernel, you can issue the following command:

```
> lsmod | grep sg
```

If the driver is not in the kernel, you can issue the *modprobe sg* command to load the *sg* driver into the kernel.

6. Verify that the device is configured properly by viewing the text files for tape devices (*/dev/tsmscsi/mtinfo*), and optical devices (*/dev/tsmscsi/opinfo*), and libraries (*/dev/tsmscsi/lbinfo*).
7. Determine the special file names for the tape drives and libraries:
 - To determine the names for tape devices, issue the following command:


```
> ls /dev/tsmscsi/mt*
```
 - To determine the names for optical devices, issue the following command:


```
> ls /dev/tsmscsi/op*
```
 - To determine the names for libraries issue the following command:


```
> ls /dev/tsmscsi/lb*
```

This information helps you identify which of the */dev/tsmscsi/mtx* and */dev/tsmscsi/lbx* special file names to provide the server when you issue a *DEFINE PATH* command.

Installing zSeries Linux fibre-channel adapter (zfcp) device drivers

The zSeries Linux fibre-channel adapter (*zfcp*) device driver is a special adapter driver on the IBM zSeries system. It provides a standard SCSI interface for up level SCSI device drivers in Linux operating systems on zSeries platforms. It is used with two support modules - *qdio* for *qdio* devices and *scsi_mod*, for general SCSI support.

Without the *zfcp* driver, the system cannot detect any SCSI device attached on the zSeries systems. Tivoli Storage Manager and IBM tape device drivers can run on zSeries platforms with Linux operating systems in both 31-bit and 64-bit environments, and support most Original Equipment Manufacturer (OEM) and IBM tape devices with fibre and SCSI interfaces.

In order to function properly on the zSeries system, IBM tape device drivers and Linux SCSI generic drivers should be loaded to the kernel *after* the two support modules and the zfcplib driver have been loaded:

1. Load the two modules in this order:
 - a. qdio module
 - b. scsi_mod module
2. Perform Fibre Channel Protocol (FCP) mapping prior to installing the driver.
3. Install the zfcplib driver.
4. Load the Linux SCSI generic (sg) driver or IBM tape device driver.

For further details, see the IBM Redpaper, *Getting Started with zSeries Fibre Channel Protocol*, which is available at <http://www.ibm.com/redbooks/>

There are some differences for the zfcplib drivers on Linux kernels 2.4.x and 2.6.x. See the zfcplib driver release notes for further details.

Tivoli Storage Manager Support for multipath I/O with IBM tape devices

Multipath I/O is the use of different paths to get to the same physical device (for example, through multiple host bus adapters, switches, and so on). Multipathing helps ensure that there is no single point of failure.

The IBM tape device driver provides multipathing support so that, if one path fails, the Tivoli Storage Manager server can use a different path to access data on a storage device. The failure and transition to a different path are undetected by the running server or by a storage agent. The IBM tape device driver also uses multipath I/O to provide dynamic load balancing for enhanced I/O performance.

To provide redundant paths for SCSI devices, each device must be connected to two or more HBA ports on a SCSI or FC Host Bus Adapter, or to different SCSI or FC Host Bus Adapters. If multipath I/O is enabled and a permanent error occurs on one path (such as a malfunctioning HBA or cable), device drivers provide automatic path failover to an alternate path.

1. Add the following line to the `/sbin/init.d/atdd` file:

```
DPF_KEYS="key1; key2; key3"
```
2. Enter the `/opt/atdd/bin/atdd_claim` command.
3. Enter the `ioscan -FunC tape` command.

Multipath I/O for medium changers and tape drives is not enabled automatically when the device driver is installed. To enable multipath I/O, follow these steps:

1. Enter the `IBMtape stop` command.
2. Enter the `rmmod IBMtape` command.
3. Add the following line to the `/etc/modprobe.conf.local` file for 2.6.x kernels and to the `/etc/modules.conf` file for 2.4.x kernels:

```
options IBMtape alternate_pathing=1
```
4. (*IBM LTO tape drives only*) Add the `dfp_keys` option. This option is used to enable multipath failover. Each drive has its own key, for example:

```
options IBMtape alternate_pathing=1 dpf_keys="key1;key2;key3;..."
```
5. Enter the `depmod` command.
6. Enter the `modprobe IBMtape` command.

7. Enter the **IBMtaped** command.

When multipath I/O is enabled for a logical device, it remains enabled until the device is deleted or the support is unconfigured.

1. Add the line `dpf_support=1` at top of or at the end of a device stanza in the `IBMtape.conf` file, for example:

```
name="IBMtape" class="scsi" target=1 lun=0 dpf_support=1
```

2. (*IBM LTO tape drives only*) Add the parameter **dfp_keys** at the top of `IBMtape.conf` file, for example:

```
dpf_keys="key1, key2, key3, ..."
```

3. Enter the `/opt/IBMtape/tmd -s` command.
4. Enter the `rem_drv IBMtape` command.
5. Enter the `add_drv -m '0666 bin bin' IBMtape` command.
6. Enter the `/opt/IBMtape/tmd` command.

After multipath I/O has been enabled, the IBM tape device driver detects all paths for a device on the host system. One path is designated as the primary path. The rest of the paths are alternate paths. (The maximum number of alternate paths for a device is 16.) For each path, the IBM tape device driver creates a file with a unique name. When specifying a path from a source to a destination (for example, from the Tivoli Storage Manager server to a tape drive) using the `DEFINE PATH` command, specify the name of the special file associated with the primary path as the value of the `DEVICE` parameter.

To display the names of special file for IBM tape drives and medium changers, use the `ls -l /dev/IBMx`, where *x* is the serial number of the device. (You can also enter the `cat /proc/scsi/IBMtape` command for tape drives.) As shown in the `IBMtape` file, primary paths and alternate paths are identified by "Primary" and "Alternate," respectively:

Number	Model	SN	HBA	FO Path
0	03592	IBM1234567	qla2xxx	Primary
1	03592	IBM1234567	qla2xxx	Alternate

The name of the special file associated with the primary path for this tape drive is `/dev/IBMtape0`. Specify `/dev/IBMtape0` as the value of the `DEVICE` parameter in the `DEFINE PATH` command for this device.

To obtain the names of the special files associated with the primary paths for all medium changers configured on the system, issue the `cat /proc/scsi/IBMchanger` command. From the `IBMchanger` file:

Number	Model	SN	HBA	FO Path
3	03584L22	IBM1002345	qla2xxx	Primary
4	03584L22	IBM1002345	qla2xxx	Alternate

The name of the special file associated with the primary path for this medium changer is `/dev/IBMchanger3`. Specify `/dev/IBMchanger3` as the value of the `DEVICE` parameter in the `DEFINE PATH` command for this device.

To display path-related details about a particular tape drive on the system, use the `IBMtapeutil -f /dev/IBMtapex path` command, where *x* is the number of a configured tape device. To display path-related details about a particular medium

changer on the system, use the **IBMtapeutil -f /dev/IBMchangerx path** command, where *x* is the number of a configured medium changer.

For an overview of multipath I/O and load balancing, as well as details about how to enable, disable or query the status of multipath I/O for a device, see the *IBM Tape Device Drivers Installation and User's Guide*.

Device special file names

To work with tape, medium changer, or removable media devices, Tivoli Storage Manager needs the device's special file name, which is specified when you issue the DEFINE PATH commands for drives and libraries.

When a device configures successfully, a logical file name is returned. Table 13 specifies the name of the device (or the special file name) that corresponds to the drive or library. In the examples, *x* denotes any number from 0 to 254.

Table 13. Device examples

Device	Device Special File Name
Tape drives that are supported by the Tivoli Storage Manager Passthru driver	/dev/tsm SCSI/mtx
Tape drives that are supported by the IBM device driver	/dev/IBMtapex
Optical drives that are supported by the Tivoli Storage Manager Passthru driver	/dev/tsm SCSI/opx
SCSI-attached libraries that are supported by the Tivoli Storage Manager Passthru driver	/dev/tsm SCSI/lbx
Automatic Cartridge Facility feature of the IBM 3590 B11 as a library	/dev/IBMchangerx
IBM 349X libraries	/dev/3494lib
IBM 3581, 3583, 3584 libraries	/dev/IBMchangerx
Mount point to use on REMOVABLEFILE device type (CD-ROM)	/mnt/cdrom

Note: Based on different Linux distributions on different architectures, the mount point can be different. Another possible mount point is /mnt/media/cdrom. The driver for the CD-ROM should be provided by the device vendor. The TSM Passthru driver and IBM tape drivers do not support CD-ROM devices.

Information about your system's SCSI devices

Information about the devices seen by your system is available in the file */proc/scsi/scsi*. This file contains a list of every detected SCSI device.

The following device information is available: the host number, channel number, SCSI ID, Logical Unit number, vendor, firmware level, type of device, and the SCSI mode. For example, if a system contains some StorageTek and IBM libraries, a SAN Gateway, and some Quantum DLT drives, the */proc/scsi/scsi* file will look similar to this:

```
Attached devices:
Host: scsi2 Channel: 00 Id: 00 Lun: 00
  Vendor: STK   Model: 9738   Rev: 2003
  Type: Medium Changer   ANSI SCSI revision: 02
Host: scsi2 Channel: 00 Id: 01 Lun: 02
```

Vendor: PATHLIGHT Model: SAN Gateway Rev: 32aC
Type: Unknown ANSI SCSI revision: 03
Host: scsi2 Channel: 00 Id: 01 Lun: 02
Vendor: QUANTUM Model: DLT7000 Rev: 2560
Type: Sequential-Access ANSI SCSI revision: 02
Host: scsi2 Channel: 00 Id: 01 Lun: 04
Vendor: IBM Model: 7337 Rev: 1.63
Type: Medium Changer ANSI SCSI revision: 02

Chapter 6. Configuring storage devices

You must understand the concepts and procedures for configuring storage devices with Tivoli Storage Manager in order to use them effectively.

For the most up-to-date list of supported devices and operating-system levels, see the Tivoli Storage Manager Supported Devices web site at

- http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_Linux.html

Some of the concepts and tasks described in this topic require an understanding of storage objects. For an introduction to these storage objects, see “Tivoli Storage Manager storage objects” on page 42.

Concepts:
“Device configuration planning”
“Mixed device types in libraries” on page 58
“Server options that affect storage operations” on page 66
“Impacts of device changes on the SAN” on page 131

Use the following table to locate instructions for specific tasks:

Tasks:
“Configuring manually mounted devices” on page 122
“Configuring SCSI libraries used by one server” on page 90
“Configuring SCSI libraries shared among servers on a SAN” on page 95
“Configuring IBM 3494 libraries” on page 99
“Configuring an IBM 3494 library for use by one server” on page 100
“Configuring a 3494 library with a single drive device type” on page 101
“Configuring a 3494 library with multiple drive device types” on page 102
“Configuring an ACSLS-managed library” on page 112
“Configuring IBM Tivoli Storage Manager for LAN-free data movement” on page 124
“Validating your LAN-free configuration” on page 125
“Configuring IBM Tivoli Storage Manager for NDMP operations” on page 125

Device configuration planning

Before Tivoli Storage Manager can use a removable media device, you must plan for and configure the device.

1. Plan for the device.
See “Planning for server storage” on page 65.
2. Attach the device to the server system, and ensure that the appropriate device driver is installed and configured.

For more information on attaching devices, see Chapter 5, “Using devices with the server system,” on page 79.

For more information about which device drivers to use, see “Device driver selection” on page 80.

3. Define the device to Tivoli Storage Manager.
Define the library, drive, paths, device class, storage pool, and storage volume objects. For an introduction to these objects, see “Tivoli Storage Manager storage objects” on page 42 and “Required definitions for storage devices” on page 63.

4. Define the Tivoli Storage Manager policy that links client data with media for the device.

Define or update the policy that associates clients with the pool of storage volumes and the device. For an introduction to Tivoli Storage Manager policy, see “How client data is stored” on page 5. For a description of the default policy, see “Reviewing the standard policy” on page 437.

Note: As an alternative to creating or modifying a Tivoli Storage Manager policy, you can place the new storage pool in the storage pool migration hierarchy by updating an already defined storage pool.

5. Prepare storage volumes for use by the device. At a minimum, you must label volumes for the device. For SCSI, 349X, and ACSLS libraries, add the volumes to the device's volume inventory by checking in the volumes.

For more information, see Chapter 3, “Storage device concepts,” on page 41.

Note: Each volume used by a server for any purpose must have a unique name. This applies to volumes that reside in different libraries, volumes used for storage pools, and volumes used for operations such as database backup or export.

6. Register clients to the domain associated with the policy that you defined or updated in the preceding step. For more information, see Chapter 14, “Implementing policies for client data,” on page 435.

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see “Configuring policy for direct-to-tape backups” on page 482.
- Have clients back up data to disk. The data is later migrated to tape. For details, see “Storage pool hierarchies” on page 241.

You can also configure devices using the device configuration wizard in the Administration Center. See Chapter 18, “Managing servers with the Administration Center,” on page 547 for more details.

Configuring SCSI libraries used by one server

In order to use a SCSI library, you must set up the device on the server system.

This involves the following tasks:

1. Set the appropriate SCSI ID for each drive and for the library or medium-changer.
2. Physically attach the devices to the server hardware.
3. Install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

As an example, assume you want to attach an automated SCSI library containing two drives to the server system. The library is not shared with other Tivoli Storage Manager servers or with storage agents and is typically attached to the server system via SCSI cables.

- In the first configuration, both drives in the SCSI library are the same device type. You define one device class.
- In the second configuration, the drives are different device types. You define a device class for each drive device type.

Drives with different device types are supported in a single library if you define a device class for each type of drive. If you are configuring this way, you must include the specific format for the drive's device type by using the `FORMAT` parameter with a value other than `DRIVE`.

For details, see “Attaching an automated library device” on page 79 and “Device driver selection” on page 80.

Configuration with a single drive device type

You can configure libraries with single drive device types, for example, a SCSI library containing two DLT tape drives.

1. Define a SCSI library named `AUTODTLIB`. The library type is `SCSI` because the library is a SCSI-controlled automated library. Enter the following command:

```
define library autodtlib libtype=scsi
```

Note: If you have a SCSI library with a barcode reader and you would like to automatically label tapes before they are checked in, you can specify the following:

```
define library autodtlib libtype=scsi autolabel=yes
```

2. Define a path from the server to the library:

```
define path server1 autodtlib srctype=server desttype=library  
device=/dev/tmscsi/1b3
```

The `DEVICE` parameter specifies the device driver's name for the library, which is the special file name.

See “Defining libraries” on page 126 and “SCSI libraries” on page 44. For more information about paths, see “Defining paths” on page 129.

3. Define the drives in the library. Both drives belong to the `AUTODTLIB` library.

```
define drive autodtlib drive01  
define drive autodtlib drive02
```

This example uses the default address for the drive's element address. The server obtains the element address from the drive itself at the time that the path is defined.

The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can have the server obtain the element address from the drive itself at the time that the path is defined, or you can specify the element address when you define the drive.

Depending on the capabilities of the library, the server may not be able to automatically detect the element address. In this case you must supply the element address when you define the drive. If you need the element address, check the Tivoli Storage Manager device driver support web site at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

See “Defining drives” on page 127. For more information about paths, see “Defining paths” on page 129.

4. Define a path from the server to each drive:

```
define path server1 drive01 srctype=server desttype=drive
library=autodlplib device=/dev/tmscsi/mt4
define path server1 drive02 srctype=server desttype=drive
library=autodlplib device=/dev/tmscsi/mt5
```

The DEVICE parameter specifies the device driver's name for the drive, which is the device special file name.

For more about device special file names, see

- “Device special file names” on page 87

If you did not include the element address when you defined the drive, the server now queries the library to obtain the default element address for the drive.

For more information about paths, see “Defining paths” on page 129.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. Use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical. For example, to classify two drives in the AUTODLTLIB library, use the following command to define a device class named AUTODLT_CLASS:

```
define devclass autodlt_class library=autodlplib devtype=dlt format=drive
```

See “Defining tape and optical device classes” on page 201.

6. Verify your definitions by issuing the following commands:

```
query library
query drive
query path
query devclass
```

See “Requesting information about libraries” on page 155, “Requesting information about drives” on page 156, “Obtaining information about device classes” on page 216, and “Requesting information about paths” on page 164.

7. Define a storage pool named AUTODLT_POOL associated with the device class named AUTODLT_CLASS.

```
define stgpool autodlt_pool autodlt_class maxscratch=20
```

Key choices:

- a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

For more information, see “Defining storage pools” on page 227.

Configuration with multiple drive device types

You can configure a library with multiple drive device types, for example, a StorageTek L40 library that contains one DLT drive and one LTO Ultrium drive.

1. Define a SCSI library named MIXEDLIB. The library type is *SCSI* because the library is a SCSI-controlled automated library. Enter the following command:

```
define library mixedlib libtype=scsi
```

2. Define a path from the server to the library:

```
define path server1 mixedlib srctype=server desttype=library  
device=/dev/tsm SCSI/1b3
```

The *DEVICE* parameter specifies the device driver's name for the drive, which is the device special file name.

For more about device special file names, see:

“Device special file names” on page 87

See “Defining libraries” on page 126 and “SCSI libraries” on page 44. For more information about paths, see “Defining paths” on page 129.

3. Define the drives in the library:

```
define drive mixedlib dlt1  
define drive mixedlib lto1
```

Both drives belong to the MIXEDLIB library.

This example uses the default for the drive's element address. The server obtains the element address from the drive itself at the time that the path is defined.

The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can have the server obtain the element address from the drive itself at the time that the path is defined, or you can specify the element address when you define the drive.

Depending on the capabilities of the library, the server may not be able to automatically detect the element address. In this case you must supply the element address when you define the drive. If you need the element address, check the Tivoli Storage Manager device driver support web site at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

See “Defining drives” on page 127. For more information about paths, see “Defining paths” on page 129.

4. Define a path from the server to each drive:

```
define path server1 dlt1 srctype=server desttype=drive  
library=mixedlib device=/dev/tsm SCSI/mt4  
define path server1 lto1 srctype=server desttype=drive  
library=mixedlib device=/dev/tsm SCSI/mt5
```

The *DEVICE* parameter specifies the device driver's name for the drive, which is the device special file name.

For more about device special file names, see:

“Device special file names” on page 87

If you did not include the element address when you defined the drive, the server now queries the library to obtain the element address for the drive.

For more information about paths, see “Defining paths” on page 129.

5. Classify the drives according to type by defining Tivoli Storage Manager device classes, which specify the recording formats of the drives.

Important: Do not use the DRIVE format, which is the default. Because the drives are different types, Tivoli Storage Manager uses the format specification to select a drive. The results of using the DRIVE format in a mixed media library are unpredictable.

```
define devclass dlt_class library=mixedlib devtype=dlt format=dlt40
define devclass lto_class library=mixedlib devtype=lto format=ultriumc
```

See “Defining tape and optical device classes” on page 201.

6. Verify your definitions by issuing the following commands:

```
query library
query drive
query path
query devclass
```

See “Requesting information about libraries” on page 155, “Requesting information about drives” on page 156, “Obtaining information about device classes” on page 216, and “Requesting information about paths” on page 164.

7. Define storage pools associated with the device classes. For example:

```
define stgpool lto_pool lto_class maxscratch=20
define stgpool dlt_pool dlt_class maxscratch=20
```

Key choices:

- a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

For more information, see “Defining storage pools” on page 227.

Checking in and labeling library volumes

You should always ensure that enough labeled volumes in the library are available to the server so that you do not run out during an operation such as client backup. Label and set aside extra scratch volumes for any potential recovery operations you might have later.

Note:

1. If you use the autolabel=yes parameter on the DEFINE LIBRARY command, you will not need to label tapes before you check them in.
2. If a volume has an entry in volume history, you cannot check it in as a scratch volume.
3. Tivoli Storage Manager only accepts tapes labeled with IBM standard labels. IBM standard labels are similar to ANSI Standard X3.27 labels except that the IBM standard labels are written in EBCDIC. For a list of IBM media sales contacts who can provide compatible tapes, visit the IBM Web site. If you are using non-IBM storage devices and media, consult your tape-cartridge distributor.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries but that are used by the same server.

The procedures for volume checkin and labeling are the same whether the library contains drives of a single device type, or drives of multiple device types.

To check in and label volumes, do the following:

1. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.
 - Check in volumes that are already labeled:

```
checkin libvolume autodlplib search=yes status=scratch checklabel=barcode
```
 - Label and check in volumes:

```
label libvolume autodlplib search=yes labelsource=barcode checkin=scratch
```
2. Depending on whether you use scratch volumes or private volumes, perform one of the following steps:
 - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
 - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See “Defining storage pool volumes” on page 237.

Configuring SCSI libraries shared among servers on a SAN

Using a SAN with Tivoli Storage Manager allows you to take advantage of certain functions.

- Multiple Tivoli Storage Manager servers share storage devices.
- Tivoli Storage Manager client systems directly access storage devices, both tape libraries and disk storage, that are defined to a Tivoli Storage Manager server (LAN-free data movement). Storage agents installed and configured on the client systems perform the data movement. See “Configuring IBM Tivoli Storage Manager for LAN-free data movement” on page 124.

The following tasks are required for Tivoli Storage Manager servers to share library devices on a SAN:

1. Set up server-to-server communications.
2. Set up the device on the server systems.
3. Set up the library on the Tivoli Storage Manager server that is going to act as the library manager. In the example used for this section, the library manager server is named ASTRO.
4. Set up the library on the Tivoli Storage Manager server that is going to act as the library client. In the example used for this section, the library client server is named JUDY.

Setting up server communications

Before Tivoli Storage Manager servers can share a storage device on a SAN, you must set up server communications.

This requires configuring each server as you would for Enterprise Administration, which means you define the servers to each other using the cross-define function. See “Setting up communications among servers” on page 676 for details. For a discussion about the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see “Operations with shared libraries” on page 150.

Note: Set up each server with a unique name.

“Operations with shared libraries” on page 150

Set up the device on the server systems and the SAN

To use a device, you must first set it up on the server system.

This involves the following tasks:

1. Set the appropriate SCSI ID for each drive and for the library or medium-changer.
2. Physically attach the devices to the SAN.
3. On each server system that will access the library and drives, install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see “Attaching an automated library device” on page 79 and “Device driver selection” on page 80.

Setting up the library manager server

You must set up the server that is the library manager before you set up servers that are the library clients.

Note: You can configure a SCSI library so that it contains all drives of the same device type or so that it contains drives of different device types. You can modify the procedure described for configuring a library for use by one server (“Configuration with multiple drive device types” on page 93) and use it for configuring a shared library.

Use the following procedure as an example of how to set up a server as a library manager. The server is named ASTRO.

1. Define a shared SCSI library named SANGROUP:

```
define library sangroup libtype=scsi shared=yes
```

This example uses the default for the library's serial number, which is to have the server obtain the serial number from the library itself at the time that the path is defined. Depending on the capabilities of the library, the server may not be able to automatically detect the serial number. In this case, the server will not record a serial number for the library, and will not be able to confirm the identity of the library when you define the path or when the server uses the library.

2. Define a path from the server to the library:

```
define path astro sangroup srctype=server desttype=library
  device=/dev/IBMchanger1
```

If you did not include the serial number when you defined the library, the server now queries the library to obtain the serial number information. If you did include the serial number when you defined the library, the server verifies what you defined and issues a message if there is a mismatch.

For more information about paths, see “Defining paths” on page 129.

3. Define the drives in the library:

```
define drive sangroup drivea
define drive sangroup driveb
```

This example uses the default for the drive's serial number. The server obtains the serial number from the drive itself at the time that the path is defined. Depending on the capabilities of the drive, the server may not be able to automatically detect the serial number. In this case, the server will not record a serial number for the drive, and will not be able to confirm the identity of the drive when you define the path or when the server uses the drive.

This example also uses the default for the drive's element address, which is to have the server obtain the element address from the drive itself at the time that the path is defined.

The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can have the server obtain the element address from the drive itself at the time that the path is defined, or you can specify the element address when you define the drive.

Depending on the capabilities of the library, the server may not be able to automatically detect the element address. In this case you must supply the element address when you define the drive. If you need the element address, check the Tivoli Storage Manager device driver support web site at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

4. Define a path from the server to each drive:

```
define path astro drivea srctype=server desttype=drive
  library=sangroup device=/dev/IBMtape4
define path astro driveb srctype=server desttype=drive
  library=sangroup device=/dev/IBMtape5
```

If you did not include the serial number or element address when you defined the drive, the server now queries the drive or the library to obtain this information.

For more information about paths, see “Defining paths” on page 129.

5. Define all the device classes that are associated with the shared library.

```
define devclass tape library=sangroup devtype=lto
```

6. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.

- Check in volumes that are already labeled:

```
checkin libvolume sangroup search=yes status=scratch checklabel=barcode
```

- Label and check in volumes:

```
label libvolume sangroup search=yes labelsource=barcode checkin=scratch
```

7. Set up any required storage pools for the shared library with a maximum of 50 scratch volumes.

```
define stgpool backtape tape maxscratch=50
```

Setting up the library client servers

You must set up the server that is the library manager server before you set up the servers that are the library clients.

Use the following sample procedure for each Tivoli Storage Manager server that will be a library client. The library client server is named JUDY. With the exception of one step, perform the procedure from the library client servers.

1. Define the server that is the library manager:

```
define server astro serverpassword=secret hladdress=9.115.3.45 lladdress=1580
  crossdefine=yes
```

2. Define the shared library named SANGROUP, and identify the library manager server's name as the primary library manager. Ensure that the library name is the same as the library name on the library manager:

```
define library sangroup libtype=shared primarylibmanager=astro
```

3. *Perform this step from the library manager.* Define a path from the library client server to each drive that the library client server will be allowed to access. The device name should reflect the way the *library client* system sees the device. There must be a path defined from the library manager to each drive in order for the library client to use the drive.

In general, it is best practice for any library sharing setup to have all drive path definitions created for the library manager also created for each library client. For example, if the library manager defines three drives, the library client should also define three drives. If you want to limit the number of drives a library client can use at a time, use the MOUNTLIMIT parameter on the library client's device class instead of limiting the drive path definitions for the library client.

The following is an example of how to define a path from the library manager to a drive in the library client:

```
define path judy drivea srctype=server desttype=drive
  library=sangroup device=/dev/IBMtape6
define path judy driveb srctype=server desttype=drive
  library=sangroup device=/dev/IBMtape7
```

For more information about paths, see “Defining paths” on page 129.

4. *Return to the library client for the remaining steps.* Define all the device classes that are associated with the shared library.

```
define devclass tape library=sangroup devtype=lto
```

Set the parameters for the device class the same on the library client as on the library manager. A good practice is to make the device class names the same on both servers, but this is not required.

The device class parameters specified on the library manager server override those specified for the library client. This is true whether or not the device class names are the same on both servers. If the device class names are different, the library manager uses the parameters specified in a device class that matches the device type specified for the library client.

Note: If a library client requires a setting that is different from what is specified in the library manager's device class (for example, a different mount limit), do the following:

- a. Create an additional device class on the library manager server. Specify the parameter settings you want the library client to use.
- b. Create a device class on the library client with the same name and device type as the new device class you created on the library server.

5. Define the storage pool, BACKTAPE, which will use the shared library.
`define stgpool backtape tape maxscratch=50`
6. Repeat this procedure to define additional servers as library clients.

Storing client data on devices

After you have attached and defined your devices, you can store client data in two ways.

- Have clients back up data directly to tape. For details, see “Configuring policy for direct-to-tape backups” on page 482.
- Have clients back up data to disk. The data is later migrated to tape. For details, see “Storage pool hierarchies” on page 241.

Migrating DEFINE DRIVE and DEFINE PATH definitions to the library manager server

DEFINE DRIVE and DEFINE PATH definitions should be migrated to the library manager server.

Definitions from Tivoli Storage Manager versions 4.2 and lower should be migrated over to the library manager server by following these steps:

1. Retrieve the path information from the library client through the device configuration (DEVCONFIG) file or the QUERY PATH command.
2. Use the data gathered in step 1 to issue the DEFINE PATH commands on the library manager server.
3. Delete the path definitions on the library client with DELETE PATH.
4. Delete the drive definitions on the library client with DELETE DRIVE.

Configuring IBM 3494 libraries

An IBM 3494 library can be added only by using Tivoli Storage Manager commands. One or more Tivoli Storage Manager servers can use a single IBM 3494 library.

See the following sections:

- “Configuring an IBM 3494 library for use by one server” on page 100
- “Sharing an IBM 3494 library among servers” on page 105
- “Migrating a shared IBM 3494 library to a library manager” on page 107
- “Sharing an IBM 3494 library by static partitioning of drives” on page 108

See also “Categories in an IBM 3494 library.”

Categories in an IBM 3494 library

The library manager built into the IBM 3494 library tracks the category number of each volume in the library. A single category number identifies all volumes used for the same purpose or application. Category numbers are useful when multiple systems share the resources of a single library.

Attention: If other systems or other Tivoli Storage Manager servers connect to the same 3494 library, each must use a unique set of category numbers. Otherwise, two or more systems may try to use the same volume, and cause corruption or loss of data.

Typically, a software application that uses a 3494 library uses volumes in one or more categories that are reserved for that application. To avoid loss of data, each application sharing the library must have unique categories. When you define a 3494 library to the server, you can use the PRIVATECATEGORY and SCRATCHCATEGORY parameters to specify the category numbers for private and scratch Tivoli Storage Manager volumes in that library. If the volumes are IBM 3592 WORM (write once, read many) volumes, you can use the WORMSCRATCHCATEGORY parameter to specify category numbers for scratch WORM volumes in the library. See “Tivoli Storage Manager volumes” on page 50 for more information on private, scratch, and scratch WORM volumes.

When a volume is first inserted into the library, either manually or automatically at the convenience I/O station, the volume is assigned to the insert category (X'FF00'). A software application such as Tivoli Storage Manager can contact the library manager to change a volume's category number. For Tivoli Storage Manager, you use the CHECKIN LIBVOLUME command (see “Checking new volumes into a library” on page 137).

The Tivoli Storage Manager server only supports 3590 and 3592 tape drives in an IBM 3494 library. The server reserves two different categories for each 3494 library object. The categories are private and scratch.

Note: A 349X library object will now only contain one device type (3590 or 3592) of drives. Thus, if you have 3590s and 3592s in your 349X library, you need to define two library objects: one for your 3590 drives and one for your 3592 drives. Each of these library objects will have the same device parameter when defining their paths.

When you define a 3494 library, you can specify the category numbers for volumes that the server owns in that library by using the PRIVATECATEGORY, SCRATCHCATEGORY, and if the volumes are IBM 3592 WORM volumes, the WORMSCRATCHCATEGORY parameters. For example:

Note: The default values for the categories may be acceptable in most cases. However, if you connect other systems or Tivoli Storage Manager servers to a single 3494 library, ensure that each uses unique category numbers. Otherwise, two or more systems may try to use the same volume, and cause a corruption or loss of data.

For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see “Operations with shared libraries” on page 150.

Configuring an IBM 3494 library for use by one server

In the following example, an IBM 3494 library containing two drives is configured for use by one Tivoli Storage Manager server.

You must first set up the IBM 3494 library on the server system. This involves the following tasks:

1. Set the symbolic name for the library in the configuration file for the library device driver (/etc/ibmatl.conf). This procedure is described in *IBM Tape Device Drivers Installation and User's Guide*.
2. Physically attach the devices to the server hardware or the SAN.
3. Install and configure the appropriate device drivers for the devices on the server that will use the library and drives.

4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see the following topics:

- “Attaching an automated library device” on page 79
- “Device driver selection” on page 80

There are two possible configurations:

- In the first configuration, both drives in the library are the same device type. See “Configuring a 3494 library with a single drive device type”
- In the second configuration, the drives are different device types.
Drives with different device types (or different generations of drives) are supported in a single physical library if you define one library to Tivoli Storage Manager for each type of drive (or generation of drive). For example, if you have two device types, such as 3590E and 3590H (or two generations of drives of the same device type), define two libraries. Then define drives and device classes for each library. In each device class definition, you can use the `FORMAT` parameter with a value of `DRIVE`, if you choose. See “Configuring a 3494 library with multiple drive device types” on page 102.

Configuring a 3494 library with a single drive device type

In this example, the 3494 library contains two IBM 3590 tape drives.

1. Define a 3494 library named 3494LIB:

```
define library 3494lib libtype=349x
```

2. Define a path from the server to the library:

```
define path server1 3494lib srctype=server desttype=library  
device=library1
```

The `DEVICE` parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (`/etc/ibmatl.conf`).

See the following topics:

“Defining libraries” on page 126

“SCSI libraries” on page 44

For more information about paths, see “Defining paths” on page 129.

3. Define the drives in the library:

```
define drive 3494lib drive01  
define drive 3494lib drive02
```

Both drives belong to the 3494LIB library.

See “Defining drives” on page 127.

4. Define a path from the server to each drive:

```
define path server1 drive01 srctype=server desttype=drive  
library=3494lib device=/dev/IBMtape0  
define path server1 drive02 srctype=server desttype=drive  
library=3494lib device=/dev/IBMtape1
```

The `DEVICE` parameter gives the device special file name for the drive. For more about device names, see the following topics:

“Device special file names” on page 87

.

For more information about paths, see “Defining paths” on page 129.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. For example, for the two 3590 drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

```
define devclass 3494_class library=3494lib devtype=3590 format=drive
```

This example uses FORMAT=DRIVE as the recording format because both drives associated with the device class use the same recording format; both are 3590 drives. If instead one drive is a 3590 and one is a 3590E, you need to use specific recording formats when defining the device classes. See “Configuring a 3494 library with multiple drive device types.”

See also “Defining tape and optical device classes” on page 201.

6. Verify your definitions by issuing the following commands:

```
query library
query drive
query path
query devclass
```

For details, see the following topics:

“Requesting information about drives” on page 156

“Obtaining information about device classes” on page 216

“Requesting information about paths” on page 164

“Requesting information about libraries” on page 155

7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS.

```
define stgpool 3494_pool 3494_class maxscratch=20
```

Key choices:

- a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. The default setting for primary storage pools is collocation by group. The default for copy storage pools and active-data pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

For more information, see “Defining storage pools” on page 227.

Configuring a 3494 library with multiple drive device types

In this example, the 3494 library contains two IBM 3590E tape drives and two IBM 3590H tape drives.

1. Define *two* libraries, one for each type of drive. For example, to define 3590ELIB and 3590HLIB enter the following commands:

```
define library 3590elib libtype=349x scratchcategory=301 privatecategory=300
define library 3590hlib libtype=349x scratchcategory=401 privatecategory=400
```

See “Defining libraries” on page 126.

Note: Specify scratch and private categories explicitly. If you accept the category defaults for both library definitions, different types of media will be assigned to the same categories.

2. Define a path from the server to each library:

```
define path server1 3590elib srctype=server desttype=library device=library1
define path server1 3590hlib srctype=server desttype=library device=library1
```

The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

For more information about paths, see “Defining paths” on page 129.

3. Define the drives, ensuring that they are associated with the appropriate libraries.

- Define the 3590E drives to 3590ELIB.

```
define drive 3590elib 3590e_drive1
define drive 3590elib 3590e_drive2
```

- Define the 3590H drives to 3590HLIB.

```
define drive 3590hlib 3590h_drive3
define drive 3590hlib 3590h_drive4
```

Note: Tivoli Storage Manager does not prevent you from associating a drive with the wrong library.

See “Defining drives” on page 127.

4. Define a path from the server to each drive. Ensure that you specify the correct library.

- For the 3590E drives:

```
define path server1 3590e_drive1 srctype=server desttype=drive
library=3590elib device=/IBMtape0
define path server1 3590e_drive2 srctype=server desttype=drive
library=3590elib device=/IBMtape1
```

- For the 3590H drives:

```
define path server1 3590h_drive3 srctype=server desttype=drive
library=3590hlib device=/IBMtape2
define path server1 3590h_drive4 srctype=server desttype=drive
library=3590hlib device=/IBMtape30
```

The DEVICE parameter gives the device special file name for the drive. For more about device names, see

“Device special file names” on page 87

For more information about paths, see “Defining paths” on page 129

5. Classify the drives according to type by defining Tivoli Storage Manager device classes, which specify the recording formats of the drives. Because there are separate libraries, you can enter a specific recording format, for example 3590H, or you can enter DRIVE.

```
define devclass 3590e_class library=3590elib devtype=3590 format=3590e
```

```
define devclass 3590h_class library=3590hlib devtype=3590 format=3590h
```

See “Defining tape and optical device classes” on page 201.

6. To check what you have defined, enter the following commands:

```
query library
query drive
query path
query devclass
```

See the following topics:

- “Requesting information about libraries” on page 155
- “Obtaining information about device classes” on page 216
- “Requesting information about paths” on page 164
- “Requesting information about drives” on page 156

7. Create the storage pools to use the devices in the device classes you just defined. For example, define a storage pool named 3590EPOOL associated with the device class 3490E_CLASS, and 3590HPOOL associated with the device class 3590H_CLASS:

```
define stgpool 3590epool 3590e_class maxscratch=20

define stgpool 3590hpool 3590h_class maxscratch=20
```

Key choices:

- a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. The default setting for primary storage pools is collocation by group. The default for copy storage pools and active-data pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

For more information, see “Defining storage pools” on page 227.

Checking in and labeling 3494 library volumes

Ensure that enough volumes in the library are available to the server. Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup. Label and set aside extra scratch volumes for any potential recovery operations you might have later.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries.

The procedures for volume check-in and labeling are the same whether the library contains drives of a single device type, or drives of multiple device types.

Note: If your library has drives of multiple device types, you defined *two* libraries to the Tivoli Storage Manager server in the procedure in “Configuring a 3494 library with multiple drive device types” on page 102. The two Tivoli Storage Manager libraries represent the *one* physical library. The check-in process finds all available volumes that are not already checked in. You must check in media *separately* to each defined library. Ensure that you check in volumes to the correct Tivoli Storage Manager library.

Do the following:

1. Check in the library inventory. The following shows two examples.
 - Check in volumes that are already labeled:

```
checkin libvolume 3494lib search=yes status=scratch checklabel=no
```
 - Label and check in volumes:

```
label libvolume 3494lib search=yes checkin=scratch
```

2. Depending on whether you use scratch volumes or private volumes, do one of the following:
 - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
 - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See “Defining storage pool volumes” on page 237.

For more information about checking in volumes, see “Checking new volumes into a library” on page 137.

Sharing an IBM 3494 library among servers

Sharing an IBM 3494 library requires one of the following environments.

- The library must be on a SAN.
- Through the use of the dual ports on 3590 drives in the library, the drives and the library are connected to two systems on which Tivoli Storage Manager servers run.

The following tasks are required for Tivoli Storage Manager servers to share library devices over a SAN:

1. Set up server-to-server communications.
2. Set up the device on the server systems.
3. Set up the library on the library manager server. In the following example, the library manager server is named MANAGER.
4. Set up the library on the library client server. In the following example, the library client server is named CLIENT.

See “Categories in an IBM 3494 library” on page 99 for additional information about configuring 3494 libraries.

Setting up a 3494 library on the server system and SAN

You must first set up the device on the server system, which involves certain tasks.

1. Set the symbolic name for the library in the configuration file for the library device driver. This procedure is described in *IBM Tape Device Drivers Installation and User's Guide*.
2. Physically attach the devices to the SAN or to the server hardware.
3. On each server system that will access the library and drives, install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see the following topics:

- “Attaching an automated library device” on page 79
- “Device driver selection” on page 80

Note: You can also configure a 3494 library so that it contains drives of multiple device types or different generations of drives of the same device type. The procedure for working with multiple drive device types is similar to the one

described for a LAN in “Configuring a 3494 library with multiple drive device types” on page 102. For details about mixing generations of drives, see “Defining 3592 device classes” on page 204 and “Defining LTO device classes” on page 211.

Setting up the 3494 library manager server

Use the following procedure as an example of how to set up a Tivoli Storage Manager server as a library manager named MANAGER.

1. Define a 3494 library named 3494SAN:

```
define library 3494san libtype=349x shared=yes
```

2. Define a path from the server to the library:

```
define path manager 3494san srctype=server desttype=library
device=library1
```

The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

For more information about paths, see “Defining paths” on page 129.

3. Define the drives in the library:

```
define drive 3494san drivea
define drive 3494san driveb
```

4. Define a path from the server to each drive:

```
define path manager drivea srctype=server desttype=drive library=3494san
device=/dev/IBMtape0
define path manager driveb srctype=server desttype=drive library=3494san
device=/dev/IBMtape1
```

For more information about paths, see “Defining paths” on page 129.

5. Define all the device classes that are associated with the shared library.

```
define devclass 3494_class library=3494san devtype=3590
```

6. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.

To check in volumes that are already labeled, use the following command:

```
checkin libvolume 3494san search=yes status=scratch checklabel=no
```

To label and check in the volumes, use the following command:

```
label libvolume 3494san checkin=scratch search=yes
```

7. Set any required storage pools for the shared library with a maximum of 50 scratch volumes.

```
define stgpool 3494_sanpool tape maxscratch=50
```

Setting up the 3494 library client servers

Use the following sample procedure for each Tivoli Storage Manager server that will be a library client server.

1. Define the server that is the library manager:

```
define server manager serverpassword=secret haddress=9.115.3.45 laddress=1580
crossdefine=yes
```

2. Define a shared library named 3494SAN, and identify the library manager:

Note: Ensure that the library name agrees with the library name on the library manager.

```
define library 3494san libtype=shared primarylibmanager=manager
```

3. *Perform this step from the library manager.* Define a path from the library client server to each drive that the library client server will be allowed to access. The device name should reflect the way the *library client* system sees the device.

There must be a path defined from the library manager to each drive in order for the library client to use the drive. The following is an example of how to define a path:

```
define path client drivea srctype=server desttype=drive
  library=3494san device=/dev/IBMtape0
define path client driveb srctype=server desttype=drive
  library=3494san device=/dev/IBMtape1
```

For more information about paths, see “Defining paths” on page 129.

4. *Return to the library client for the remaining steps.* Define all the device classes that are associated with the shared library.

```
define devclass 3494_class library=3494san devtype=3590
```

Set the parameters for the device class the same on the library client as on the library manager. Making the device class names the same on both servers is also a good practice, but is not required.

The device class parameters specified on the library manager server override those specified for the library client. This is true whether or not the device class names are the same on both servers. If the device class names are different, the library manager uses the parameters specified in a device class that matches the device type specified for the library client.

Note: If a library client requires a setting that is different from what is specified in the library manager's device class (for example, a different mount limit), do the following:

- a. Create an additional device class on the library manager server. Specify the parameter settings you want the library client to use.
 - b. Create a device class on the library client with the same name and device type as the new device class you created on the library server.
5. Define the storage pool, BACKTAPE, that will use the shared library.

```
define stgpool backtape 3494_class maxscratch=50
```
 6. Repeat this procedure to define additional servers as library clients. For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see “Operations with shared libraries” on page 150

Migrating a shared IBM 3494 library to a library manager

If you have been sharing an IBM 3494 library among Tivoli Storage Manager servers by using the 3494SHARED option in the dsmserv.opt file, you can migrate to sharing the library by using a library manager and library clients.

To help ensure a smoother migration and to ensure that all tape volumes that are being used by the servers get associated with the correct servers, perform the following migration procedure.

1. Do the following on *each* server that is sharing the 3494 library:
 - a. Update the storage pools using the UPDATE STGPOOL command. Set the value for the HIGHMIG and LOWMIG parameters to 100%.
 - b. Stop the server by issuing the HALT command.
 - c. Edit the dsmserv.opt file and make the following changes:
 - 1) Comment out the 3494SHARED YES option line
 - 2) Activate the DISABLESCHEDULES YES option line if it is not active
 - 3) Activate the EXPINTERVAL X option line if it is not active and change its value to 0, as follows:

```
EXPINTERVAL 0
```

- d. Start the server.
- e. Enter the following Tivoli Storage Manager command:

```
disable sessions
```
2. Set up the library manager on the Tivoli Storage Manager server of your choice (see “Setting up server communications” on page 96 and “Setting up the library manager server” on page 96).
3. Do the following on the remaining servers (the library clients):
 - a. Save the volume history file.
 - b. Check out all the volumes in the library inventory. Use the CHECKOUT LIBVOLUME command with REMOVE=NO.
 - c. Follow the library client setup procedure (“Setting up the 3494 library client servers” on page 106).
4. Do the following on the library manager server:
 - a. Check in each library client's volumes. Use the CHECKIN LIBVOLUME command with the following parameter settings:
 - STATUS=PRIVATE
 - OWNER=<library client name>

Note: You can use the saved volume history files from the library clients as a guide.
 - b. Check in any remaining volumes as scratch volumes. Use the CHECKIN LIBVOLUME command with STATUS=SCRATCH.
5. Halt all the servers.
6. Edit the dsmserv.opt file and comment out the following lines in the file:

```
DISABLESCHEDULES YES
EXPINTERVAL 0
```
7. Start the servers.

Sharing an IBM 3494 library by static partitioning of drives

If your IBM 3494 library is not on a SAN, you can use partitioning to share that library among Tivoli Storage Manager servers.

Tivoli Storage Manager uses the capability of the 3494 library manager, which allows you to partition a library between multiple Tivoli Storage Manager servers. Library partitioning differs from library sharing on a SAN in that with partitioning, there are no Tivoli Storage Manager library managers or library clients.

When you partition a library on a LAN, each server has its own access to the same library. For each server, you define a library with tape volume categories unique to that server. Each drive that resides in the library is defined to only one server. Each server can then access only those drives it has been assigned. As a result, library partitioning does not allow dynamic sharing of drives or tape volumes because they are pre-assigned to different servers using different names and category codes.

In the following example, an IBM 3494 library containing four drives is attached to a Tivoli Storage Manager server named ASTRO and to another Tivoli Storage Manager server named JUDY.

Note: Tivoli Storage Manager can also share the drives in a 3494 library with other servers by enabling the 3494SHARED server option. When this option is enabled,

you can define all of the drives in a 3494 library to multiple servers, if there are SCSI connections from all drives to the systems on which the servers are running. This type of configuration is not recommended, however, because when this type of sharing takes place there is a risk of contention between servers for drive usage, and operations can fail.

Setting up the 3494 library on the server system

You must first set up the 3494 library on the server system.

This involves the following tasks:

1. Set the symbolic name for the library in the configuration file for the library device driver. This procedure is described in *IBM Tape Device Drivers Installation and User's Guide*.
2. Physically attach the devices to the server hardware.
3. On each server system that will access the library and drives, install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see “Attaching an automated library device” on page 79 and “Device driver selection” on page 80.

Defining 3494 library devices to the Tivoli Storage Manager server ASTRO

Complete the following steps to define the 3493 library.

1. Define the 3494 library named 3494LIB:

```
define library 3494lib libtype=349x privatecategory=400 scratchcategory=600
```

The PRIVATECATEGORY and SCRATCHCATEGORY are set differently from the default settings. See “Categories in an IBM 3494 library” on page 99.

2. Define the path from the server, ASTRO, to the library:

```
define path astro 3494lib srctype=server desttype=library
device=library1
```

The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

See “Defining libraries” on page 126 and “SCSI libraries” on page 44.

For more information about paths, see “Defining paths” on page 129.

3. Define the drives that are partitioned to server ASTRO:

```
define drive 3494lib drive1
define drive 3494lib drive2
```

4. Define the path from the server, ASTRO, to each of the drives:

```
define path astro drive1 srctype=server desttype=drive library=3494lib
device=/dev/IBMtape0
define path astro drive2 srctype=server desttype=drive library=3494lib
device=/dev/IBMtape1
```

The DEVICE parameter gives the device special file name for the drive. For more about device names, see “Device special file names” on page 87.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. For example, to classify the two drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

```
define devclass 3494_class library=3494lib devtype=3590 format=drive
```

This example uses FORMAT=DRIVE as the recording format because both drives associated with the device class use the same recording format; both are

3590 drives. If instead one drive is a 3590 and one is a 3590E, you need to use specific recording formats when defining the device classes. See “Configuring a 3494 library with multiple drive device types” on page 102.

See “Defining tape and optical device classes” on page 201.

6. Verify your definitions by issuing the following commands:

```
query library
query drive
query path
query devclass
```

See the following topics:

- “Requesting information about libraries” on page 155
- “Obtaining information about device classes” on page 216
- “Requesting information about paths” on page 164
- “Requesting information about drives” on page 156

7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS:

```
define stgpool 3494_pool 3494_class maxscratch=20
```

Key choices:

- a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. The default setting for primary storage pools is collocation by group. The default for copy storage pools and active-data pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

For more information, see “Defining storage pools” on page 227.

Defining 3494 library devices to the Tivoli Storage Manager server JUDY

The DEVICE parameter specifies the device special file for the LMCP.

1. Define the 3494 library named 3494LIB:

```
define library 3494lib libtype=3494 privatecategory=112 scratchcategory=300
```

The PRIVATECATEGORY and SCRATCHCATEGORY are defined differently than the first server's definition. See “Categories in an IBM 3494 library” on page 99.

2. Define the path from the server, JUDY, to the library:

```
define path judy 3494lib srctype=server desttype=library
device=library1
```

The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

See “Defining libraries” on page 126 and “SCSI libraries” on page 44.

For more information about paths, see “Defining paths” on page 129.

3. Define the drives that are partitioned to server JUDY:

```
define drive 3494lib drive3
define drive 3494lib drive4
```

4. Define the path from the server, JUDY, to each of the drives:

```
define path judy drive3 srctype=server desttype=drive library=3494lib
  device=/dev/IBMtape2
define path judy drive4 srctype=server desttype=drive library=3494lib
  device=/dev/IBMtape3
```

The DEVICE parameter gives the device special file name for the drive. For more about device names, see “Device special file names” on page 87.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. For example, to classify the two drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

```
define devclass 3494_class library=3494lib devtype=3590 format=drive
```

This example uses FORMAT=DRIVE as the recording format because both drives associated with the device class use the same recording format; both are 3590 drives. If instead one drive is a 3590 and one is a 3590E, you need to use specific recording formats when defining the device classes. See “Configuring a 3494 library with multiple drive device types” on page 102.

See “Defining tape and optical device classes” on page 201.

6. Verify your definitions by issuing the following commands:

```
query library
query drive
query path
query devclass
```

See the following topics:

- “Requesting information about libraries” on page 155
- “Obtaining information about device classes” on page 216
- “Requesting information about drives” on page 156

7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS.

```
define stgpool 3494_pool 3494_class maxscratch=20
```

Key choices:

- a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. The default setting for primary storage pools is collocation by group. The default for copy storage pools and active-data pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

For more information, see “Defining storage pools” on page 227.

ACSLs-Managed libraries

Tivoli Storage Manager supports tape libraries controlled by StorageTek Automated Cartridge System Library Software (ACSLs). The ACSLS library server manages the physical aspects of tape cartridge storage and retrieval.

The ACSLS client application communicates with the ACSLS library server to access tape cartridges in an automated library. Tivoli Storage Manager is one of the applications that gains access to tape cartridges by interacting with ACSLS through its client, which is known as the control path. The Tivoli Storage Manager server reads and writes data on tape cartridges by interacting directly with tape drives through the data path. The control path and the data path are two different paths.

The ACSLS client daemon must be initialized before starting the server. See the script `rc.acs_ssi` in the server's installation directory for the client daemon invocation. For detailed installation, configuration, and system administration of ACSLS, refer to the appropriate StorageTek documentation.

Configuring an ACSLS-managed library

The library ACSLS is attached to the ACSLS server, and the drives are attached to the Tivoli Storage Manager server. The ACSLS server and the Tivoli Storage Manager server must be on different systems. Refer to the ACSLS installation documentation for details about how to set up the library.

There are two configurations described in this section:

- In the first configuration, both drives in the ACSLS library are the same device type. See “Configuring an ACSLS library with a single drive device type”
- In the second configuration, the drives are different device types.

Drives with different device types (or different generations of drives) are supported in a single physical library if you define one library to Tivoli Storage Manager for each type of drive (or generation of drive). If you have two device types, such as 9840 and 9940 (or two generations of drives of the same device type), define two libraries. Then define drives and device classes for each library. In each device class definition, you can use the `FORMAT` parameter with a value of `DRIVE`, if you choose. See “Configuring an ACSLS library with multiple drive device types” on page 114.

Configuring an ACSLS library with a single drive device type

The parameter `ACSID` specifies the number that the Automatic Cartridge System Administrator (ACSSA) assigned to the library. Issue the `QUERY ACS` command to your ACSLS system to determine the number for your library ID.

1. Define an ACSLS library named `ACSLIB`:

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

2. Define the drives in the library:

```
define drive acslib drive01 acsdrvid=1,2,3,4
define drive acslib drive02 acsdrvid=1,2,3,5
```

The `ACSDRVID` parameter specifies the ID of the drive that is being accessed. The drive ID is a set of numbers that indicate 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.

See “Defining drives” on page 127.

3. Define a path from the server to each drive:

```
define path server1 drive01 srctype=server desttype=drive
  library=acslib device=/dev/tmscsi/mt0
```

```
define path server1 drive02 srctype=server desttype=drive
  library=acslib device=/dev/tmscsi/mt1
```

4. Classify drives according to type by defining Tivoli Storage Manager device classes. For example, to classify the two drives in the ACSLIB library, issue the following command to define a device class named ACS_CLASS:

```
define devclass acs_class library=acslib devtype=ecartridge format=drive
```

This example uses FORMAT=DRIVE as the recording format because both drives associated with the device class use the same recording format; for example, both are 9940 drives. If instead one drive is a 9840 and one is a 9940, you must use specific recording formats when defining the device classes. See “Configuring an ACSLS library with multiple drive device types” on page 114. See “Defining tape and optical device classes” on page 201.

5. To check what you have defined, issue the following commands:

```
query library
query drive
query path
query devclass
```

See the following topics:

- “Obtaining information about device classes” on page 216
- “Requesting information about libraries” on page 155
- “Requesting information about paths” on page 164
- “Requesting information about drives” on page 156

6. Create the storage pool to use the devices in the device class you just defined. For example, define a storage pool named ACS_POOL associated with the device class ACS_CLASS:

```
define stgpool acs_pool acs_class maxscratch=20
```

Key choices:

- a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. The default setting for primary storage pools is collocation by group. The default for copy storage pools and active-data pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

For more information, see “Defining storage pools” on page 227.

Configuring an ACSLS library with multiple drive device types

The following example shows how to set up an ACSLS library with a mix of two 9840 drives and two 9940 drives.

1. Define *two* ACSLS libraries that use the same ACSID. For example to define 9840LIB and 9940LIB, enter the following commands:

```
define library 9840lib libtype=acsls acsid=1
define library 9940lib libtype=acsls acsid=1
```

The ACSID parameter specifies the number that the Automatic Cartridge System System Administrator (ACSSA) assigned to the libraries. Issue the QUERY ACS command to your ACSLS system to determine the number for your library ID.

2. Define the drives, ensuring that they are associated with the appropriate libraries.

Note: Tivoli Storage Manager does not prevent you from associating a drive with the wrong library.

- Define the 9840 drives to 9840LIB.

```
define drive 9840lib 9840_drive1 acsdrvid=1,2,3,1
define drive 9840lib 9840_drive2 acsdrvid=1,2,3,2
```

- Define the 9940 drives to 9940LIB.

```
define drive 9940lib 9940_drive3 acsdrvid=1,2,3,3
define drive 9940lib 9940_drive4 acsdrvid=1,2,3,4
```

The ACSDRVID parameter specifies the ID of the drive that is being accessed. The drive ID is a set of numbers that indicate 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.

See "Defining drives" on page 127.

3. Define a path from the server to each drive. Ensure that you specify the correct library.

- For the 9840 drives:

```
define path server1 9840_drive1 srctype=server desttype=drive
library=9840lib device=/dev/tsm SCSI/mt0
```

```
define path server1 9840_drive2 srctype=server desttype=drive
library=9840lib device=/dev/tsm SCSI/mt1
```

- For the 9940 drives:

```
define path server1 9940_drive3 srctype=server desttype=drive
library=9940lib device=/dev/tsm SCSI/mt2
```

```
define path server1 9940_drive4 srctype=server desttype=drive
library=9940lib device=/dev/tsm SCSI/mt3
```

4. Classify the drives according to type by defining Tivoli Storage Manager device classes, which specify the recording formats of the drives. Because there are separate libraries, you can enter a specific recording format, for example 9840, or you can enter DRIVE. For example, to classify the drives in the two libraries, use the following commands to define one device class for each type of drive:

```
define devclass 9840_class library=9840lib devtype=ecartridge format=9840
```

```
define devclass 9940_class library=9940lib devtype=ecartridge format=9940
```

See "Defining tape and optical device classes" on page 201.

5. To check what you have defined, enter the following commands:

```
query library
query drive
query path
query devclass
```

See the following topics:

- “Requesting information about libraries” on page 155
 - “Obtaining information about device classes” on page 216
 - “Requesting information about paths” on page 164
 - “Requesting information about drives” on page 156
6. Create the storage pools to use the devices in the device classes that you just defined. For example, define storage pools named 9840_POOL associated with the device class 9840_CLASS and 9940_POOL associated with the device class 9940_CLASS:

```
define stgpool 9840_pool 9840_class maxscratch=20

define stgpool 9940_pool 9940_class maxscratch=20
```

Key choices:

- a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. The default setting for primary storage pools is collocation by group. The default for copy storage pools and active-data pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

For more information, see “Defining storage pools” on page 227.

Setting up an ACSLS library manager server

You can configure an ACSLS library so that it contains all drives of the same device type. In addition, you can configure an ACSLS library so that it contains drives of different device types or different generations of drives.

You can modify the procedure described for configuring a library for use by one server (“Configuration with multiple drive device types” on page 93) and use it for configuring a shared library. You must set up the server that is the library manager before you set up servers that are the library clients.

When upgrading multiple servers participating in library sharing to Version 5.3, upgrade all the servers at once, or upgrade the library manager servers and then the library client servers. A Version 5.3 library manager server is compatible with downlevel library clients. However, 5.3 library clients are not compatible with a library manager server that is at a level earlier than 5.3.

Note: An exception to this rule is when a fix or product enhancement requires concurrent code changes to the server, storage agent, and library client.

The following example procedure describes how to set up a server named GLENCOE as a library manager. Perform the following steps on the server GLENCOE:

1. Define a server as a library client server named WALLACE:

```
define server wallace HLA=x.x.x.x LLA=y
```

2. Define a shared ACSLS library named MACGREGOR:

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

This example used an ACSLS managed library configured as acs 1.

3. Define all the device classes that are associated with the shared library MACGREGOR:

```
define devclass tape library=macgregor devtype=9840
```

4. Define the drives in the library MACGREGOR:

```
define drive macgregor drivea acsdrvid=1,0,1,0
define drive macgregor driveb acsdrvid=1,0,1,1
```

This example uses the acsdrvid value, which 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.

5. Define a path from the server GLENCOE to each drive in the library MACGREGOR:

```
define path glencoe drivea srctype=server desttype=drive
  library=macgregor device=/dev/IBMtape4
define path glencoe driveb srctype=server desttype=drive
  library=macgregor device=/dev/IBMtape5
```

For more information about paths, see “Defining paths” on page 129.

Setting up an ACSLS library client server

You must set up the server that is the library manager server before you set up the servers that are the library clients.

Use the following sample procedure for each Tivoli Storage Manager server that will be a library client. The library client server is named WALLACE. With the exception of one step, perform the procedure from the library client server.

1. Define the server that is the library manager:

```
define server glencoe serverpassword=secret haddress=9.115.3.45 lladdress=1580
  crossdefine=yes
```

2. Define the shared library named MACGREGOR, and identify the library manager server's name as the primary library manager. Ensure that the library name is the same as the library name on the library manager:

```
define library macgregor libtype=shared primarylibmanager=glencoe
```

3. *Perform this step from the library manager.* Define a path from the library client server to each drive that the library client server will be allowed to access. The device name should reflect the way the *library client* system sees the device. There must be a path defined from the library manager to each drive in order for the library client to use the drive. The following is an example of how to define a path:

```
define path wallace drivea srctype=server desttype=drive
  library=macgregor device=/dev/IBMtape6
define path wallace driveb srctype=server desttype=drive
  library=macgregor device=/dev/IBMtape7
```

For more information about paths, see “Defining paths” on page 129.

4. *Return to the library client for the remaining steps.* Define all the device classes that are associated with the shared library:

```
define devclass tape library=macgregor devtype=1to
```

Set the parameters for the device class the same on the library client as on the library manager. Making the device class names the same on both servers is also a good practice, but is not required.

The device class parameters specified on the library manager server override those specified for the library client. This is true whether or not the device class names are the same on both servers. If the device class names are different, the library manager uses the parameters specified in a device class that matches the device type specified for the library client.

Note: If a library client requires a setting that is different from what is specified in the library manager's device class (for example, a different mount limit), do the following:

- a. Create an additional device class on the library manager server. Specify the parameter settings you want the library client to use.
 - b. Create a device class on the library client with the same name and device type as the new device class you created on the library server.
5. Define the storage pool, LOCHNESS, that will use the shared library:

```
define stgpool lochness tape maxscratch=50
```

6. Repeat this procedure to define additional servers as library clients.

Checking in and labeling ACSLS library volumes

Ensure that enough volumes are available to the server in the library. You must label volumes that do not already have a standard label. Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries.

Tip: If your library has drives of multiple device types, you defined *two* libraries to the Tivoli Storage Manager server in the procedure in “Configuring an ACSLS library with multiple drive device types” on page 114. The two Tivoli Storage Manager libraries represent the *one* physical library. The check-in process finds all available volumes that are not already checked in. You must check in media *separately* to each defined library. Ensure that you check in volumes to the correct Tivoli Storage Manager library.

1. Check in the library inventory. The following shows examples for libraries with a single drive device type and with multiple drive device types.

- Check in volumes that are already labeled:

```
checkin libvolume acslib search=yes status=scratch checklabel=no
```

- Label and check in volumes:

```
label libvolume acslib search=yes overwrite=no checkin=scratch
```

2. Depending on whether you use scratch volumes or private volumes, do one of the following:
 - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
 - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See “Defining storage pool volumes” on page 237.

For more information about checking in volumes, see “Checking new volumes into a library” on page 137

Removable file device configuration

Support for removable file devices allows portability of media among UNIX and Linux systems.

It also allows this media to be used to transfer data between systems that support the media. Removable file support allows the server to read data from a FILE device class that is copied to removable file media through software that is acquired from another vendor. The media is then usable as input media on a target Tivoli Storage Manager server that uses the REMOVABLEFILE device class for input.

Note: Software for writing CDs may not work consistently across platforms.

Removable file support includes support for rewritable CDs.

Use a MAXCAPACITY value that is less than one CD's usable space to allow for a one-to-one match between files from the FILE device class and copies that are on CD. Use the DEFINE DEVCLASS or UPDATE DEVCLASS commands to set the MAXCAPACITY parameter of the FILE device class to a value less than 650 MB.

Example of removable file support

You can take an export object and move it from one server to another by using a CD.

Server A

1. Define a device class with a device type of FILE.

```
define devclass file devtype=file directory=/home/user1
```
2. Export the node. This command results in a file name */home/user1/CDR03* that contains the export data for node USER1

```
export node user1 filedata=all devclass=file vol=cdr03
```

You can use software for writing CDs to create a CD with volume label CDR03 that contains a single file that is also named CDR03.

Server B

1. Follow the manufacturer's instructions to attach the device to your server.
2. Issue this command on your system to mount the CD.

```
mount -t iso9660 /dev/cdrom /cdrom
```

-t iso9660

Specifies that the media has a CD file system

/dev/cdrom

Specifies the physical description of the first CD on the system

/cdrom

Specifies the mount point of the first CD drive

Notes:

- a. CD drives lock while the file system is mounted. This prevents use of the eject button on the drive.

Note: CD drives lock while the file system is mounted. This prevents use of the eject button on the drive.

3. Ensure that the media is labeled. The software that you use for making a CD also labels the CD. Before you define the drive, you must put formatted, labeled media in the drive. For label requirements, see “Labeling requirements for removable file device types.” When you define the drive, the server verifies that a valid file system is present.
4. Define a manual library named CDR0M:

```
define library cdrom libtype=manual
```
5. Define the drive in the library:

```
define drive cdrom cddrive
```
6. Define a path from the server to the drive at mount point */cdrom*:

```
define path serverb cddrive srctype=server desttype=drive  
library=cdrom device=/cdrom
```
7. Define a device class with a device type of REMOVABLEFILE. The device type must be REMOVABLEFILE.

```
define devclass cdrom devtype=removablefile library=cdrom
```
8. Issue the following Tivoli Storage Manager command to import the node data on the CD volume CDR03.

```
import node user1 filedata=all devclass=cdrom vol=cdr03
```

For mounting a removable file device other than CD, use the following command:

```
mount -t ext2 /dev/hdc1 /tsmrfile
```

Labeling requirements for removable file device types

Tivoli Storage Manager does not provide utilities to format or label media for the REMOVABLEFILE device type.

You must use another application to copy the FILE device class data from the CD as a file that has the same name as the volume label. The software used to copy the FILE device class data must also label the removable media.

CDs must be formatted using the ISO9660 filesystem. Other removable file volumes such as Jaz or Zip must be formatted using the EXT2 filesystem, and the -L option on the mke2fs command must be used to specify the volume label.

The label on the media must meet the following restrictions:

- No more than 11 characters
- No embedded blanks or periods
- File name must be the same as the volume label

Configuration for libraries controlled by media manager programs

You can use an external media manager program with Tivoli Storage Manager to manage your removable media.

While the server tracks and manages client data, the media manager, operating entirely outside of the I/O data stream, labels, catalogs, and tracks physical volumes. The media manager also controls library drives, slots, and doors.

Tivoli Storage Manager provides a programming interface that lets you use a variety of media managers. See “Setting up Tivoli Storage Manager to work with an external media manager” for setup procedures.

To use a media manager with Tivoli Storage Manager, define a library that has a library type of EXTERNAL. The library definition will point to the media manager rather than a physical device.

Setting up Tivoli Storage Manager to work with an external media manager

To use the external media management interface with a media manager, complete the following procedure. This example is for a device containing two StorageTek drives.

1. Set up the media manager to interface with Tivoli Storage Manager. For more information, see Appendix A, “External media management interface description,” on page 859 and the documentation for the media manager.

2. Define an external library named MEDIAMGR:

```
define library mediamgr libtype=external
```

Note: You do not define the drives to the server in an externally managed library.

3. Define a path from the server to the library:

```
define path server1 mediamgr srctype=server desttype=library  
externalmanager=/usr/sbin/mediamanager
```

In the EXTERNALMANAGER parameter, specify the media manager's installed path. For more information about paths, see “Defining paths” on page 129.

4. Define device class, EXTCLASS, for the library with a device type that matches the drives. For this example the device type is ECARTRIDGE.

```
define devclass extclass library=mediamgr devtype=ecartridge  
mountretention=5 mountlimit=2
```

The MOUNTLIMIT parameter specifies the number of drives in the library device.

Note:

- a. For environments in which devices are shared across storage applications, the MOUNTRETENTION setting should be carefully considered. This parameter determines how long an idle volume remains in a drive. Because some media managers will 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.
 - b. It is recommended that you explicitly specify the mount limit instead of using MOUNTLIMIT=DRIVES.
5. Define a storage pool, EXTPOOL, for the device class. For example:

```
define stgpool extpool extclass maxscratch=500
```

Key choices:

- a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

Externally-controlled IBM Tivoli Storage Manager media

There are some issues to consider when controlling Tivoli Storage Manager media externally.

Labeling Media

The media manager handles the labeling of media. However, you must ensure that an adequate supply of blank media is available.

Checking Media into the Library

Externally managed media are not tracked in the Tivoli Storage Manager volume inventory. Therefore, you do *not* perform library check-in procedures by using Tivoli Storage Manager commands.

Using DRM

If you are using DRM, you can use the MOVE DRMEDIA command to request the removal of media from the library. For more information, see Chapter 26, “Using disaster recovery manager,” on page 801.

Migrating Media to External Media Manager Control

We strongly recommend that you not migrate media from Tivoli Storage Manager control to control by an external media manager. Instead, use external media management on a new Tivoli Storage Manager configuration or when defining externally managed devices to the server.

Deleting Tivoli Storage Manager Storage Pools from Externally Managed Libraries

Before deleting storage pools associated with externally managed libraries, first delete any volumes associated with the Tivoli Storage Manager library. For more information, see “Deleting storage pool volumes that contain data” on page 362.

Refer to the documentation for the media manager for detailed setup and management information.

Media manager database errors

Error conditions can cause the Tivoli Storage Manager volume information to be different from the media manager's volume database.

The most likely symptom of this problem is that the volumes in the media manager's database are not known to the server, and thus not available for use. Verify the Tivoli Storage Manager volume list and any disaster recovery media. If volumes not identified to the server are found, use the media manager interface to deallocate and delete the volumes.

Configuring manually mounted devices

In order to configure mounted devices manually, you must first set up the device on the server system.

The following tasks must be completed:

1. Set the appropriate SCSI ID for the device.
2. Physically attach the device to the server hardware.
3. Install and configure the appropriate device driver for the device.
4. Determine the device name that is needed to define the device to Tivoli Storage Manager.

See “Attaching a manual drive” on page 79 and “Device driver selection” on page 80 for details.

Defining devices as part of a manual library

In order to manually mount tapes, you need to define drives as part of a *manual* library.

In the following example, two DLT drives are attached to the server system and defined as part of a manual library:

1. Define a manual library named MANUALDLT:

```
define library manualdlt libtype=manual
```

2. Define the drives in the library:

```
define drive manualdlt drive01
define drive manualdlt drive02
```

See “Defining drives” on page 127 and http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

3. Define a path from the server to each drive:

```
define path server1 drive01 srctype=server desttype=drive
  library=manualdlt device=/dev/tsm SCSI/mt1
define path server1 drive02 srctype=server desttype=drive
  library=manualdlt device=/dev/tsm SCSI/mt2
```

For more about device special file names, see:

“Device special file names” on page 87

For more information about paths, see “Defining paths” on page 129.

4. Classify the drives according to type by defining a device class named TAPEDLT_CLASS. Use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical.

```
define devclass tapedlt_class library=manualdlt devtype=dlt format=drive
```

A closer look: When you associate more than one drive to a single device class through a manual library, ensure that the recording formats and media types of the devices are compatible. If you have a 4mm tape drive and a DLT tape drive, you must define separate manual libraries and device classes for each drive.

See “Defining tape and optical device classes” on page 201.

5. Verify your definitions by issuing the following commands:

```
query library
query drive
query path
query devclass
```

See “Requesting information about libraries” on page 155, “Requesting information about drives” on page 156, “Obtaining information about device classes” on page 216, and “Requesting information about paths” on page 164.

6. Define a storage pool named TAPEDLT_POOL associated with the device class named TAPEDLT_CLASS:

```
define stgpool tapedlt_pool tapedlt_class maxscratch=20
```

Key choices:

- a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can use any scratch volumes available without further action on your part. If you do not allow scratch volumes (MAXSCRATCH=0), you must perform the extra step of explicitly defining each volume to be used in the storage pool.
- b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping client files together using collocation” on page 310 and “How collocation affects reclamation” on page 329.

See “Defining storage pools” on page 227.

Labeling volumes

Use the following procedure to ensure that volumes are available to the server. Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup. Label and set aside extra scratch volumes for any potential recovery operations you might have later.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries.

Do the following:

1. Label volumes. For example, enter the following command to use one of the drives to label a volume with the ID of vol001:

```
label libvolume manualdlt vol001
```

Note: Tivoli Storage Manager only accepts tapes labeled with IBM standard labels. IBM standard labels are similar to ANSI Standard X3.27 labels except

that the IBM standard labels are written in EBCDIC. For a list of IBM media sales contacts who can provide compatible tapes, go to the IBM Web site. If you are using non-IBM storage devices and media, consult your tape-cartridge distributor.

2. Depending on whether you use scratch volumes or private volumes, do one of the following:
 - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
 - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled. For information on defining volumes, see “Defining storage pool volumes” on page 237.

Configuring IBM Tivoli Storage Manager for LAN-free data movement

You can configure the Tivoli Storage Manager client and server so that the client, through a storage agent, can move its data directly to storage on a SAN. This function, called LAN-free data movement, is provided by IBM Tivoli Storage Manager for Storage Area Networks.

As part of the configuration, a storage agent is installed on the client system. Tivoli Storage Manager supports SCSI, 349X, and ACSLS tape libraries as well as FILE libraries for LAN-free data movement. The configuration procedure you follow will depend on the type of environment you implement; however in all cases you must do the following:

1. Verify the network connection.
2. Establish communications among client, storage agent, and Tivoli Storage Manager.
3. Configure devices for the storage agent to access.
4. If you are using shared FILE storage, install and configure IBM TotalStorage SAN File System, Tivoli SANergy, or IBM General Parallel File System.
5. Start the storage agent and verify the LAN-free configuration.

To help you tune the use of your LAN and SAN resources, you can control the path that data transfers take for clients with the capability of LAN-free data movement. For each client you can select whether data read and write operations use:

- The LAN path only
- The LAN-free path only
- Either path

See the REGISTER NODE and UPDATE NODE commands in the *Administrator's Reference* for more about these options.

For more information on configuring Tivoli Storage Manager for LAN-free data movement see the *Storage Agent User's Guide*.

Validating your LAN-free configuration

Once you have configured your Tivoli Storage Manager client for LAN-free data movement, you can verify your configuration and server definitions by using the VALIDATE LANFREE command.

The VALIDATE LANFREE command allows you to determine which destinations for a given node using a specific storage agent are capable of LAN-free data movement. You can also issue this command to determine if there is a problem with an existing LAN-free configuration. You can evaluate the policy, storage pool, and path definitions for a given node using a given storage agent to ensure that an operation is working properly.

To determine if there is a problem with the client node FRED using the storage agent FRED_STA, issue the following:

```
validate lanfree fred fred_sta
```

The output will allow you to see which management class destinations for a given operation type are not LAN-free capable, and provide a brief explanation about why. It will also report the total number of LAN-free destinations.

See the VALIDATE LANFREE command in the *Administrator's Reference* for more information.

Configuring IBM Tivoli Storage Manager for NDMP operations

Tivoli Storage Manager can use NDMP (network data management protocol) to communicate with NAS (network attached storage) file servers and provide backup and restore services for NAS backups to locally attached library devices. This feature supports SCSI, 349X, and ACSLS (automated cartridge system library software) tape libraries.

To configure Tivoli Storage Manager for NDMP operations, you must perform the following steps:

1. Define the libraries and their associated paths.

Important: An NDMP device class can only use an Tivoli Storage Manager library in which all of the drives can read and write all of the media in the library.

2. Define a device class for NDMP operations.
3. Define the storage pool for backups performed by using NDMP operations.
4. *Optional:* Select or define a storage pool for storing tables of contents for the backups.
5. Configure Tivoli Storage Manager policy for NDMP operations.
6. Register the NAS nodes with the server.
7. Define a data mover for the NAS file server.
8. Define the drives and their associated paths.

For more information on configuring Tivoli Storage Manager for NDMP operations, see Chapter 8, "Using NDMP for operations with NAS file servers," on page 167 and "NDMP backup operations" on page 57.

Defining devices and paths

The following topics describe how to define libraries, drives, and paths to Tivoli Storage Manager.

See “Managing libraries” on page 154, “Managing drives” on page 156, and “Managing paths” on page 164 for information about displaying library, drive, and path information, and updating and deleting libraries and drives.

Defining libraries

Before you can use a drive, you must first define the library to which the drive belongs. This is true for both manually mounted drives and drives in automated libraries.

For example, you have several stand-alone tape drives. You can define a library named `MANUALMOUNT` for these drives by using the following command:

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

For all libraries other than manual libraries, you define the library and then define a path from the server to the library. For example, if you have an IBM 3583 device, you can define a library named `ROBOTMOUNT` using the following command:

```
define library robotmount libtype=scsi
```

Next, you use the `DEFINE PATH` command. In the path, you must specify the `DEVICE` parameter. The `DEVICE` parameter is required and specifies the device driver's name for the drive, which is the device special file name. The library's robotic mechanism is known by the device special file name.

For more about device special file names, see “Device special file names” on page 87.

```
define path server1 robotmount srctype=server desttype=library  
device=/dev/tmscsi/lb0
```

For more information about paths, see “Defining paths” on page 129.

If you have an IBM 3494 Tape Library Dataserver, you can define a library named `AUTOMOUNT` using the following command:

```
define library automount libtype=349x
```

Next, assuming that you have defined the symbolic name `lib3494` for the library in the configuration file for the library device driver, you define a path for the library:

```
define path server1 automount srctype=server desttype=library device=lib3494
```

Defining SCSI libraries on a SAN

For a library type of SCSI on a SAN, the server can track the library's serial number. With the serial number, the server can confirm the identity of the device when you define the path or when the server uses the device.

If you choose, you can specify the serial number when you define the library to the server. For convenience, the default is to allow the server to obtain the serial number from the library itself at the time that the path is defined.

If you specify the serial number, the server confirms that the serial number is correct when you define the path to the library. When you define the path, you can

set AUTODETECT=YES to allow the server to correct the serial number if the number that it detects does not match what you entered when you defined the library.

Depending on the capabilities of the library, the server may not be able to automatically detect the serial number. Not all devices are able to return a serial number when asked for it by an application such as the server. In this case, the server will not record a serial number for the device, and will not be able to confirm the identity of the device when you define the path or when the server uses the device. See “Impacts of device changes on the SAN” on page 131.

Defining drives

To inform the server about a drive that can be used to access storage volumes, issue the DEFINE DRIVE command, followed by the DEFINE PATH command.

When issuing the DEFINE DRIVE command, you must provide some or all of the following information:

Library name

The name of the library in which the drive resides.

Drive name

The name assigned to the drive.

Serial number

The serial number of the drive. The serial number parameter applies only to drives in SCSI libraries. With the serial number, the server can confirm the identity of the device when you define the path or when the server uses the device.

You can specify the serial number if you choose. The default is to allow the server to obtain the serial number from the drive itself at the time that the path is defined. If you specify the serial number, the server confirms that the serial number is correct when you define the path to the drive. When you define the path, you can set AUTODETECT=YES to allow the server to correct the serial number if the number that it detects does not match what you entered when you defined the drive.

Depending on the capabilities of the drive, the server may not be able to automatically detect the serial number. In this case, the server will not record a serial number for the device, and will not be able to confirm the identity of the device when you define the path or when the server uses the device. See “Impacts of device changes on the SAN” on page 131.

Element address

The element address of the drive. The ELEMENT parameter applies only to drives in SCSI libraries. The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can allow the server to obtain the element address from the drive itself at the time that the path is defined, or you can specify the element number when you define the drive.

Depending on the capabilities of the library, the server may not be able to automatically detect the element address. In this case you must supply the element address when you define the drive, if the library has more than one drive. If you need the element address, check the Tivoli Storage Manager support Web site at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

For example, to define a drive that belongs to the manual library named MANLIB, enter this command:

```
define drive manlib tapedrv3
```

Next, you define the path from the server to the drive, using the device name used to access the drive:

```
define path server1 tapedrv3 srctype=server desttype=drive library=manlib  
device=/dev/tmsmcsi/mt3
```

For more information about paths, see “Defining paths” on page 129.

Defining data movers

Data movers are network-attached devices that, through a request from Tivoli Storage Manager, transfer client data for backup or restore purposes. Data movers are defined as unique objects to Tivoli Storage Manager. Types of data mover devices include NAS file servers.

When issuing the DEFINE DATAMOVER command, you must provide some or all of the following information:

Data mover name

The name of the defined data mover.

Type The type of data mover (NAS).

High level address

The high level address is either the numerical IP address or the domain name of a NAS file server.

Low level address

The low level address specifies the TCP port number used to access a NAS file server.

User ID

The user ID specifies the ID for a user when initiating a Network Data Management Protocol (NDMP) session with a NAS file server.

Password

The password specifies the password associated with a user ID when initiating an NDMP session with a NAS file server. Check with your NAS file server vendor for user ID and password conventions.

Online

The online parameter specifies whether the data mover is online.

Data format

The data format parameter specifies the data format used according to the type of data mover device used.

An example of defining a NAS data mover named NAS1 would be :

```
define datamover nas1 type=nas hladdress=netapp2.tucson.ibm.com  
lladdress=10000 userid=root password=admin dataformat=netappdump
```

Defining paths

Before a device can be used, a path must be defined between the device and the server or the device and the data mover responsible for outboard data movement.

The DEFINE PATH command must be used to define the following path relationships:

- Between a server and a drive or a library.
- Between a storage agent and a drive.
- Between a data mover and a drive or a library.

When issuing the DEFINE PATH command, you must provide some or all of the following information:

Source name

The name of the server, storage agent, or data mover that is the source for the path.

Destination name

The assigned name of the device that is the destination for the path.

Source type

The type of source for the path. (A storage agent is considered a type of server for this purpose.)

Destination type

The type of device that is the destination for the path.

Library name

The name of the library that a drive is defined to if the drive is the destination of the path.

Device

The special file name of the device. This parameter is used when defining a path between a server, a storage agent, or a NAS data mover and a library or drive.

Automatic detection of serial number and element address

For devices on a SAN, you can specify whether the server should correct the serial number or element address of a drive or library, if it was incorrectly specified on the definition of the drive or library. The server uses the device name to locate the device and compares the serial number (and the element address for a drive) that it detects with that specified in the definition of the device. The default is to not allow the correction.

See the following examples:

If you have a SCSI type library named AUTODLTLIB that has a device name of /dev/tmsmcsi/lb3, define the path to the server named ASTRO1 by doing the following:

```
define path astrol autodlplib srctype=server desttype=library
device=/dev/tmsmcsi/lb3
```

If you have a drive, DRIVE01, that resides in library AUTODLTLIB, and has a device name of /dev/tmsmcsi/mt4, define it to server ASTRO1 by doing the following:

```
define path astrol drive01 srctype=server desttype=drive library=autodlplib
device=/dev/tmsmcsi/mt4
```

Shared FILE volumes

The Tivoli Storage Manager server and any storage agents associated with it are separate systems and each has a different view of the storage it is trying to access.

Because of this, problems can arise if path definitions to that storage are not accurate. The server has no way of validating the directory structure and storage paths that storage agents see, so diagnosing failures of this nature is very difficult.

The mechanisms to map the server view of storage to the storage agent view of storage are DEFINE DEVCLASS-FILE for the server and DEFINE PATH for the storage agent or agents. The DIRECTORY parameter in the DEFINE DEVCLASS-FILE command specifies the directory location or locations where the server places files that represent storage volumes for the FILE device class. For storage agents, the DIRECTORY parameter in the DEFINE PATH command serves the same purpose. The device class definition sets up a directory structure for the server and the DEFINE PATH definition tells the storage agent what that directory structure is. If path information is incorrect, the server and storage agent or agents will not be able to store files.

In order for the server and storage agent to be consistent on the storage they are sharing, the directories defined in the device class definition for the server and on the DEFINE PATH command for the storage agent should reference the same storage, in the same order and with an equal number of directories. This should be the same for each FILE drive that the storage agent is using. Shared file libraries are used to set up the storage pool that will be shared between the server and storage agents. FILE drives within that library are used so that the DEFINE PATH command can convey the information to the storage agent.

SANergy and shared FILE volumes

In a typical file sharing configuration, a file server is connected to a server and storage agent or agents on the LAN.

When the server or storage agent needs to write data to storage, it contacts the file server over the LAN. The file server then contacts the hard disk or storage drive over the SAN and reserves the space needed for the storage agent or server to store volumes. Once the space is reserved, the server or storage agent writes the data to be stored to the File Server over the LAN and then the File Server writes the data again to storage over the SAN. Only one operation can take place at a time, so if the server is in contact with the File Server during an operation, a storage agent attempting to contact the File Server will have to wait its turn.

To maximize throughput and improve performance, SANergy allows a server or storage agent to write directly to a hard disk or storage space over the SAN. A SANergy client is installed on each server and storage agent on the LAN and this client communicates with the SANergy server. When a storage agent or server is ready to write data to storage, the SANergy client on that system contacts the SANergy server. The SANergy server reserves the space on the hard disk and then informs the SANergy client that the server or storage agent can begin writing data directly to storage. This system bypasses the LAN and maintains the integrity of file systems, while maximizing throughput.

Impacts of device changes on the SAN

The SAN environment can shift dramatically due to device or cabling changes. Device IDs assigned by the SAN may be altered due to bus resets or other environmental changes. This "dynamically" changing nature of the SAN can cause the "static" definitions defined and known to the server (or storage agent) to fail or become unpredictable.

For instance, the server may know a device as *id=1* based on the original path specification to the server and original configuration of the LAN. However, some event in the SAN (new device added, cabling change) causes the device to be assigned *id=2*. When the server tries to access the device with *id=1*, it will either get a failure or the wrong target device. The server assists in recovering from changes to devices on the SAN by using serial numbers to confirm the identity of devices it contacts.

When you define a device (drive or library) you have the option of specifying the serial number for that device. If you do not specify the serial number when you define the device, the server obtains the serial number when you define the path for the device. In either case, the server then has the serial number in its database. From then on, the server uses the serial number to confirm the identity of a device for operations.

When the server uses drives and libraries on a SAN, the server attempts to verify that the device it is using is the correct device. The server contacts the device by using the device name in the path that you defined for it. The server then requests the serial number from the device, and compares that serial number with the serial number stored in the server database for that device.

If the serial number does not match, the server begins the process of discovery on the SAN to attempt to find the device with the matching serial number. If the server finds the device with the matching serial number, it corrects the definition of the path in the server's database by updating the device name in that path. The server issues a message with information about the change made to the device. Then the server proceeds to use the device.

You can monitor the activity log for messages if you want to know when device changes on the SAN have affected Tivoli Storage Manager. The following are the number ranges for messages related to serial numbers:

- ANR8952 through ANR8958
- ANR8961 through ANR8968
- ANR8974 through ANR8975

Restriction: Some devices do not have the capability of reporting their serial numbers to applications such as the Tivoli Storage Manager server. If the server cannot obtain the serial number from a device, it cannot assist you with changes to that device's location on the SAN.

Chapter 7. Managing removable media operations

Routine removable media operations include preparing media for use, controlling how and when media are reused, and ensuring that sufficient media are available. You also need to know how to respond to operator requests and how to manage libraries, drives, disks, paths, and datamovers.

Tasks
"Preparing removable media"
"Labeling removable media volumes" on page 134
"Checking new volumes into a library" on page 137
"Controlling access to volumes" on page 143
"Reusing tapes in storage pools" on page 144
"Reusing volumes used for database backups and export operations" on page 146
"Managing volumes in automated libraries" on page 147
"Managing server requests for media" on page 151
"Managing libraries" on page 154
"Managing drives" on page 156
"Managing paths" on page 164
"Managing data movers" on page 165

The examples in topics show how to perform tasks using the Tivoli Storage Manager command-line interface. For information about the commands, see the *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Preparing removable media

Removable media must be labeled before it can be used. When Tivoli Storage Manager accesses a removable media volume, it checks the volume name in the label header to ensure that the correct volume is accessed.

To prepare a volume for use, perform the following steps:

1. Label the volume. Any tape or optical volumes must be labeled before the server can use them.
2. For automated libraries, check the volume into the library.

Tip: When you use the LABEL LIBVOLUME command with drives in an automated library, you can label and check in the volumes with one command.

3. If the storage pool cannot contain scratch volumes (MAXSCRATCH=0), identify the volume to Tivoli Storage Manager by name so that it can be accessed later. For details, see "Defining storage pool volumes" on page 237.

If the storage pool can contain scratch volumes (MAXSCRATCH is set to a non-zero value), skip this step.

See "Checking new volumes into a library" on page 137.

See “Labeling removable media volumes.”

Labeling removable media volumes

Tape or optical volumes must be labeled before the server can use them. To check in and label volumes in one operation, issue the LABEL LIBVOLUME command. You can also use the AUTOLABEL parameter on the DEFINE and UPDATE LIBRARY commands.

When you use the LABEL LIBVOLUME command, which is issued from the server console or an administrative client, you provide parameters that specify the following information:

- The name of the library where the storage volume is located
- The name of the storage volume
- Whether to overwrite a label on the volume
- Whether to search an automated library for volumes for labeling
- Whether to read media labels:
 - To prompt for volume names in SCSI libraries
 - To read the barcode label for each cartridge in SCSI, 349X, and automated cartridge system library software (ACSL) libraries
- Whether to check in the volume:
 - To add the volume to the scratch pool
 - To designate the volume as private
- The type of device (applies to 349X libraries only)

To use the LABEL LIBVOLUME command, there must be at least one drive that is not in use by another Tivoli Storage Manager process. This includes volumes that are mounted but idle. 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 parameter.

Attention:

- By overwriting a volume label, you destroy all of the data that resides on the volume. Use caution when overwriting volume labels to avoid destroying important data.
- The labels on VolSafe volumes can be overwritten only once. Therefore, you should use the LABEL LIBVOLUME command only once for VolSafe volumes. You can guard against overwriting the label by using the OVERWRITE=NO option on the LABEL LIBVOLUME command.

By overwriting a volume label, you destroy all of the data that resides on the volume. Use caution when overwriting volume labels to avoid destroying important data.

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 in an empty slot. For manual libraries, you are prompted to load the volume directly into a drive.

For information about the AUTOLABEL parameter, see “Labeling new volumes using AUTOLABEL” on page 136.

Labeling volumes in a manual drive

To label volumes in a manual drive, issue the LABEL LIBVOLUME command.

Suppose that you want to label a few new volumes by using a manual tape drive that is defined as the following:

```
/dev/tsm SCSI/mt5
```

The drive is attached at SCSI address 5. Issue the following command:

```
label libvolume tsmlibname volname
```

Restriction: The LABEL LIBVOLUME command selects the next free drive. If you have more than one free drive, it cannot be:

```
/dev/tsm SCSI/mt5
```

Labeling volumes in a SCSI or ACSLS library

You can label volumes one-at-a-time or let the Tivoli Storage Manager search the library for volumes.

Labeling volumes one-at-a-time:

When you label volumes one-at-a-time, you can specify a volume name.

Perform the following steps to label volumes one-at-a-time:

1. Insert volumes into the library when prompted to do so. The library mounts each inserted volume into a drive.
2. For a SCSI library, enter a volume name when you are prompted (LABELSOURCE=PROMPT). A label is written to the volume using the name that you entered.
3. If the library does not have an entry/exit port, you are prompted to remove the tape from a specified slot number (not a drive). If the library has an entry/exit port, the command by default returns each labeled volume to the entry/exit port of the library.

Labeling new volumes in a SCSI library:

You can use the LABEL LIBVOLUME command to overwrite existing volume labels.

Suppose you want to label a few new volumes in a SCSI library that does not have entry and exit ports. You want to manually insert each new volume into the library, and you want the volumes to be placed in storage slots inside the library after their labels are written. You know that none of the new volumes contains valid data, so it is acceptable to overwrite existing volume labels. You only want to use one of the library's four drives for these operations.

Issue the following command:

```
label libvolume tsmlibname volname overwrite=yes checkin=scratch
```

Labeling new volumes using AUTOLABEL:

To automatically label tape volumes, you can use the AUTOLABEL parameter on the DEFINE and UPDATE LIBRARY commands. Using this parameter eliminates the need to pre-label a set of tapes.

It is also 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.

Searching the library:

Tivoli Storage Manager can search all of the storage slots in a library for volumes and can attempt to label each volume that it finds.

Use the LABEL LIBVOLUME command the SEARCH=YES parameter to search a library.

After a volume is labeled, the volume is returned to its original location in the library. Specify SEARCH=BULK if you want the server to search through all the slots of bulk entry/exit ports for labeled volumes that it can check in automatically. The server searches through all slots even if it encounters an unavailable slot.

When you specify LABELSOURCE=PROMPT, the volume is moved from its location in the library or in the entry/exit ports to the drive. The server prompts you to issue the REPLY command containing the label string, and that label is written to the tape.

If the library has a barcode reader, the LABEL LIBVOLUME command can use the reader to obtain volume names, instead of prompting you for volume names. Use the SEARCH=YES and LABELSOURCE=BARCODE parameters. If you specify the LABELSOURCE=BARCODE parameter, the volume bar code is read, and the tape is moved from its location in the library or in the entry/exit ports to a drive where the barcode 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.

Suppose that you want to label all volumes in a SCSI library. Enter the following command:

```
label libvolume tsm libname search=yes labelsource=barcode
```

Tivoli Storage Manager will select the next available drive.

Note: The LABELSOURCE=BARCODE parameter is valid only for SCSI libraries.

Labeling volumes in a 349x library

For a 349X library, the server attempts to label only volumes in certain categories. All other volumes are ignored by the labeling process. This precaution prevents the inadvertent destruction of that data on volumes being actively used by other systems connected to the library device.

The LABEL LIBVOLUME command labels volumes in the INSERT category, the private category (PRIVATECATEGORY), the scratch category (SCRATCHCATEGORY) and the WORM scratch category (WORMSCRATCHCATEGORY), but does not label the volumes already checked into the library.

Suppose that you want to label all of the volumes that are in the INSERT category in an IBM TotalStorage 3494 Tape Library. Enter the following command:

```
label libvolume tsmlibname search=yes devtype=3590
```

Checking new volumes into a library

You can inform the server that a new volume is available in an automated library. You can also check in and label volumes in one operation. The required privilege class is system, or unrestricted storage.

To inform the server that a new volume is available in an automated library, check in the volume with the CHECKIN LIBVOLUME command or LABEL LIBVOLUME command with the CHECKIN option specified. When a volume is checked in, the server adds the volume to its library volume inventory. You can use the LABEL LIBVOLUME command to check in and label volumes in one operation.

Note:

1. Do not mix volumes with barcode labels and volumes without barcode labels in a library device because barcode scanning can take a long time for unlabeled volumes.
2. You must use the CHECKLABEL=YES (not NO or BARCODE) option on the CHECKIN LIBVOLUME command when checking VolSafe volumes into a library. This is true for both automated cartridge system library software (ACSL) and SCSI libraries.

When you check in a volume, you must supply the name of the library and the status of the volume (private or scratch). To check in one or just a few volumes, you can specify the name of the volume with the command, and issue the command for each volume. To check in a larger number of volumes, you can use the search capability of the CHECKIN command or you can use the VOLRANGE parameter of the CHECKIN command.

When using the CHECKIN LIBVOLUME command, be prepared to supply some or all of the following information:

Library name

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

Volume name

Specifies the volume name of the storage volume being checked in.

Status Specifies the status that is assigned to the storage volume being checked in. If you check in a volume that has already been defined in a storage pool or in the volume history file, you must specify a volume status of *private* (STATUS=PRIVATE). This status ensures that the volume is not overwritten

when a scratch mount is requested. The server does not check in a volume with scratch status when that volume already belongs to a storage pool or is a database, export, or dump volume.

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

Check label

Specifies whether Tivoli Storage Manager should read sequential media labels of volumes during CHECKIN command processing, or use a barcode reader.

Swap Specifies whether Tivoli Storage Manager will initiate a swap operation when an empty slot is not available during CHECKIN command processing.

Mount wait

Specifies the maximum length of time, in minutes, to wait for a storage volume to be mounted.

Search

Specifies whether Tivoli Storage Manager searches the library for volumes that have not been checked in.

Device type

This parameter allows you to specify the device type for the volume being checked in. Use this parameter only for 349X libraries in which the drives do not have defined paths.

For more information, see:

- “Checking volumes into a SCSI library one-at-a-time”
- “Checking in volumes in library slots” on page 139
- “Checking in volumes in library entry or exit ports” on page 140
- “Checking media labels” on page 140
- “Allowing swapping of volumes when the library is full” on page 140

Checking volumes into a SCSI library one-at-a-time

You can check in only a single volume that is not currently in the library by issuing the CHECKIN LIBVOLUME command and specifying SEARCH=NO. Tivoli Storage Manager requests that the mount operator load the volume in the entry/exit port of the library.

If the library does not have an entry/exit port, Tivoli Storage Manager requests that the mount operator load the volume into a slot within the library. The request specifies the location with an *element address*. For any library or medium changer that does not have an entry/exit port, you need to know the element addresses for the cartridge slots and drives. If there is no worksheet listed for your device in http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager, see the documentation that came with your library.

Note: Element addresses are sometimes numbered starting with a number other than one. Check the worksheet to be sure.

For example, to check in volume VOL001 manually, enter the following command:

```
checkin libvolume tapelib vol001 search=no status=scratch
```

If the library has an entry/exit port, you are prompted to insert a cartridge into the entry/exit port. If the library does not have an entry/exit port, you are prompted

to insert a cartridge into one of the slots in the library. Element addresses identify these slots. For example, Tivoli Storage Manager finds that the first empty slot is at element address 5. The message is:

```
ANR8306I 001: Insert 8MM volume VOL001 R/W in slot with element
address 5 of library TAPELIB within 60 minutes; issue 'REPLY' along
with the request ID when ready.
```

Check the worksheet for the device if you do not know the location of element address 5 in the library. To find the worksheet, see http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager. When you have inserted the volume as requested, respond to the message from a Tivoli Storage Manager administrative client. Use the request number (the number at the beginning of the mount request):

```
reply 1
```

Note: A REPLY command is not required if you specify a wait time of zero using the optional WAITTIME parameter on the CHECKIN LIBVOLUME command. The default wait time is 60 minutes.

Checking volumes into a 349x library one-at-a-time

You can use the SEARCH=NO parameter on the CHECKIN LIBVOLUME command to search for volumes that have already been inserted into the library from the convenience or bulk I/O station.

The following command syntax allows you to search for volumes that have already been inserted into a 349X library from the convenience or bulk I/O station while specifying SEARCH=NO:

```
checkin libvolume 3494lib vol001 search=no status=scratch
```

If the volume has already been inserted, the server finds and processes it. If not, you can insert the volume into the I/O station during the processing of the command.

Checking in volumes in library slots

You can use the SEARCH=YES parameter on the CHECKIN LIBVOLUME command to search the library slots for new volumes that have not already been added to the library volume inventory.

Use this mode when you have a large number of volumes to check in, and you want to avoid issuing an explicit CHECKIN LIBVOLUME command for each volume. For example, for a SCSI library you can simply open the library access door, place all of the new volumes in unused slots, close the door, and issue the CHECKIN LIBVOLUME command with SEARCH=YES.

If you are using a 349X library, the server searches only for new volumes in the following categories:

- INSERT
- Tivoli Storage Manager's private category (PRIVATECATEGORY, specified when you define the library)
- Tivoli Storage Manager's scratch category (SCRATCHCATEGORY, specified when you define the library)
- Tivoli Storage Manager's WORM scratch category (WORMSCRATCHCATEGORY, specified when you define the library)

This restriction prevents the server from using volumes owned by another application that is accessing the library simultaneously.

Checking in volumes in library entry or exit ports

You can use the SEARCH=BULK parameter on the CHECKIN LIBVOLUME command to search through all of the slots of bulk entry and exit ports for labeled volumes that the Tivoli Storage Manager server can check in automatically.

The server searches through all slots even if it encounters an unavailable slot. For SCSI libraries, the server scans all of the entry/exit ports in the library for volumes. If a volume is found that contains a valid volume label, it is checked in automatically. The CHECKLABEL option NO is invalid with this SEARCH option. When you use the CHECKLABEL=YES parameter, the volume is moved from the entry/exit ports to the drive where the label is read. After reading the label, the tape is moved from the drive to a storage slot. When you use the CHECKLABEL=BARCODE parameter, the volume's bar code is read and the tape is moved from the entry/exit port to a storage slot. For barcode support to work correctly, the Tivoli Storage Manager or IBMtape device driver must be installed for libraries controlled by Tivoli Storage Manager.

Checking media labels

You can reduce the amount of time for checking in volumes by using a barcode reader, if your library has one.

When you check in a volume, you can specify whether Tivoli Storage Manager should read the labels of the media during checkin processing. When label-checking is on, Tivoli Storage Manager mounts each volume to read the internal label and only checks in a volume if it is properly labeled. This can prevent future errors when volumes are actually used in storage pools, but also increases processing time at check in.

If a library has a barcode reader and the volumes have barcode labels, you can save time in the check in process. Tivoli Storage Manager uses the characters on the label as the name for the volume being checked in. If a volume has no barcode label, Tivoli Storage Manager mounts the volumes in a drive and attempts to read the recorded label. For example, to use the barcode reader to check in all volumes found in the TAPELIB library as scratch volumes, enter the following command:

```
checkin libvolume tapelib search=yes status=scratch checklabel=barcode
```

For information on how to label new volumes, see "Preparing removable media" on page 133.

Allowing swapping of volumes when the library is full

If no empty slots are available in the library when you are checking in volumes, the checkin fails unless you allow *swapping*. If you allow swapping and the library is full, Tivoli Storage Manager selects a volume to eject before checking in the volume you requested.

Use the CHECKIN LIBVOLUME command to allow swapping. When you specify YES for the SWAP parameter, Tivoli Storage Manager initiates a swap operation if an empty slot is not available to check in a volume. Tivoli Storage Manager ejects the volume that it selects for the swap operation from the library and replaces the ejected volume with the volume that is being checked in. For example:

```
checkin libvolume auto wpdv00 swap=yes
```

Tivoli Storage Manager selects the volume to eject by checking first for any available scratch volume, then for the least frequently mounted volume.

Write-once, read-many tape media

Write-once, read-many (WORM) media helps prevent accidental or deliberate deletion of critical data. However, Tivoli Storage Manager imposes certain restrictions and guidelines to follow when using WORM media.

Tivoli Storage Manager supports the following types of WORM media:

- StorageTek VolSafe
- Sony AIT50 and AIT100
- IBM 3592
- IBM LTO-3 and LTO-4; HP LTO-3 and LTO-4; and Quantum LTO-3
- Quantum SDLT 600, Quantum DLT V4, and Quantum DLT S4

Tips:

- External and manual libraries use separate logical libraries to segregate their media. Ensuring that the correct media are loaded is the responsibility of the operator and the library manager software.
- A storage pool can consist of either WORM or RW media, but not both.
- Do not use WORM tapes for database backup or export operations. Doing so wastes tape following a restore or import operation.

For information about defining device classes for WORM tape media, see “Defining device classes for StorageTek VolSafe devices” on page 215 and “Defining tape and optical device classes” on page 201.

For information about selecting device drivers for IBM and devices from other vendors, see:

“Device driver selection” on page 80.

WORM-capable drives

To use WORM media in a library, all the drives in the library must be WORM-capable. A mount will fail if a WORM cartridge is mounted in a read write (RW) drive.

However, a WORM-capable drive can be used as a RW drive if the WORM parameter in the device class is set to NO. Any type of library can have both WORM and RW media if *all* of the drives are WORM enabled. The only exception to this rule is NAS-attached libraries in which WORM tape media cannot be used.

Checkin of WORM media

The type of WORM media determines whether the media label needs to be read during checkin.

Library changers cannot identify the difference between standard read-write (RW) tape media and the following types of WORM tape media:

- VolSafe
- Sony AIT
- LTO
- SDLT
- DLT

To determine the type of WORM media that is being used, a volume must be loaded into a drive. Therefore, when checking in one of these types of WORM volumes, you must use the CHECKLABEL=YES option on the CHECKIN LIBVOLUME command.

If they provide support for WORM media, IBM 3592 library changers can detect whether a volume is WORM media without loading the volume into a drive. Specifying CHECKLABEL=YES is not required. Verify with your hardware vendors that your 3592 drives and libraries provide the required support.

Restrictions on WORM media

You cannot use pre-labeled WORM media with the LTO or ECARTRIDGE device class. You cannot use WORM media in IBM LTO-4 drives, HP LTO-4, or Sun StorageTek T10000B drives with Tivoli Storage Manager specified as the drive-encryption key manager.

Mount failures with WORM media

If WORM tape media are loaded into a drive for a read-write (RW) device-class mount, it will cause a mount failure. Similarly, if RW tape media are loaded into a drive for a WORM device-class mount, the mount will fail.

Relabeling WORM media

You cannot relabel a WORM cartridge if it contains data. This applies to Sony AIT WORM, LTO WORM, SDLT WORM, DLT WORM, and IBM 3592 cartridges. The label on a VolSafe volume should be overwritten only once and only if the volume does not contain usable, deleted, or expired data.

Issue the LABEL LIBVOLUME command only once for VolSafe volumes. You can guard against overwriting the label by using the OVERWRITE=NO option on the LABEL LIBVOLUME command.

Removing private WORM volumes from a library

If you perform some action on a WORM volume (for example, if you delete file spaces) and the server does not mark the volume as full, the volume is returned to scratch status. If a WORM volume is not marked as full and you delete it from a storage pool, the volume will remain private. To remove a private WORM volume from a library, you must issue the CHECKOUT LIBVOLUME command.

Creation of DLT WORM volumes

DLT WORM volumes can be converted from read-write (RW) volumes.

If you have SDLT-600, DLT-V4, or DLT-S4 drives and you want to enable them for WORM media, upgrade the drives using V30 or later firmware available from Quantum. You can also use DLTIce software to convert unformatted read-write (RW) volumes or blank volumes to WORM volumes.

In SCSI or automated-cartridge system-library software (ACSL) libraries, the Tivoli Storage Manager server creates scratch DLT WORM volumes automatically when the server cannot locate any scratch WORM volumes in a library's inventory. The server converts available unformatted or blank RW scratch volumes or empty RW private volumes to scratch WORM volumes. The server also rewrites labels on newly created WORM volumes using the label information on the existing RW volumes.

In manual libraries, you can use the server to format empty volumes to WORM.

Support for short and normal 3592 WORM tapes

Tivoli Storage Manager supports both short and normal 3592 WORM tapes. For best results, define them in separate storage pools

Querying a device class for the WORM-parameter setting

You can determine the setting of the WORM parameter for a device class by using the QUERY DEVCLASS command. The output contains a field, labeled WORM, and a value (YES or NO).

Managing the volume inventory

You can manage your volume inventory by controlling Tivoli Storage Manager access to volumes, by reusing tapes, and by reusing volumes used for database backups and export operations. You can also manage inventory by maintaining a supply of scratch volumes.

With Tivoli Storage Manager, you manage your volume inventory by performing the following tasks: Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries but that are used by the same server.

Controlling access to volumes

If you want to allow a volume to be read but not written to, you can change its access mode. You can also control access taking off-site volumes in a copy storage pool or an active-data pool.

Tivoli Storage Manager expects to be able to access all volumes it knows about. For example, Tivoli Storage Manager tries to fill up tape volumes. If a volume containing client data is only partially full, Tivoli Storage Manager will later request that volume be mounted to store additional data. If the volume cannot be mounted, an error occurs.

To make volumes that are not full available to be read but not written to, you can change the volume access mode. For example, use the UPDATE VOLUME command with ACCESS=READONLY. The server will not attempt to mount a volume that has an access mode of unavailable.

If you want to make volumes unavailable in order to send the data they contain off-site for safekeeping, a more controlled way to do this is to use a copy storage pool or an active-data pool. You can back up your primary storage pools to a copy storage pool and then send the copy storage pool volumes off-site. You can also copy active versions of client backup data to active-data pools, and then send the volumes off-site. You can track copy storage pool volumes and active-data pool volumes by changing their access mode to off-site, and updating the volume history to identify their location. For more information, see “Backing up storage pools” on page 760.

Reusing tapes in storage pools

Tape reuse is controlled by expiration and reclamation processing. You can run either of these processes automatically or manually.

To reuse tapes in storage pools:

Expire client files

Expiration processing deletes from the database information about any client files that are expired (no longer valid according to the policies you have set). For example, suppose four backup versions of a file exist in server storage, and only three versions are allowed in the backup policy (the management class) for the file. Expiration processing deletes information about the oldest of the four versions of the file. The space that the file occupied in the storage pool can then be reclaimed.

You can run expiration processing automatically or by command. See “Running expiration processing to delete expired files” on page 472.

Reclaim volumes

You can have Tivoli Storage Manager reclaim volumes that pass a *reclamation threshold*, a percentage of unused space on the volume. Tivoli Storage Manager moves data to consolidate valid, unexpired files onto fewer tapes. The reclamation threshold is set for each storage pool. See “Reclaiming space in sequential-access storage pools” on page 319.

For a storage pool associated with a library that has more than one drive, the reclaimed data is moved to other volumes in the same storage pool. For a storage pool associated with a library that has only one drive, the reclaimed data is moved to volumes in another storage pool that you must define, called a reclamation storage pool. See “Reclaiming volumes in a storage pool with one drive” on page 323.

Setting up a tape rotation

To help ensure an adequate supply of tapes, you can expire old files, reclaim volumes, and delete volumes that have reached end of life. You can also maintain a supply of scratch volumes.

Over time, media ages, and some of the backup data located on it may no longer be needed. You can set Tivoli Storage Manager policy to determine how many backup versions are retained and how long they are retained. Then, expiration processing allows the server to delete files you no longer want to keep. You can keep the useful data on the media and then reclaim and reuse the media themselves.

Deleting data - expiration processing

Expiration processing deletes data that is no longer valid either because it exceeds the retention specifications in policy or because users or administrators have deleted the active versions of the data.

For more information, see:

- “Basic policy planning” on page 436
- “Running expiration processing to delete expired files” on page 472
- “File expiration and expiration processing” on page 439

Reusing media - reclamation processing

Data on tapes may expire, move, or be deleted. Reclamation processing

consolidates any unexpired data by moving it from multiple volumes onto fewer volumes. The media can then be returned to the storage pool and reused.

You can set a reclamation threshold that allows Tivoli Storage Manager to reclaim volumes whose valid data drops below a threshold. The threshold is a percentage of unused space on the volume and is set for each storage pool. The amount of data on the volume and the reclamation threshold for the storage pool affects when the volume is reclaimed. See “Reclaiming space in sequential-access storage pools” on page 319.

Determining when media have reached end of life

You can use Tivoli Storage Manager to display statistics about volumes, including the number of write operations performed on the media and the number of write errors. For media initially defined as private volumes, Tivoli Storage Manager maintains this statistical data, even as the volume is reclaimed. You can compare the information with the number of write operations and write errors recommended by the manufacturer. For media initially defined as scratch volumes, Tivoli Storage Manager overwrites this statistical data each time the media are reclaimed.

Reclaim any valid data from volumes that have reached end of life. If the volumes are in automated libraries, check them out of the volume inventory. Delete private volumes from the database with the DELETE VOLUME command.

For more information, see “Reclaiming space in sequential-access storage pools” on page 319.

Ensuring media are available for the tape rotation

Over time, the demand for volumes may cause the storage pool to run out of space. You can set the maximum number of scratch volumes high enough to meet demand by doing one or both of the following:

- Increase the maximum number of scratch volumes by updating the storage pool definition. Label and check in new volumes to be used as scratch volumes if needed.
- Make volumes available for reuse by running expiration processing and reclamation, to consolidate data onto fewer volumes. For more information, see “Reusing tapes in storage pools” on page 144.

For automated libraries, see “Managing server requests for media” on page 151.

1. Ensure that the maximum number of scratch volumes for the device storage pool is at least equal to the number of storage slots in the library.
2. Check enough volumes into the device's volume inventory for the expected load.

Reusing volumes used for database backups and export operations

You cannot reuse volumes that were used for database backups and export operations until you delete the volume information from the volume history file.

When you back up the database or export server information, Tivoli Storage Manager records information about the volumes used for these operations in the *volume history* file. Tivoli Storage Manager will not allow you to reuse these volumes until you delete the volume information from the volume history file. To reuse volumes that were previously used for database backup or export, use the DELETE VOLHISTORY command.

Note: If your server uses the disaster recovery manager function, the volume information is automatically deleted during MOVE DRMEDIA command processing.

For additional information about DRM, see Chapter 26, “Using disaster recovery manager,” on page 801.

For information about the volume history file, see “Saving the volume history file” on page 768.

Maintaining a supply of scratch volumes

You must set the maximum number of scratch volumes high enough for the expected usage.

When you define a storage pool, you must specify the maximum number of scratch volumes that the storage pool can use. Tivoli Storage Manager automatically requests a scratch volume when needed. When the number of scratch volumes that Tivoli Storage Manager is using for the storage pool exceeds the maximum number of scratch volumes specified, the storage pool can run out of space.

When you exceed the maximum number of scratch volumes, you can do one or both of the following:

- Increase the maximum number of scratch volumes by updating the storage pool definition. Label new volumes to be used as scratch volumes if needed.
- Make volumes available for reuse by running expiration processing and reclamation, to consolidate data onto fewer volumes. See “Reusing tapes in storage pools” on page 144.

Remember: Because you might need additional volumes for future recovery operations, consider labeling and setting aside extra scratch volumes.

For information about automated libraries, see “Maintaining a supply of scratch volumes in an automated library” on page 150.

Managing volumes in automated libraries

Tivoli Storage Manager tracks the scratch and private volumes available in an automated library through a library volume inventory. You must ensure that the inventory is consistent with the volumes that are physically in the library.

Tivoli Storage Manager tracks the scratch and private volumes available in an automated library through a *library volume inventory*. Tivoli Storage Manager maintains an inventory for each automated library. The library volume inventory is separate from the inventory of volumes for each storage pool. To add a volume to a library's volume inventory, you *check in* a volume to that Tivoli Storage Manager library.

To ensure that Tivoli Storage Manager's library volume inventory remains accurate, you must *check out* volumes when you need to physically remove volumes from a SCSI, 349X, or automated cartridge system library software (ACSL) library. When you check out a volume that is being used by a storage pool, the volume remains in the storage pool. If Tivoli Storage Manager requires the volume to be mounted while it is checked out, a message to the mount operator's console is displayed with a request to check in the volume. If the check in is not successful, Tivoli Storage Manager marks the volume as unavailable.

While a volume is in the library volume inventory, you can change its status from scratch to private.

To check whether Tivoli Storage Manager's library volume inventory is consistent with the volumes that are physically in the library, you can audit the library. The inventory can become inaccurate if volumes are moved in and out of the library without informing the server using volume checkin or checkout.

Task	Required Privilege Class
Changing the status of a volume in an automated library	System or unrestricted storage
Removing volumes from a library	System or unrestricted storage
Returning volumes to a library	System or unrestricted storage

For details on the checkin procedure, see “Checking new volumes into a library” on page 137.

Changing the status of a volume

You can issue the UPDATE LIBVOLUME command to change the status of a volume in an automated library from scratch to private, or private to scratch.

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 use this command if you make a mistake when checking in volumes to the library and assign the volumes the wrong status.

Removing volumes from a library

You might want to remove a volume from an automated library if you have exported data to a volume in the library and want to take it to another system for an import operation. You might also want to remove volumes that are unlikely to be accessed to make room for new volumes.

To remove a volume from an automated library, use the CHECKOUT LIBVOLUME command. By default, the server mounts the volume being checked out and verifies the internal label. When the label is verified, the server removes the volume from the library volume inventory, and then moves it to the entry/exit port or convenience I/O station of the library. If the library does not have an entry/exit port, Tivoli Storage Manager requests that the mount operator remove the volume from a slot within the library.

If you check out a volume that is defined in a storage pool, the server may attempt to access it later to read or write data. If this happens, the server requests that the volume be checked in.

Perform the following steps to return volumes to a library:

1. Check in the volume for the library, with private status. Use the CHECKIN LIBVOLUME command with the parameter STATUS=PRIVATE.
2. If the volume was marked unavailable, update the volume's ACCESS value to read/write or read-only. Use the UPDATE VOLUME command with the ACCESS parameter.

Managing a full library

As Tivoli Storage Manager fills volumes in a storage pool, the number of volumes needed for the pool might exceed the physical capacity of the library. To make room for new volumes while keeping track of existing volumes, define a storage pool overflow location near the library and then move media to the overflow location as needed.

Perform the following steps to manage a full library:

1. Define or update the storage pool associated with the automated library, including the overflow location parameter. For example, you have a storage pool named ARCHIVEPOOL associated with an automated library. Update the storage pool to add an overflow location of Room2948. Enter this command:

```
update stgpool archivepool ovflocation=Room2948
```
2. When the library becomes full, move the full volumes out of the library and to the overflow location that you defined for the storage pool. For example, to move all full volumes in the specified storage pool out of the library, enter this command:

```
move media * stgpool=archivepool
```

All full volumes are checked out of the library. Tivoli Storage Manager records the location of the volumes as Room2948. You can use the DAYS parameter to specify the number of days that must elapse before a volume is eligible for processing by the MOVE MEDIA command.

3. Check in new scratch volumes, if needed. If a volume has an entry in volume history, you cannot check it in as a scratch volume.
4. Reuse the empty scratch storage volumes in the overflow location. For example, enter this command:

```

query media * stg=* whereovflocation=Room2948 wherestatus=empty
move media * stg=* wherestate=mountablenotinlib wherestatus=empty
cmd="checkin libvol autolib &vol status=scratch"
cmdfilename=/tsm/move/media/checkin.vols

```

For more information, see the *Administrator's Reference*.

- As requested through Tivoli Storage Manager mount messages, check in volumes that Tivoli Storage Manager needs for operations. The mount messages include the overflow location of the volumes.

To find the overflow location of a storage pool, you can use the QUERY MEDIA command. This command can also be used to generate commands. For example, you can issue a QUERY MEDIA command to get a list of all volumes in the overflow location, and at the same time generate the commands to check in all those volumes to the library. For example, enter this command:

```

query media format=cmd stgpool=archivepool whereovflocation=Room2948
cmd="checkin libvol autolib &vol status=private"
cmdfilename="/tsm/move/media/checkin.vols"

```

Use the DAYS parameter to specify the number of days that must elapse before the volumes are eligible for processing by the QUERY MEDIA command.

The file that contains the generated commands can be run using the Tivoli Storage Manager MACRO command. For this example, the file may look like this:

```

checkin libvol autolib TAPE13 status=private
checkin libvol autolib TAPE19 status=private

```

Auditing a library's volume inventory

You can audit an automated library to ensure that the library volume inventory is consistent with the volumes that physically reside in the library. You may want to do this if the library volume inventory is disturbed due to manual movement of volumes in the library or to database problems.

Task	Required Privilege Class
Audit the volume inventory of a library	System or unrestricted storage

Issue the AUDIT LIBRARY command to restore the inventory to a consistent state. Missing volumes are deleted, and the locations of the moved volumes are updated. However, new volumes are not added during an audit.

Unless your SCSI library has a barcode reader, the server mounts each volume during the audit to verify the internal labels on volumes. For 349X libraries, the server uses the information from the Library Manager.

Issue the AUDIT LIBRARY command only when there are no volumes mounted in the library drives. If any volumes are mounted but in the IDLE state, you can issue the DISMOUNT VOLUME command to dismount them.

If a SCSI library has a barcode reader, you can save time by using the barcode reader to verify the identity of volumes. If a volume has a barcode label, the server uses the characters on the label as the name for the volume. The volume is not mounted to verify that the barcode name matches the internal volume name. If a volume has no barcode label, the server mounts the volume and attempts to read the recorded label. For example, to audit the TAPELIB library using its barcode reader, issue the following command:

Maintaining a supply of scratch volumes in an automated library

When you define a storage pool that is associated with an automated library, you can specify a maximum number of scratch volumes equal to the physical capacity of the library. However, the number of scratch volumes that Tivoli Storage Manager is using for the storage pool can exceed that number.

If the number of scratch volumes that Tivoli Storage Manager is using for the storage pool exceeds the number specified in the storage pool definition, perform the following steps:

1. Add scratch volumes to the library by checking in volumes. Label them if necessary. You might need to use an overflow location to move volumes out of the library to make room for these scratch volumes.
2. Increase the maximum number of scratch volumes by updating the storage pool definition. The increase should equal the number of scratch volumes that you checked in.

Keep in mind that you might need additional volumes for future recovery operations, so consider labeling and setting aside extra scratch volumes.

See “Maintaining a supply of scratch volumes” on page 146.

Operations with shared libraries

Shared libraries are logical libraries that are represented physically by SCSI, 349X, or ACSLS libraries. The physical library is controlled by the Tivoli Storage Manager server configured as a library manager. Tivoli Storage Manager servers using the SHARED library type are library clients to the library manager server.

The library client contacts the library manager, when the library manager starts and the storage device initializes, or after a library manager is defined to a library client. The library client confirms that the contacted server is the library manager for the named library device. The library client also compares drive definitions with the library manager for consistency. The library client contacts the library manager for each of the following operations:

Volume Mount

A library client sends a request to the library manager for access to a particular volume in the shared library device. For a scratch volume, the library client does not specify a volume name. If the library manager cannot access the requested volume, or if scratch volumes are not available, the library manager denies the mount request. If the mount is successful, the library manager returns the name of the drive where the volume is mounted.

Volume Release (free to scratch)

When a library client no longer needs to access a volume, it notifies the library manager that the volume should be returned to scratch. The library manager's database is updated with the volume's new location. The volume is deleted from the volume inventory of the library client.

Table 14 on page 151 shows the interaction between library clients and the library manager in processing Tivoli Storage Manager operations.

Table 14. How SAN-enabled servers process Tivoli Storage Manager Operations

Operation (Command)	Library Manager	Library Client
Query library volumes (QUERY LIBVOLUME)	Displays the volumes that are checked into the library. For private volumes, the owner server is also displayed.	Not applicable.
Check in and check out library volumes (CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME)	Performs the commands to the library device.	Not applicable. When a checkin operation must be performed because of a client restore, a request is sent to the library manager server.
Move media and move DRM media (MOVE MEDIA, MOVE DRMEDIA)	Only valid for volumes used by the library manager server.	Requests that the library manager server perform the operations. Generates a checkout process on the library manager server.
Audit library inventory (AUDIT LIBRARY)	Performs the inventory synchronization with the library device.	Performs the inventory synchronization with the library manager server.
Label a library volume (LABEL LIBVOLUME)	Performs the labeling and checkin of media.	Not applicable.
Dismount a volume (DISMOUNT VOLUME)	Sends the request to the library device.	Requests that the library manager server perform the operation.
Query a volume (QUERY VOLUME)	Checks whether the volume is owned by the requesting library client server and checks whether the volume is in the library device.	Requests that the library manager server perform the operation.

Managing server requests for media

Tivoli Storage Manager displays requests and status messages to all administrative clients that are started in console mode. These request messages often have a time limit. If the request is not fulfilled within the time limit, the operation times out and fails.

For manual libraries, Tivoli Storage Manager detects when there is a cartridge loaded in a drive, and no operator reply is necessary. For automated libraries, the CHECKIN LIBVOLUME and LABEL LIBVOLUME commands involve inserting cartridges into slots and, depending on the value of the WAITTIME parameter, issuing a reply message. (If the value of the parameter is zero, no reply is required.) The CHECKOUT LIBVOLUME command involves inserting cartridges into slots and, in all cases, issuing a reply message.

Using the administrative client for mount messages

The server sends mount request status messages to the server console and to all administrative clients in mount mode or console mode parameter.

For example, to start an administrative client in mount mode, enter the following command:

```
> dsmadm -mountmode
```

Mount operations for manual libraries

Volumes are mounted as a result of mount requests from Tivoli Storage Manager. For manual libraries, you can monitor the mount requests on the server console or through an administrative client in mount mode or console mode.

Someone you designate as the operator must respond to the mount requests by putting in tape volumes as requested.

Messages for automated libraries

You can see mount messages and messages about problems with the library on administrative clients in mount mode or console mode.

For automated libraries, mount messages are sent to the library and not to an operator. Messages about problems with the library are sent to the mount message queue. You cannot use the Tivoli Storage Manager REPLY command to respond to these messages.

Requesting information about pending operator requests

You can get information about pending operator requests either by issuing the QUERY REQUEST command or by checking the mount message queue on an administrative client started in mount mode.

Task	Required Privilege Class
Request information about operator requests or mounted volumes	Any administrator

When you issue the QUERY REQUEST command, Tivoli Storage Manager displays requested actions and the amount of time remaining before the requests time out. For example, you enter the command as follows:

```
query request
```

The following shows an example of a response to the command:

```
ANR8352I Requests outstanding:  
ANR8326I 001: Mount 8MM volume DSM001 R/W in drive TAPE01 (/dev/tsm SCSI/mt1)  
of MANUAL8MM within 60 minutes.
```

Replying to operator requests

When the server requires that an explicit reply be provided when a mount request is completed, you can reply with the REPLY command.

Task	Required Privilege Class
Reply to operator requests	Operator

The first parameter for the REPLY command is the request identification number that tells the server which of the pending operator requests has been completed. This three-digit number is always displayed as part of the request message. It can also be obtained by issuing a QUERY REQUEST command. If the request requires the operator to provide a device to be used for the mount, the second parameter for this command is a device name.

For example, enter the following command to respond to request 001 for tape drive TAPE01:

```
reply 1
```

Canceling an operator request

If a mount request for a manual library cannot be satisfied, you can issue the CANCEL REQUEST command. This command forces the server to cancel the request and causes the operation that needed the requested volume to fail.

Task	Required Privilege Class
Cancel operator requests	Operator

The CANCEL REQUEST command must include the request identification number. This number is included in the request message. You can also obtain it by issuing a QUERY REQUEST command. See “Requesting information about pending operator requests” on page 152.

You can specify the PERMANENT parameter if you want to mark the requested volume as UNAVAILABLE. This process is useful if, for example, the volume has been moved to a remote site or is otherwise inaccessible. By specifying PERMANENT, you ensure that the server does not try to mount the requested volume again.

For most of the requests associated with automated (SCSI) libraries, an operator must perform a hardware or system action to cancel the requested mount. For such requests, the CANCEL REQUEST command is not accepted by the server.

Responding to requests for volume check-in

If the server cannot find a particular volume to be mounted in an automated library, the server requests that the operator check in the volume. Your response depends on whether the volume is available or unavailable.

For example, a client requests that an archived file be retrieved. The file was archived in a storage pool in an automated library. The server looks for the volume containing the file in the automated library, but cannot find the volume. The server then requests that the volume be checked in.

If the volume that the server requests is available, put the volume in the library and check in the volume using the normal procedures (“Checking new volumes into a library” on page 137).

If the volume requested is unavailable (lost or destroyed), update the access mode of the volume to UNAVAILABLE by using the UPDATE VOLUME command. Then cancel the server's request for checkin by using the CANCEL REQUEST command. (Do *not* cancel the client process that caused the request.) To get the ID of the request to cancel, use the QUERY REQUEST command.

If you do not respond to the server's checkin request within the mount-wait period of the device class for the storage pool, the server marks the volume as unavailable.

Determining which volumes are mounted

For a report of all volumes currently mounted for use by the server, issue the QUERY MOUNT command. The report shows which volumes are mounted, which drives have accessed them, and if the volumes are currently being used.

Task	Required Privilege Class
Request information about which volumes are mounted	Operator

Dismounting idle volumes

After a volume becomes idle, the server keeps it mounted for a time specified by the mount retention parameter for the device class. Use of mount retention can reduce the access time if volumes are used repeatedly.

Task	Required Privilege Class
Request a volume dismount	Operator

To dismount an idle volume, issue the DISMOUNT VOLUME command. This command causes the server to dismount the named volume from the drive in which it is currently mounted.

For information about setting mount retention times, see “Controlling the amount of time that a volume remains mounted” on page 203.

Managing libraries

Using Tivoli Storage Manager commands, you can query and delete libraries. You can also update libraries.

Requesting information about libraries

You can request information about one or more libraries by issuing the QUERY LIBRARY command.

Task	Required Privilege Class
Request information about libraries	Any administrator

You can request either a standard or a detailed report. For example, to display information about all libraries, issue the following command:

```
query library
```

The following shows an example of the output from this command.

Library Name	Library Type	Private Category	Scratch Category	WORM Scratch Category	External Manager
MANLIB	MANUAL				
EXB	SCSI				
3494LIB	349X	300	301	302	

Updating libraries

You can update an existing library by issuing the UPDATE LIBRARY command. To update the device names of a library, issue the UPDATE PATH command. You cannot update a MANUAL library.

Task	Required Privilege Class
Update libraries	System or unrestricted storage

Automated libraries

If your system or device is re-configured and the device name changes, you might need to update the device name.

The examples below show you how you can use the UPDATE LIBRARY and UPDATE PATH commands for the following library types:

- SCSI
- 349X
- ACSLS
- External

Examples:

- **SCSI library**

Update the path from SERVER1 to a SCSI library named SCsilIB:

```
update path server1 scsilib srctype=server desttype=library  
device=/dev/tmscsi/lb1
```

Update the definition of a SCSI library named SCsilIB defined to a library client so that a new library manager is specified:

```
update library scsilib primarylibmanager=server2
```

- **349X library**

Update the path from SERVER1 to an IBM 3494 library named 3494LIB with new device names.

```
update path server1 3494lib srctype=server desttype=library
device=library2,library3,library4
```

Update the definition of an IBM 3494 library named 3494LIB defined to a library client so that a new library manager is specified:

```
update library 3494lib primarylibmanager=server2
```

- **ACSLs library**

Update an automated cartridge system library software (ACSLs) library named ACSLSLIB with a new ID number.

```
update library acslslib ascid=1
```

- **External library**

Update an external library named EXTLIB with a new media manager path name.

```
update path server1 extlib srctype=server desttype=library
externalmanager=/v/server/mediamanager.exe
```

Update an EXTERNAL library named EXTLIB in a LAN-free configuration so that the server uses the value set for mount retention in the device class associated with the library:

```
update library extlib obeymountretention=yes
```

Deleting libraries

You can delete libraries by issuing the DELETE LIBRARY command.

Task	Required Privilege Class
Delete libraries	System or unrestricted storage

Before you delete a library with the DELETE LIBRARY command, you must delete all of the drives that have been defined as part of the library and delete the path to the library.

For example, suppose that you want to delete a library named 8MMLIB1. After deleting all of the drives defined as part of this library and the path to the library, issue the following command to delete the library itself:

```
delete library 8mmlib1
```

See “Deleting drives” on page 163.

Managing drives

You can query, update, clean, and delete drives by using Tivoli Storage Manager commands.

Requesting information about drives

You can request information about drives by using the QUERY DRIVE command.

Task	Required Privilege Class
Request information about drives	Any administrator

The UPDATE DRIVE command accepts wildcard characters for both a library name and a drive name. See the *Administrator's Reference* for information about this command and the use of wildcard characters.

For example, to query all drives associated with your server, enter the following command:

```
query drive
```

The following shows an example of the output from this command.

Library Name	Drive Name	Device Type	On Line
MANLIB	8MM.0	8MM	Yes
AUTOLIB	8MM.2	8MM	Yes

Updating drives

You can change the attributes of a drive definition by issuing the UPDATE DRIVE command.

Task	Required Privilege Class
Update drives	System or unrestricted storage

The following are attributes of a drive definition that you can change:

- The element address, if the drive resides in a SCSI library
- The ID of a drive in an ACSLS library
- The cleaning frequency
- Change whether the drive is online or offline

For example, to change the element address of a drive named DRIVE3 to 119, issue the following command:

```
update drive auto drive3 element=119
```

If you are reconfiguring your system, you can change the device name of a drive by issuing the UPDATE PATH command. For example, to change the device name of a drive named DRIVE3, issue the following command:

```
update path server1 drive3 srctype=server desttype=drive library=scsilib  
device=/dev/IBMtape0
```

Remember: You cannot change the element number or the device name if a drive is in use. See “Taking drives offline.” If a drive has a volume mounted, but the volume is idle, it can be explicitly dismounted. See “Dismounting idle volumes” on page 154.

Taking drives offline

You can take a drive offline while it is in use. For example, you might take a drive offline for another activity, such as maintenance.

If you take a drive offline while it is in use, the mounted volume completes its current process. If this volume was part of a series of volumes in a transaction, the drive is no longer available to complete mounting the series. If no other drives are available, the active process may fail. The offline state is retained even if the server is halted and brought up again. If a drive is marked offline when the server is brought up, a warning is issued noting that the drive must be manually brought online. If all the drives in a library are taken offline, processes requiring a library mount point will fail, rather than queue up for one.

The ONLINE parameter specifies the value of the drive's online state, even if the drive is in use. ONLINE=YES indicates that the drive is available for use. ONLINE=NO indicates that the drive is not available for use (offline). Do not specify other optional parameters along with the ONLINE parameter. If you do, the drive will not be updated, and the command will fail when the drive is in use. You can specify the ONLINE parameter when the drive is involved in an active process or session, but this is not recommended.

Drive encryption

Drive encryption protects tapes that contain critical or sensitive data (for example, tapes that contain sensitive financial information). Drive encryption is particularly beneficial for tapes that are moved from the Tivoli Storage Manager server environment to an off-site location.

Tivoli Storage Manager supports encryption for the following drives:

- IBM 3592 generation 2 and generation 3
- IBM LTO generation 4
- HP LTO-4 generation 4
- Sun StorageTek T10000B

Drives must be able to recognize the correct format. With Tivoli Storage Manager, you can use the following encryption methods:

Table 15. Encryption methods supported

	Application method	Library method	System method
3592 generation 3	Yes	Yes	Yes
3592 generation 2	Yes	Yes	Yes
IBM LTO generation 4	Yes	Yes, but only if your system hardware (for example, 3584) supports it	Yes
HP LTO generation 4	Yes	No	No
Sun StorageTek T10000B	Yes	No	No

To enable drive encryption with IBM LTO-4, you must have the IBM RMSS Ultrium device driver installed. You cannot use SCSI drives with IBM LTO-4 encryption. To enable encryption with HP LTO-4, you must have the Tivoli Storage Manager device driver installed.

Drive encryption is enabled by specifying the DRIVEENCRYPTION parameter on the DEFINE DEVCLASS and UPDATE DEVCLASS commands for the 3592, LTO, and ECARTRIDGE device types.

A library can contain a mixture of drives, some of which support encryption and some that do not. (For example, a library might contain two LTO-2 drives, two LTO-3 drives, and two LTO-4 drives.) You can also mix media in a library using, for example, a mixture of encrypted and non-encrypted device classes having different tape and drive technologies. However, all LTO-4 drives must support encryption if Tivoli Storage Manager is to use drive encryption. In addition, all drives within a logical library must use the same method of encryption. When

using Tivoli Storage Manager, do not create an environment in which some drives use the Application method and some drives use the Library or System methods of encryption.

When using encryption-capable drives with a supported encryption method, a different format is used to write encrypted data to tapes. When data is written to volumes using the different format and if the volumes are then returned to scratch, they contain labels that are only readable by encryption-enabled drives. To use these scratch volumes in a drive that is not enabled for encryption, either because the hardware is not capable of encryption or because the encryption method is set to NONE, you must relabel the volumes.

For more information about setting up your hardware environment to use drive encryption, refer to your hardware documentation.

For details about the DRIVEENCRYPTION parameter, see the following topics:

- “Encrypting data with 3592 generation 2 and generation 3 drives” on page 206
- “Encrypting data using LTO generation 4 tape drives” on page 213
- “Enabling ECARTRIDGE drive encryption” on page 216 and “Disabling ECARTRIDGE drive encryption” on page 216

Cleaning drives

The server can control cleaning tape drives in SCSI libraries and offers partial support for cleaning tape drives in manual libraries.

Task	Required Privilege Class
Clean drives	System or unrestricted storage

For automated library devices, you can automate cleaning by specifying the frequency of cleaning operations and checking a cleaner cartridge into the library's volume inventory. Tivoli Storage Manager mounts the cleaner cartridge as specified. For manual library devices, Tivoli Storage Manager issues a mount request for the cleaner cartridge.

Drive-cleaning considerations

Some SCSI libraries provide automatic drive cleaning. In such cases, choose either the library drive cleaning or the Tivoli Storage Manager drive cleaning, but not both.

Manufacturers that include library cleaning recommend its use to prevent premature wear on the read/write heads of the drives. Drives and libraries from different manufacturers differ in how they manage cleaner cartridges and how they report the presence of a cleaner cartridge in a drive. The device driver may not be able to open a drive that contains a cleaner cartridge. Sense codes and error codes that are issued by devices for drive cleaning vary. Library drive cleaning is usually transparent to all applications. Therefore, Tivoli Storage Manager may not always detect cleaner cartridges in drives and may not be able to determine when cleaning has begun.

Some devices require a small amount of idle time between mount requests to start drive cleaning. However, Tivoli Storage Manager tries to minimize the idle time for a drive. The result may be to prevent the library drive cleaning from functioning

effectively. If this happens, try using Tivoli Storage Manager to control drive cleaning. Set the frequency to match the cleaning recommendations from the manufacturer.

If you have Tivoli Storage Manager control drive cleaning, disable the library drive cleaning function to prevent problems. If the library drive cleaning function is enabled, some devices automatically move any cleaner cartridge found in the library to slots in the library that are dedicated for cleaner cartridges. An application does not know that these dedicated slots exist. You will not be able to check a cleaner cartridge into the Tivoli Storage Manager library inventory until you disable the library drive cleaning function.

Cleaning drives in an automated library

When you set up server-controlled drive cleaning in an automated library, you can specify how often you want the drives cleaned.

To set up server-controlled drive cleaning in an automated library:

1. Define or update the drives in a library, using the CLEANFREQUENCY parameter.

The CLEANFREQUENCY parameter sets how often you want the drive cleaned. Refer to the DEFINE DRIVE and UPDATE DRIVE commands. Consult the manuals that accompany the drives for recommendations on cleaning frequency.

Remember: The CLEANFREQUENCY parameter is not valid for externally managed libraries, for example, 3494 libraries or Sun StorageTek libraries managed under automated cartridge system library software (ACSL). For example, to have DRIVE1 cleaned after 100 GB is processed on the drive, issue the following command:

```
update drive autolib1 drive1 cleanfrequency=100
```

Consult the drive manufacturer's information for cleaning recommendations. If the information gives recommendations for cleaning frequency in terms of hours of use, convert to a gigabytes value by doing the following:

- a. Use the bytes-per-second rating for the drive to determine a gigabytes-per-hour value.
- b. Multiply the gigabytes-per-hour value by the recommended hours of use between cleanings.
- c. Use the result as the cleaning frequency value.

Restrictions:

- For IBM 3590 and 3592 drives, specify a value for the CLEANFREQUENCY parameter rather than specify ASNEEDED. Using the cleaning frequency recommended by the product documentation will not overclean the drives.
 - The CLEANFREQUENCY=ASNEEDED parameter value does not work for all tape drives. To determine whether a drive supports this function, see the following Web site: http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_Linux.html. At this Web site, click the drive to view detailed information. If ASNEEDED is not supported, you can use the *gigabytes* value for automatic cleaning.
2. Check a cleaner cartridge into the library's volume inventory with the CHECKIN LIBVOLUME command. For example:

```
checkin libvolume autolib1 cleanv status=cleaner cleanings=10 checklabel=no
```

After the cleaner cartridge is checked in, the server will mount the cleaner cartridge in a drive when the drive needs cleaning. The server will use that cleaner cartridge for the number of cleanings specified. See “Checking in cleaner cartridges” and “Operations with cleaner cartridges in a library” on page 162 for more information.

For details on the commands, see the *Administrator's Reference*.

Checking in cleaner cartridges:

To have the server control drive cleaning without operator intervention, you must check a cleaner cartridge into an automated library's volume inventory. As a best practice, check in cleaner cartridges one-at-a-time and do not use the search function when checking in a cleaner cartridge.

When checking in a cleaner cartridge to a library, ensure that it is correctly identified to the server as a cleaner cartridge. Also use caution when a cleaner cartridge is already checked in and you are checking in data cartridges. Ensure that cleaner cartridges are in their correct home slots, or errors and delays can result.

When checking in data cartridges with SEARCH=YES, ensure that a cleaner cartridge is not in a slot that will be detected by the search process. Errors and delays of 15 minutes or more can result from a cleaner cartridge being improperly moved or placed. For best results, check in the data cartridges first when you use the search function. Then check in the cleaner cartridge separately.

For example, if you need to check in both data cartridges and cleaner cartridges, put the data cartridges in the library and check them in first. You can use the search function of the CHECKIN LIBVOLUME command (or the LABEL LIBVOLUME command if you are labeling and checking in volumes). Then check in the cleaner cartridge to the library by using one of the following methods.

- Check in without using search:

```
checkin libvolume autolib1 cleanv status=cleaner cleanings=10  
checklabel=no
```

The server then requests that the cartridge be placed in the entry/exit port, or into a specific slot.

- Check in using search, but limit the search by using the VOLRANGE or VOLLIST parameter:

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

The process scans the library by using the barcode reader, looking for the CLEANV volume.

Manual drive cleaning in an automated library:

If your library has limited capacity and you do not want to use a slot in your library for a cleaner cartridge, you can still make use of the server's drive cleaning function.

Set the cleaning frequency for the drives in the library. When a drive needs cleaning based on the frequency setting, the server issues the message, ANR8914I. For example:

```
ANR8914I Drive DRIVE1 in library AUTOLIB1 needs to be cleaned.
```

You can use that message as a cue to manually insert a cleaner cartridge into the drive. However, the server cannot track whether the drive has been cleaned.

Operations with cleaner cartridges in a library:

To ensure that drives are cleaned as needed, you must monitor the cleaning messages for any problems.

When a drive needs to be cleaned, the server runs the cleaning operation after dismounting a data volume if a cleaner cartridge is checked in to the library. If the cleaning operation fails or is canceled, or if no cleaner cartridge is available, then the indication that the drive needs cleaning is lost. Monitor cleaning messages for these problems. If necessary, use the CLEAN DRIVE command to have the server try the cleaning again, or manually load a cleaner cartridge into the drive.

The server uses a cleaner cartridge for the number of cleanings that you specify when you check in the cleaner cartridge. If you check in two or more cleaner cartridges, the server uses only one of the cartridges until the designated number of cleanings for that cartridge is reached. Then the server begins to use the next cleaner cartridge. If you check in two or more cleaner cartridges and issue two or more CLEAN DRIVE commands concurrently, the server uses multiple cartridges at the same time and decrements the remaining cleanings on each cartridge.

Visually verify that cleaner cartridges are in the correct storage slots before issuing any of the following commands:

- AUDIT LIBRARY
- CHECKIN LIBVOLUME with SEARCH specified
- LABEL LIBVOLUME with SEARCH specified

To find the correct slot for a cleaner cartridge, use the QUERY LIBVOLUME command.

Drive cleaning in a manual library

The server can issue messages telling you that a drive in a manual library needs to be cleaned.

Cleaning a drive in a manual library is the same as setting up drive cleaning without checking in a cleaner cartridge for an automated library. The server issues the ANR8914I message when a drive needs cleaning. For example:

```
ANR8914I Drive DRIVE1 in library MANLIB1 needs to be cleaned.
```

Monitor the activity log or the server console for these messages and load a cleaner cartridge into the drive as needed. The server cannot track whether the drive has been cleaned.

Error checking for drive cleaning

Occasionally an administrator might move some cartridges around within a library and put a data cartridge where Tivoli Storage Manager shows that there is a cleaner cartridge. Tivoli Storage Manager uses the process in this section to recover from the error.

When a drive needs cleaning, the server loads what its database shows as a cleaner cartridge into the drive. The drive then moves to a READY state, and Tivoli Storage Manager detects that the cartridge is a data cartridge. The server then performs the following steps:

1. The server attempts to read the internal tape label of the data cartridge.
2. The server ejects the cartridge from the drive and moves it back to the home slot of the “cleaner” cartridge within the library. If the eject fails, the server marks the drive offline and issues a message that the cartridge is still in the drive.
3. The server checks out the “cleaner” cartridge to avoid selecting it for another drive cleaning request. The “cleaner” cartridge remains in the library but no longer appears in the Tivoli Storage Manager library inventory.
4. If the server was able to read the internal tape label, the server checks the volume name against the current library inventory, storage pool volumes, and the volume history file.
 - If there is not a match, an administrator probably checked in a data cartridge as a cleaner cartridge by mistake. Now that the volume is checked out, you do not need to do anything else.
 - If there is a match, the server issues messages that manual intervention and a library audit are required. Library audits can take considerable time, so an administrator should issue the command when sufficient time permits. See “Auditing a library’s volume inventory” on page 149.

Deleting drives

You can delete drive definitions by issuing the DELETE DRIVE command.

Task	Required Privilege Class
Delete drives	System or unrestricted storage

A drive cannot be deleted if it is currently in use. If a drive has a volume mounted, but the volume is currently idle, it can be dismounted as described in “Dismounting idle volumes” on page 154. A drive cannot be deleted until the defined path to the drive has been deleted. Also, a library cannot be deleted until all of the drives defined within it are deleted.

Managing paths

Using Tivoli Storage Manager commands, you can query, update, and delete paths.

Requesting information about paths

You can issue the QUERY PATH command to obtain information about paths.

You can request either a standard or a detailed report. This command accepts wildcard characters for both a source name and a destination name. See the *Administrator's Reference* for information about this command and the use of wildcard characters.

For example, to display information about all paths, issue the following command:

```
query path
```

The following shows an example of the output from this command.

Source Name	Source Type	Destination Name	Destination Type	Online
SERVER1	server	TSMLIB	Library	Yes
NETAPP1	Data mover	DRIVE1	Drive	Yes
NETAPP1	Data mover	NASLIB	Library	Yes
datamover2	Data mover	drive4	Drive	Yes

Updating paths

You can update an existing path by issuing the UPDATE PATH command.

The following examples show how you can use the UPDATE PATH commands for the certain path types:

- **Library paths**

Update the path to change the device name for a SCSI library named SCSILIB:

```
update path server1 scsilib srctype=server desttype=library  
device=/dev/tmscsi/lb1
```

- **Drive paths**

Update the path to change the device name for a drive named NASDRV1:

```
update path nas1 nasdrv1 srctype=datamover desttype=drive  
library=naslib device=/dev/tmscsi/mt1
```

Deleting paths

You can delete an existing path definition by issuing the DELETE PATH command.

Task	Required Privilege Class
Delete paths	System or unrestricted storage

A path cannot be deleted if the destination is currently in use. Before you can delete a path to a device, you must delete the device.

Delete a path from a NAS data mover NAS1 to the library NASLIB.

```
delete path nas1 naslib srctype=datamover desttype=library
```

Attention: If you delete the path to a device or make the path offline, you disable access to that device.

Managing data movers

Using Tivoli Storage Manager commands, you can query, update, and delete data movers.

Requesting information about data movers

You can obtain information about SCSI and NAS data movers by issuing the QUERY DATAMOVER command.

You can request either a standard or a detailed report. For example, to display a standard report about all data movers, issue the following command:

```
query datamover *
```

The following shows an example of the output from this command.

Data Mover Name	Type	Online
NASMOVER1	NAS	Yes
NASMOVER2	NAS	No

Updating data movers

You can update an existing data mover definition by issuing the UPDATE DATAMOVER command.

For example, to update the data mover for the node named NAS1 to change the IP address, issue the following command:

```
update datamover nas1 haddress=9.67.97.109
```

Deleting data movers

You can delete an existing data mover definition by issuing the DELETE DATAMOVER command.

Before you can delete a data mover definition, you must delete all paths defined for the data mover. To delete a data mover named NAS1, issue the following command:

```
delete datamover nas1
```

Tape alert messages

Tape alert messages are generated by tape and library devices to report hardware errors. These messages help to determine problems that are not related to the IBM Tivoli Storage Manager server.

A log page is created and can be retrieved at any given time or at a specific time such as when a drive is dismounted.

There are three severity levels of tape alert messages:

- Informational (for example, you may have tried to load a cartridge type that is not supported)
- Warning (for example, a hardware failure is predicted)
- Critical (for example, there is a problem with the tape and your data is at risk)

Tape alert messages are turned off by default. You may set tape alert messages to ON or OFF by issuing the SET TAPEALERTMSG command. You may query tape alert messages by issuing the QUERY TAPEALERTMSG command.

Chapter 8. Using NDMP for operations with NAS file servers

You can plan, configure, and manage a backup environment that protects your network-attached storage (NAS) file server by using NDMP (network data management protocol). Tivoli Storage Manager Extended Edition includes support for the use of NDMP to back up and recover NAS file servers.

Tasks:
"Configuring Tivoli Storage Manager for NDMP operations" on page 173
"Determining the location of NAS backup" on page 175
"Setting up tape libraries for NDMP operations" on page 179
"Configuring Tivoli Storage Manager policy for NDMP operations" on page 174
"Registering NAS nodes with the Tivoli Storage Manager server" on page 185
"Defining a data mover for the NAS file server" on page 186
"Defining a path to a library" on page 187
"Defining a path to a library" on page 187
"Defining tape drives and paths for NDMP operations" on page 186
"Labeling and checking tapes into the library" on page 188
"Scheduling NDMP operations" on page 188
"Defining virtual file spaces" on page 188
"Tape-to-tape copy to back up data" on page 189
"Tape-to-tape copy to move data" on page 189
"Backing up and restoring NAS file servers using NDMP" on page 189
"Performing NDMP filer to Tivoli Storage Manager server backups" on page 191
"Managing table of contents" on page 172
"NDMP operations management" on page 170
"Managing NAS file server nodes" on page 170
"Managing data movers used in NDMP operations" on page 171
"Storage pool management for NDMP operations" on page 172

NDMP requirements

You must meet certain requirements when using NDMP (network data management protocol) for operations with network-attached storage (NAS) file servers.

Tivoli Storage Manager Extended Edition

Licensed program product that includes support for the use of NDMP.

NAS File Server

A NAS file server. The operating system on the file server must be supported by Tivoli Storage Manager. Visit http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager for a list of NAS file servers that are certified through the "Ready for IBM Tivoli software."

Note: Vendors on the “Ready for IBM Tivoli software” list follow guidelines to implement NDMP as specified by Tivoli Storage Manager. If a file server is on the list, it has undergone tests to ensure it is compatible with Tivoli Storage Manager.

The combination of file server model and operating system must be supported by the NAS file server. For more specifics, consult the product information for the NAS file server.

Tape Libraries

This requirement is only necessary for a backup to a locally attached NAS device. The Tivoli Storage Manager server supports two types of libraries for operations using NDMP. The libraries supported are SCSI and ACSLS (automated cartridge system library software). 349X tape libraries can also be used with certain NAS file servers.

- **SCSI library**

A SCSI library that is supported by the Tivoli Storage Manager server. Visit http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager. This type of library can be attached directly either to the Tivoli Storage Manager server or to the NAS file server. When the library is attached directly to the Tivoli Storage Manager server, the Tivoli Storage Manager server controls the library operations by passing the SCSI commands directly to the library. When the library is attached directly to the NAS file server, the Tivoli Storage Manager server controls the library by passing SCSI commands to the library through the NAS file server.

- **ACSLs library**

An ACSLS library can only be directly connected to the Tivoli Storage Manager server. The Tivoli Storage Manager server controls the library by passing the library request through TCP/IP to the library control server.

Note: The Tivoli Storage Manager server does not include External Library support for the ACSLS library when the library is used for NDMP operations.

- **349X library**

A 349X library can only be directly connected to the Tivoli Storage Manager server. The Tivoli Storage Manager server controls the library by passing the library request through TCP/IP to the library manager.

Library Sharing: The Tivoli Storage Manager server that performs NDMP operations can be a library manager for either an ACSLS, SCSI, or 349X library, but cannot be a library client. The Tivoli Storage Manager server can also be a library client, in a configuration where the NAS filer sends data to a Tivoli Storage Manager server using TCP/IP rather than to a tape library attached to the NAS filer. If the Tivoli Storage Manager server that performs NDMP operations is a library manager, that server must control the library directly and not by passing commands through the NAS file server.

Tape Drives

One or more tape drives in the tape library. A tape drive is only necessary for backup to a locally attached NAS device. The NAS file server must be able to access the drives. A NAS device is not supported in a mixed device library. The drives must be supported for tape backup operations by the

NAS file server and its operating system. For complete NDMP device support, refer to the NAS file server product documentation.

Drive Sharing: The tape drives can be shared by the Tivoli Storage Manager server and one or more NAS file servers. Also, when a SCSI or a 349X library is connected to the Tivoli Storage Manager server and not to the NAS file server, the drives can be shared by one or more NAS file servers and one or more Tivoli Storage Manager:

- Library clients
- Storage agents

Verify the compatibility of specific combinations of a NAS file server, tape devices, and SAN-attached devices with the hardware manufacturers.

Attention: Tivoli Storage Manager supports NDMP Version 4 for all NDMP operations. Tivoli Storage Manager will continue to support all NDMP backup and restore operations with a NAS device running NDMP version 3. The Tivoli Storage Manager server will negotiate the highest protocol level (either Version 3 or Version 4) with the NDMP server when establishing an NDMP connection. If you experience any issues with Version 4, you may want to try using Version 3.

Interfaces for NDMP operations

You can use several interfaces to perform NDMP (network data management protocol) operations. You can schedule an NDMP operation using the BACKUP NODE and RESTORE NODE commands, and scheduling the operation as an administrative schedule.

Client Interfaces:

- Backup-archive command-line client (on a Windows, 64 bit AIX, or 64 bit Sun Solaris system)
- Web client

Server Interfaces:

- Server console
- Command line on the administrative client

Tip: All examples in this chapter use server commands.

- Administration Center

The Tivoli Storage Manager Web client interface, available with the backup-archive client, displays the file systems of the network-attached storage (NAS) file server in a graphical view. The client function is not required, but you can use the client interfaces for NDMP operations. The client function is recommended for file-level restore operations. See “File-level backup and restore for NDMP operations” on page 192 for more information about file-level restore.

Tivoli Storage Manager prompts you for an administrator ID and password when you perform NDMP functions using either of the client interfaces. See the Backup-Archive Clients Installation and User's Guide for more information about installing and activating client interfaces.

Attention: In order to use the Tivoli Storage Manager backup-archive client or Web client to perform NAS operations, the file system names on the NAS device must have a forward slash ("/") as the first character. This restriction does not affect NAS operations initiated from the Tivoli Storage Manager server command line.

Data formats for NDMP backup operations

During filer-to-filer backup operations that use NDMP (network data management protocol) and are not stored in the Tivoli Storage Manager server storage hierarchy, the network-attached storage (NAS) file server controls the format of the data written to the tape library.

The NDMP format is not the same as the data format used for traditional Tivoli Storage Manager backups. When you define a NAS file server as a data mover and define a storage pool for NDMP operations, you specify the data format. For example, you would specify NETAPPDUMP if the NAS file server is a NetApp or an IBM System Storage N Series device. You would specify CELERRADUMP if the NAS file server is an EMC Celerra device. For all other devices, you would specify NDMPDUMP.

NDMP operations management

There are several administrator activities for NDMP operations.

These include:

- NAS nodes
- Data movers
- Tape libraries and drives
- Paths
- Device classes
- Storage pools
- Table of contents

Managing NAS file server nodes

You can update, query, rename, and remove NAS (network attached storage) nodes.

For example, assume you have created a new policy domain named NASDOMAIN for NAS nodes and you want to update a NAS node named NASNODE1 to include it in the new domain.

1. Query the node.
`query node nasnode1 type=nas`
2. Change the domain of the node by issuing the following command:
`update node nasnode1 domain=nasdomain`

Renaming a NAS node

To rename a NAS (network attached storage) node, you must also rename the corresponding NAS data mover; both must have the same name.

For example, to rename NASNODE1 to NAS1 you must perform the following steps:

1. Delete all paths between data mover NASNODE1 and libraries and between data mover NASNODE1 and drives.
2. Delete the data mover defined for the NAS node.
3. To rename NASNODE1 to NAS1, issue the following command:

```
rename node nasnode1 nas1
```
4. Define the data mover using the new node name. In this example, you must define a new data mover named NAS1 with the same parameters used to define NASNODE1.

Attention: When defining a new data mover for a node that you have renamed, ensure that the data mover name matches the new node name and that the new data mover parameters are duplicates of the original data mover parameters. Any mismatch between a node name and a data mover name or between new data mover parameters and original data mover parameters can prevent you from establishing a session with the NAS file server.

5. For SCSI or 349X libraries, define a path between the NAS data mover and a library only if the tape library is physically connected directly to the NAS file server.
6. Define paths between the NAS data mover and any drives used for NDMP (network data management protocol) operations.

Deleting a NAS node

To delete a NAS (network attached storage) node, first delete any file spaces for the node. Then delete any paths from the data mover before deleting the data mover.

1. Delete any virtual file space definitions for the node.
2. Enter the following command:

```
remove node nas1
```

Managing data movers used in NDMP operations

You can update, query, and delete the data movers that you define for NAS (network attached storage) file servers.

For example, if you shut down a NAS file server for maintenance, you might want to take the data mover offline.

1. Query your data movers to identify the data mover for the NAS file server that you want to maintain.

```
query datamover nasnode1
```
2. Issue the following command to make the data mover offline:

```
update datamover nasnode1 online=no
```

To delete the data mover, you must first delete any path definitions in which the data mover has been used as the source.

3. Issue the following command to delete the data mover:

```
delete datamover nasnode1
```

Attention: If the data mover has a path to the library, and you delete the data mover or make the data mover offline, you disable access to the library.

Dedicating a Tivoli Storage Manager drive to NDMP operations

If you are already using a drive for Tivoli Storage Manager operations, you can dedicate that drive to NDMP (network data management protocol) operations.

Remove Tivoli Storage Manager server access by deleting the path definition with the following command:

```
delete path server1 nasdrive1 srctype=server desttype=drive library=naslib
```

Storage pool management for NDMP operations

When NETAPPDUMP, CELERRADUMP, or NDMPDUMP are designated as the type of storage pool, managing the storage pools produced by NDMP (network data management protocol) operations is different from managing storage pools containing media for traditional Tivoli Storage Manager backups.

You can query and update storage pools. You cannot update the DATAFORMAT parameter.

Maintaining separate storage pools for data from different NAS vendors is suggested even though the data format for both is NDMPDUMP.

The following DEFINE STGPOOL and UPDATE STGPOOL parameters are ignored because storage pool hierarchies, reclamation, and migration are not supported for these storage pools:

- MAXSIZE
- NEXTSTGPOOL
- LOWMIG
- HIGHMIG
- MIGDELAY
- MIGCONTINUE
- RECLAIMSTGPOOL
- OVFLOLOCATION

Attention: Ensure that you do not accidentally use storage pools that have been defined for NDMP operations in traditional Tivoli Storage Manager operations. Be especially careful when assigning the storage pool name as the value for the DESTINATION parameter of the DEFINE COPYGROUP command. Unless the destination is a storage pool with the appropriate data format, the backup fails.

Managing table of contents

You can use several commands to manage different aspects of your data contents.

The SET TOCLOADRETENTION command can be used to specify the approximate number of minutes that an unreferenced table of contents (TOC) remains loaded in the Tivoli Storage Manager database. The Tivoli Storage Manager server-wide table of contents retention value will determine how long a loaded TOC is retained in the database after the latest access to information in the TOC.

Because TOC information is loaded into temporary database tables, this information is lost if the server is halted, even if the TOC retention period has not

elapsed. At installation, the retention time is set to 120 minutes. Use the `QUERY STATUS` command to see the TOC retention time.

Issue the `QUERY NASBACKUP` command to display information about the file system image objects that have been backed up for a specific NAS (network attached storage) node and file space. By issuing the command, you can see a display of all backup images generated by NDMP (network data management protocol) and whether each image has a corresponding table of contents.

Note: The Tivoli Storage Manager server may store a full backup in excess of the number of versions you specified, if that full backup has dependent differential backups. Full NAS backups with dependent differential backups behave like other base files with dependent subfiles. Due to retention time specified in the `RETAIN EXTRA` setting, the full NAS backup will not be expired, and the version will be displayed in the output of a `QUERY NASBACKUP` command. See “File expiration and expiration processing” on page 439 for details.

Use the `QUERY TOC` command to display files and directories in a backup image generated by NDMP. By issuing the `QUERY TOC` server command, you can display all directories and files within a single specified TOC. The specified TOC will be accessed in a storage pool each time the `QUERY TOC` command is issued because this command does not load TOC information into the Tivoli Storage Manager database. Then, use the `RESTORE NODE` command with the `FILELIST` parameter to restore individual files.

Configuring Tivoli Storage Manager for NDMP operations

Before beginning the configuration of Tivoli Storage Manager for NDMP (network data management protocol) operations, ensure that you register the required license.

Perform the following steps to configure the Tivoli Storage Manager for NDMP operations:

1. Set up the tape library and media. See “Setting up tape libraries for NDMP operations” on page 179, where the following steps are described in more detail.
 - a. Attach the SCSI library to the NAS file server or to the Tivoli Storage Manager server, or attach the ACSLS library or 349X library to the Tivoli Storage Manager server.
 - b. Define the library with a library type of SCSI, ACSLS, or 349X.
 - c. Define a device class for the tape drives.
 - d. Define a storage pool for NAS backup media.
 - e. Define a storage pool for storing a table of contents. This step is optional.
2. Configure Tivoli Storage Manager policy for managing NAS image backups. See “Configuring Tivoli Storage Manager policy for NDMP operations” on page 174.
3. Register a NAS file server node with the Tivoli Storage Manager server. See “Registering NAS nodes with the Tivoli Storage Manager server” on page 185.
4. Define a data mover for the NAS file server. See “Defining a data mover for the NAS file server” on page 186.
5. Define a path from either the Tivoli Storage Manager server or the NAS file server to the library. See “Defining a path to a library” on page 187.

6. Define the tape drives to Tivoli Storage Manager, and define the paths to those drives from the NAS file server and optionally from the Tivoli Storage Manager server. See “Defining tape drives and paths for NDMP operations” on page 186.
7. Check tapes into the library and label them. See “Labeling and checking tapes into the library” on page 188.
8. Set up scheduled backups for NAS file servers. This step is optional. See “Scheduling NDMP operations” on page 188.
9. Define a virtual file space name. This step is optional. See “Defining virtual file spaces” on page 188.
10. Configure for tape-to-tape copy to back up data. This step is optional. See “Tape-to-tape copy to back up data” on page 189.
11. Configure for tape-to-tape copy to move data to a different tape technology. This step is optional. See “Tape-to-tape copy to move data” on page 189.

Configuring Tivoli Storage Manager policy for NDMP operations

Policy lets you manage the number and retention time of NDMP (network data management protocol) image backup versions.

See “Configuring policy for NDMP operations” on page 485 for more information. Complete the following steps to configure Tivoli Storage Manager policy for NDMP operations:

1. Create a policy domain for NAS (network attached storage) file servers. For example, to define a policy domain that is named NASDOMAIN, enter the following command:


```
define domain nasdomain description='Policy domain for NAS file servers'
```
2. Create a policy set in that domain. For example, to define a policy set named STANDARD in the policy domain named NASDOMAIN, issue the following command:


```
define policyset nasdomain standard
```
3. Define a management class, and then assign the management class as the default for the policy set. For example, to define a management class named MC1 in the STANDARD policy set, and assign it as the default, issue the following commands:


```
define mgmtclass nasdomain standard mc1
assign defmgmtclass nasdomain standard mc1
```
4. Define a backup copy group in the default management class. The destination must be the storage pool you created for backup images produced by NDMP operations. In addition, you can specify the number of backup versions to retain. For example, to define a backup copy group for the MC1 management class where up to four versions of each file system are retained in the storage pool named NASPOOL, issue the following command:


```
define copygroup nasdomain standard mc1 destination=naspool verexists=4
```

If you also chose the option to create a table of contents, TOCDESTINATION must be the storage pool you created for the table of contents.

```
define copygroup nasdomain standard mc1 destination=naspool
tocdestination=tocpool verexists=4
```

Attention: When defining a copy group for a management class to which a file system image produced by NDMP will be bound, be sure that the DESTINATION parameter specifies the name of a storage pool that is defined for NDMP operations. If the DESTINATION parameter specifies an invalid storage pool, backups via NDMP will fail.

5. Activate the policy set. For example, to activate the STANDARD policy set in the NASDOMAIN policy domain, issue the following command:

```
activate policysat nasdomain standard
```

The policy is ready to be used. Nodes are associated with Tivoli Storage Manager policy when they are registered. For more information, see “Registering NAS nodes with the Tivoli Storage Manager server” on page 185.

Policy for backups initiated with the client interface

When a client node initiates a backup, the policy is affected by the option file for that client node.

You can control the management classes that are applied to backup images produced by NDMP (network data management protocol) operations regardless of which node initiates the backup. You can do this by creating a set of options to be used by the client nodes. The option set can include an `include.fs.nas` statement to specify the management class for NAS (network attached storage) file server backups. See “Creating client option sets on the server” on page 411 for more information.

Determining the location of NAS backup

When Tivoli Storage Manager uses NDMP (network data management protocol) to protect NAS (network attached storage) file servers, the Tivoli Storage Manager server controls operations while the NAS file server transfers the data, either to an attached library or directly to the Tivoli Storage Manager server.

You can also use a backup-archive client to back up a NAS file server by mounting the NAS file-server file system on the client machine (with either an NFS [network file system] mount or a CIFS [common internet file system] map) and then backing up as usual. Table 16 compares the three backup-and-restore methods.

Note: You can use a single method or a combination of methods in your individual storage environment.

Table 16. Comparing methods for backing up NDMP data

Property	NDMP: Filer to server	NDMP: Filer to attached library	Backup-archive client to server
Network data traffic	All backup data goes across the LAN from the NAS file server to the server.	The server controls operations remotely, but the NAS device moves the data locally.	All backup data goes across the LAN from the NAS device to the client and then to the server.
File server processing during backup	Less file server processing is required, compared to the backup-archive client method, because the backup does not use file access protocols such as NFS and CIFS.	Less file server processing is required, compared to the backup-archive client method, because the backup does not use file access protocols such as NFS and CIFS.	More file server processing is required because file backups require additional overhead for file access protocols such as NFS and CIFS.

Table 16. Comparing methods for backing up NDMP data (continued)

Property	NDMP: Filer to server	NDMP: Filer to attached library	Backup-archive client to server
Distance between devices	The Tivoli Storage Manager server must be within SCSI or Fibre Channel range of the tape library.	The Tivoli Storage Manager server can be distant from the NAS file server and the tape library.	The Tivoli Storage Manager server must be within SCSI or Fibre Channel range of the tape library.
Firewall considerations	More stringent than filer-to-attached-library because communications can be initiated by either the Tivoli Storage Manager server or the NAS file server.	Less stringent than filer-to-server because communications can be initiated only by the Tivoli Storage Manager server.	Client passwords and data are encrypted.
Security considerations	Data is sent unencrypted from NAS file server to the Tivoli Storage Manager server.	Method must be used in a trusted environment because port numbers are not secure.	Port number configuration allows for secure administrative sessions within a private network.
Load on the Tivoli Storage Manager server	Higher CPU workload is required to manage all back end data processes (for example, migration).	Lower CPU workload is required because migration and reclamation are not supported.	Higher CPU workload is required to manage all back end data processes.
Backup of primary storage pools to copy storage pools	Data can be backed up only to copy storage pools that have the NATIVE data format.	Data can be backed up only to copy storage pools that have the same NDMP data format (NETAPPDUMP, CELERRADUMP, or NDMPDUMP).	Data can be backed up only to copy storage pools that have the NATIVE data format.
Restore of primary storage pools and volumes from copy storage pools	Data can be restored only to storage pools and volumes that have the NATIVE data format.	Data can be restored only to storage pools and volumes that have the same NDMP format.	Data can be restored only to storage pools and volumes that have the NATIVE data format.
Moving NDMP data from storage pool volumes	Data can be moved to another storage pool only if it has a NATIVE data format.	Data can be moved to another storage pool only if it has the same NDMP data format.	Data can be moved to another storage pool only if it has a NATIVE data format.
Migration from one primary storage pool to another	Supported	Not supported	Supported
Reclamation of a storage pool	Supported	Not supported	Supported
Simultaneous-write operations during backups	Not supported	Not supported	Supported
Export and import operations	Not supported	Not supported	Supported
Backup set generation	Not supported	Not supported	Supported
Cyclic Redundancy Checking (CRC) when data is moved using Tivoli Storage Manager processes	Supported	Not supported	Supported
Validation using Tivoli Storage Manager audit commands	Supported	Not supported	Supported

Table 16. Comparing methods for backing up NDMP data (continued)

Property	NDMP: Filer to server	NDMP: Filer to attached library	Backup-archive client to server
Disaster recovery manager	Supported	Supported	Supported

Tape libraries and drives for NDMP operations

Most of the planning required to implement backup and recovery operations that use NDMP (network data management protocol) is related to device configuration. You have choices about how to connect and use the libraries and drives.

Many of the configuration choices you have for libraries and drives are determined by the hardware features of your libraries. You can set up NDMP operations with any supported library and drives. However, the more features your library has, the more flexibility you can exercise in your implementation.

You might start by answering the following questions:

- What type of library (SCSI, ACSLS, or 349X) will you use?
- If you are using a SCSI library, do you want to attach tape library robotics to the Tivoli Storage Manager server or to the network-attached storage (NAS) file server?
- Will you want to move your NDMP data to tape?
- How do you want to use the tape drives in the library?
 - Dedicate all tape drives to NDMP operations.
 - Dedicate some tape drives to NDMP operations and others to traditional Tivoli Storage Manager operations.
 - Share tape drives between NDMP operations and traditional Tivoli Storage Manager operations.
- Will you back up data tape-to-tape for disaster recovery functions?
- Will you send backup data to a single Tivoli Storage Manager server instead of attaching a tape library to each NAS device?
- Do you want to keep all hardware on the Tivoli Storage Manager server and send NDMP data over the LAN?

Determining library drive usage when backing up to NAS-attached libraries

Drives can be used for multiple purposes because of the flexible configurations allowed by Tivoli Storage Manager. For NDMP (network data management protocol) operations, the NAS (network attached storage) file server must have access to the drive. The Tivoli Storage Manager server can also have access to the same drive, depending on your hardware connections and limitations.

All drives are defined to the Tivoli Storage Manager server. However, the same drive may be defined for both traditional Tivoli Storage Manager operations and NDMP operations. Figure 11 on page 178 illustrates one possible configuration. The Tivoli Storage Manager server has access to drives 2 and 3, and each NAS file server has access to drives 1 and 2.

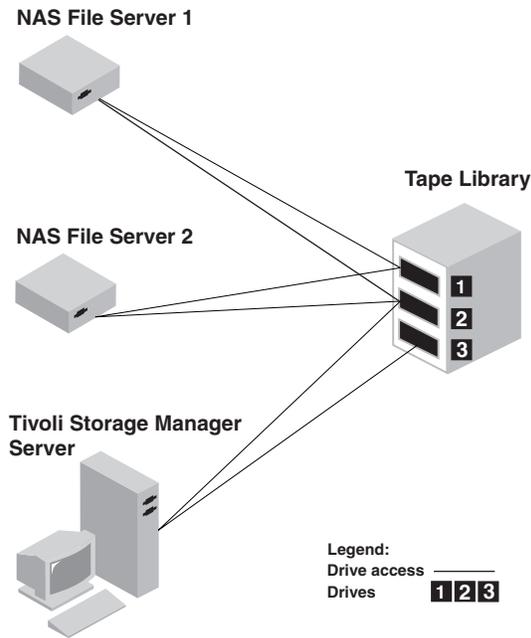


Figure 11. Tivoli Storage Manager drive usage example

To create the configuration shown in Figure 11, perform the following steps:

1. Define all three drives to Tivoli Storage Manager.
2. Define paths from the Tivoli Storage Manager server to drives 2 and 3. Because drive 1 is not accessed by the server, no path is defined.
3. Define each NAS file server as a separate data mover.
4. Define paths from each data mover to drive 1 and to drive 2.

To use the Tivoli Storage Manager back end data movement operations, the Tivoli Storage Manager server requires two available drive paths from a single NAS data mover. The drives can be in different libraries and can have different device types that are supported by NDMP. You can make copies between two different tape devices, for example, the source tape drive can be an DLT drive in a library and the target drive can be an LTO drive in another library.

During Tivoli Storage Manager back end data movements, the Tivoli Storage Manager server locates a NAS data mover that supports the same data format as the data to be copied from and that has two available mount points and paths to the drives. If the Tivoli Storage Manager server cannot locate such a data mover, the requested data movement operation is not performed. The number of available mount points and drives depends on the mount limits of the device classes for the storage pools involved in the back end data movements.

If the back end data movement function supports multiprocessing, each concurrent Tivoli Storage Manager back end data movement process requires two available mount points and two available drives. To run two Tivoli Storage Manager processes concurrently, at least four mount points and four drives must be available.

See “Defining tape drives and paths for NDMP operations” on page 186 for more information.

Setting up tape libraries for NDMP operations

You must complete several tasks to set up a tape library for NDMP (network data management protocol) operations.

Perform the following steps to set up tape libraries for NDMP operations:

1. Connect the library and drives for NDMP operations.
 - a. Connect the SCSI library. Before setting up a SCSI tape library for NDMP operations, you should have already determined whether you want to attach your library robotics control to the Tivoli Storage Manager server or to the NAS (network attached storage) file server. See “Tape libraries and drives for NDMP operations” on page 177. Connect the SCSI tape library robotics to the Tivoli Storage Manager server or to the NAS file server. See the manufacturer's documentation for instructions.

Library Connected to Tivoli Storage Manager: Make a SCSI or Fibre Channel connection between the Tivoli Storage Manager server and the library robotics control port. Then connect the NAS file server with the drives you want to use for NDMP operations.

Library Connected to NAS File Server: Make a SCSI or Fibre Channel connection between the NAS file server and the library robotics and drives.

- b. Connect the ACSLS Library. Connect the ACSLS tape library to the Tivoli Storage Manager server.
 - c. Connect the 349X Library. Connect the 349X tape library to the Tivoli Storage Manager server.
2. Define the library for NDMP operations. (The library has to be a single device type, not a mixed device one.)

SCSI Library

```
define library tsmlib libtype=scsi
```

ACSLs Library

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

349X Library

```
define library tsmlib libtype=349x
```

3. Define a device class for NDMP operations. Create a device class for NDMP operations. A device class defined with a device type of NAS is not explicitly associated with a specific drive type (for example, 3570 or 8 mm). However, we recommend that you define separate device classes for different drive types.

In the device class definition:

- Specify NAS as the value for the DEVTYPE parameter.
- Specify 0 as the value for the MOUNTRETENTION parameter. MOUNTRETENTION=0 is required for NDMP operations.
- Specify a value for the ESTCAPACITY parameter.

For example, to define a device class named NASCLASS for a library named NASLIB and media whose estimated capacity is 40 GB, issue the following command:

```
define devclass nasclass devtype=nas library=naslib mountretention=0  
estcapacity=40g
```

4. Define a storage pool for NDMP media. When NETAPPDUMP, CELERRADUMP, or NDMPDUMP is designated as the type of storage pool, managing the storage pools produced by NDMP operations is different from

managing storage pools containing media for traditional Tivoli Storage Manager backups. Tivoli Storage Manager operations use storage pools defined with a NATIVE or NONBLOCK data format. If you select NETAPPDUMP, CELERRADUMP, or NDMPDUMP, NDMP operations require storage pools with a data format that matches the NAS file server and the selected backup method. Maintaining separate storage pools for data from different NAS vendors is recommended, even though the data format for both is NDMPDUMP. For example, to define a storage pool named NDMPPool for a file server which is neither a NetApp nor a Celerra file server, issue the following command:

```
define stgpool ndmppool nasclass maxscratch=10 dataformat=ndmpdump
```

To define a storage pool named NASPOOL for a NetApp file server, issue the following command:

```
define stgpool naspool nasclass maxscratch=10 dataformat=netappdump
```

To define a storage pool named CELERRAPool for an EMC Celerra file server, issue the following command:

```
define stgpool celerrapool nasclass maxscratch=10 dataformat=celerradump
```

Attention: Ensure that you do not accidentally use storage pools that have been defined for NDMP operations in traditional Tivoli Storage Manager operations. Be especially careful when assigning the storage pool name as the value for the DESTINATION parameter of the DEFINE COPYGROUP command. Unless the destination is a storage pool with the appropriate data format, the backup will fail.

5. Define a storage pool for a table of contents. If you plan to create a table of contents, you should also define a disk storage pool in which to store the table of contents. You must set up policy so that the Tivoli Storage Manager server stores the table of contents in a different storage pool from the one where the backup image is stored. The table of contents is treated like any other object in that storage pool. This step is optional.

For example, to define a storage pool named TOCPool for a DISK device class, issue the following command:

```
define stgpool tocpool disk
```

Then, define volumes for the storage pool. For more information see:

“Configuring random access volumes on disk devices” on page 75.

For more information on connecting libraries, see Chapter 5, “Using devices with the server system,” on page 79.

Attaching tape library robotics for NAS-attached libraries

If you have decided to back up your network-attached storage (NAS) data to a library directly attached to the NAS device and are using a SCSI tape library, one of the first steps in planning for NDMP (network data management protocol) operations is to determine where to attach it.

You must determine whether to attach the library robotics to the Tivoli Storage Manager server or to the NAS file server. Regardless of where you connect library robotics, tape drives must always be connected to the NAS file server for NDMP operations.

Distance and your available hardware connections are factors to consider for SCSI libraries. If the library does not have separate ports for robotics control and drive access, the library must be attached to the NAS file server because the NAS file

server must have access to the drives. If your SCSI library has separate ports for robotics control and drive access, you can choose to attach the library robotics to either the Tivoli Storage Manager server or the NAS file server. If the NAS file server is at a different location from the Tivoli Storage Manager server, the distance may mean that you must attach the library to the NAS file server.

Whether you are using a SCSI, ACSLS, or 349X library, you have the option of dedicating the library to NDMP operations, or of using the library for NDMP operations as well as most traditional Tivoli Storage Manager operations.

Table 17. Summary of configurations for NDMP operations

Configuration	Distance between Tivoli Storage Manager server and library	Library sharing	Drive sharing between Tivoli Storage Manager and NAS file server	Drive sharing between NAS file servers	Drive sharing between storage agent and NAS file server
Configuration 1 (SCSI library connected to the Tivoli Storage Manager server)	Limited by SCSI or FC connection	Supported	Supported	Supported	Supported
Configuration 2 (SCSI library connected to the NAS file server)	No limitation	Not supported	Supported	Supported	Not supported
Configuration 3 (349X library)	May be limited by 349X connection	Supported	Supported	Supported	Supported

Configuration 1: SCSI library connected to the Tivoli Storage Manager server

In this configuration, the tape library must have separate ports for robotics control and for drive access. In addition, the library must be within Fibre-Channel range or SCSI bus range of both the Tivoli Storage Manager server and the network-attached storage (NAS) file server.

In this configuration, the Tivoli Storage Manager server controls the SCSI library through a direct, physical connection to the library robotics control port. For NDMP (network data management protocol) operations, the drives in the library are connected directly to the NAS file server, and a path must be defined from the NAS data mover to each of the drives to be used. The NAS file server transfers data to the tape drive at the request of the Tivoli Storage Manager server. To also use the drives for Tivoli Storage Manager operations, connect the Tivoli Storage Manager server to the tape drives and define paths from the Tivoli Storage Manager server to the tape drives. This configuration also supports a Tivoli Storage Manager storage agent having access to the drives for its LAN-free operations, and the Tivoli Storage Manager server can be a library manager.

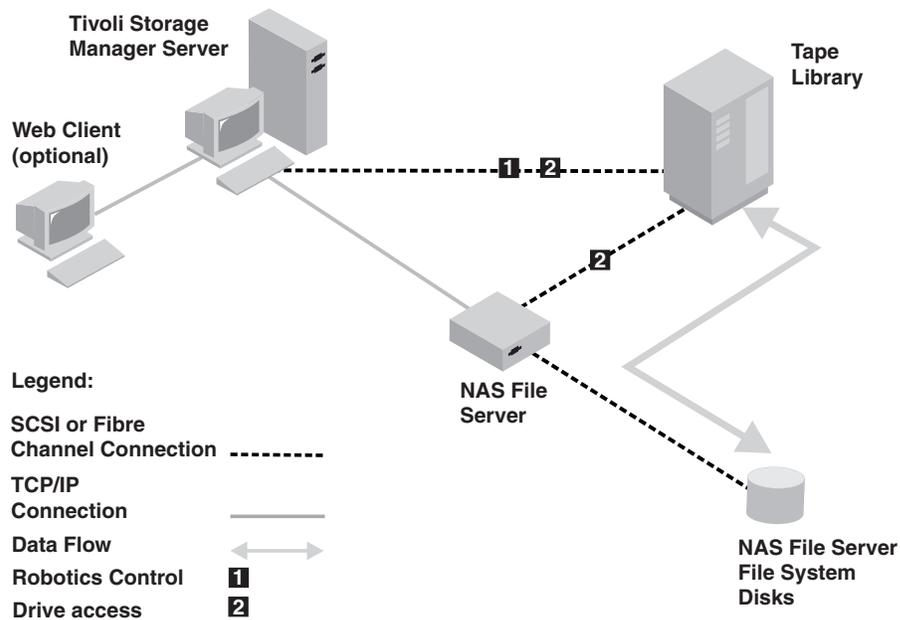


Figure 12. Configuration 1: SCSI library connected to Tivoli Storage Manager server

Configuration 2: SCSI library connected to the NAS file server

In this configuration, the library robotics and the drives must be physically connected directly to the NAS (network attached storage) file server, and paths must be defined from the NAS data mover to the library and drives. No physical connection is required between the Tivoli Storage Manager server and the SCSI library.

The Tivoli Storage Manager server controls library robotics by sending library commands across the network to the NAS file server. The NAS file server passes the commands to the tape library. Any responses generated by the library are sent to the NAS file server, and passed back across the network to the Tivoli Storage Manager server. This configuration supports a physically distant Tivoli Storage Manager server and NAS file server. For example, the Tivoli Storage Manager server could be in one city, while the NAS file server and tape library are in another city.

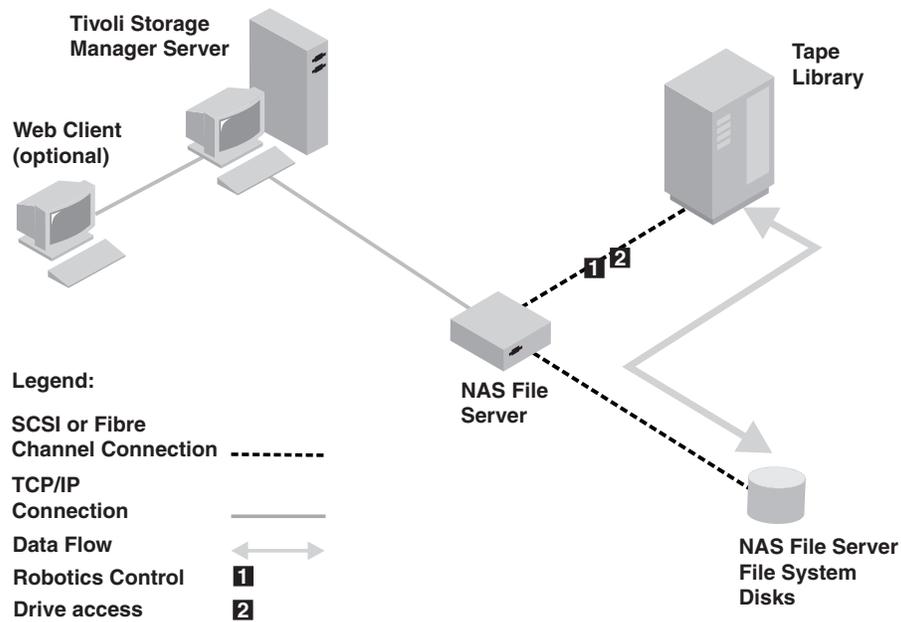


Figure 13. Configuration 2: SCSI library connected to the NAS file server

Configuration 3: 349x library connected to the Tivoli Storage Manager server

For this configuration, you connect the tape library to the system as for traditional operations.

In this configuration, the 349X tape library is controlled by the Tivoli Storage Manager server. The Tivoli Storage Manager server controls the library by passing the request to the 349X library manager through TCP/IP.

In order to perform NAS (network attached storage) backup or restore operations, the NAS file server must be able to access one or more tape drives in the 349X library. Any tape drives used for NAS operations must be physically connected to the NAS file server, and paths need to be defined from the NAS data mover to the drives. The NAS file server transfers data to the tape drive at the request of the Tivoli Storage Manager server. Follow the manufacturer's instructions to attach the device to the server system.

This configuration supports a physically distant Tivoli Storage Manager server and NAS file server. For example, the Tivoli Storage Manager server could be in one city, while the NAS file server and tape library are in another city.

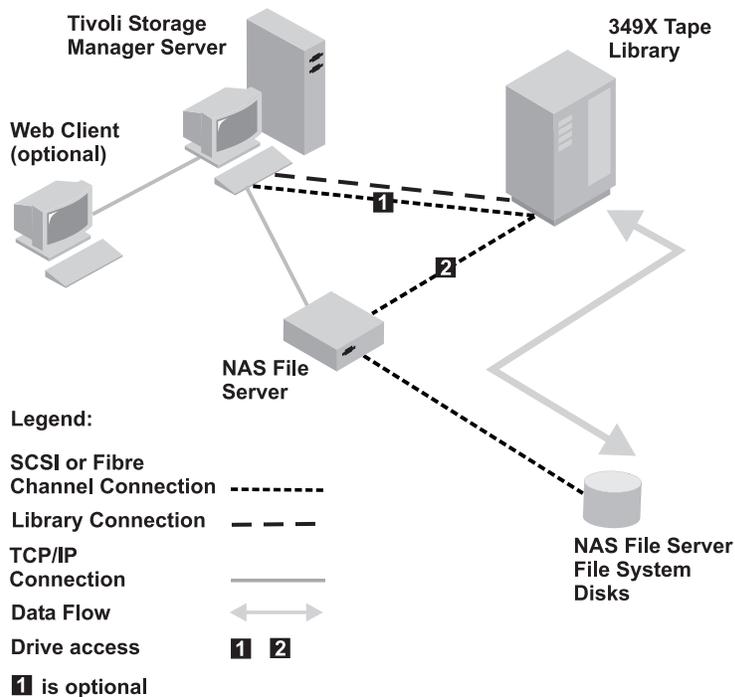


Figure 14. Configuration 3: 349x library connected to the Tivoli Storage Manager server

See Chapter 5, “Using devices with the server system,” on page 79 for more information.

Configuration 4: ACSLS library connected to the Tivoli Storage Manager server

For this configuration, connect the tape library to the system as you do for traditional Tivoli Storage Manager operations.

The ACSLS (automated cartridge system library software) tape library is controlled by the Tivoli Storage Manager server. The Tivoli Storage Manager server controls the library by passing the request to the ACSLS library server through TCP/IP. The ACSLS library supports library sharing and LAN-free operations.

In order to perform NAS (network attached storage) backup or restore operations, the NAS file server must be able to access one or more tape drives in the ACSLS library. Any tape drives used for NAS operations must be physically connected to the NAS file server, and any paths need to be defined from the NAS data mover to the drives. The NAS file server transfers data to the tape drive at the request of the Tivoli Storage Manager server. Follow the manufacturer's instructions to attach the device to the server system.

This configuration supports a physically distant Tivoli Storage Manager server and NAS file server. For example, the Tivoli Storage Manager server could be in one city while the NAS file server and tape library are in another city.

To also use the drives for Tivoli Storage Manager operations, connect the Tivoli Storage Manager server to the tape drives and define paths from the Tivoli Storage Manager server to the tape drives.

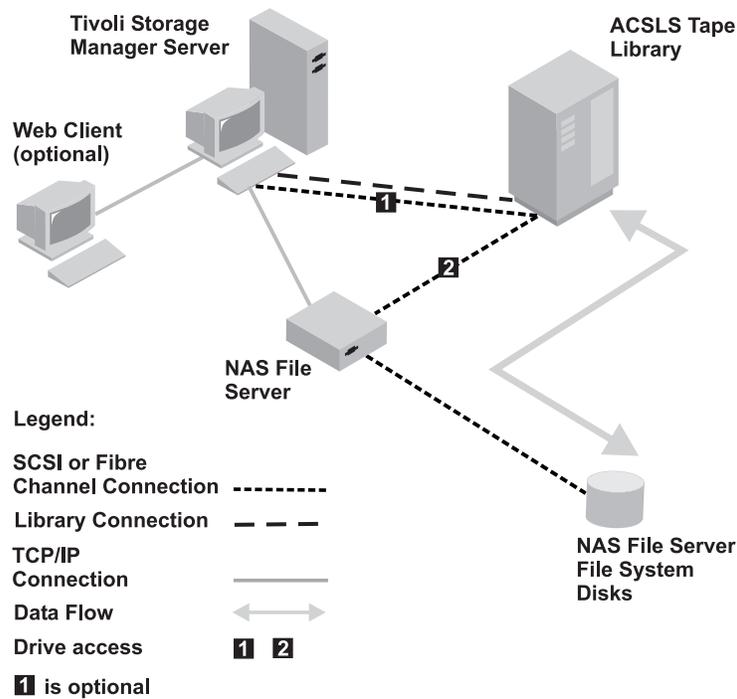


Figure 15. Configuration 4: ACSLS library connected to the Tivoli Storage Manager server

See Chapter 5, “Using devices with the server system,” on page 79 for more information.

Registering NAS nodes with the Tivoli Storage Manager server

Register the NAS (network attached storage) file server as a Tivoli Storage Manager node, specifying TYPE=NAS. This node name is used to track the image backups for the NAS file server.

To register a NAS file server as a node named NASNODE1, with a password of NASPWD1, in a policy domain named NASDOMAIN, issue the following example command:

```
register node nasnode1 naspwd1 domain=nasdomain type=nas
```

If you are using a client option set, specify the option set when you register the node.

You can verify that this node is registered by issuing the following command:

```
query node type=nas
```

Important: You must specify TYPE=NAS so that only NAS nodes are displayed.

Defining a data mover for the NAS file server

Define a data mover for each NAS (network attached storage) file server, using NDMP (network data management protocol) operations in your environment. The data mover name must match the node name that you specified when you registered the NAS node to the Tivoli Storage Manager server.

To define a data mover for a NAS node named NASNODE1, enter the following example command:

```
define datamover nasnode1 type=nas haddress=netapp2 lladdress=10000 userid=root  
password=admin dataformat=netappdump
```

In this command:

- The high-level address is an IP address for the NAS file server, either a numerical address or a host name.
- The low-level address is the IP port for NDMP sessions with the NAS file server. The default is port number 10000.
- The user ID is the ID defined to the NAS file server that authorizes an NDMP session with the NAS file server (for this example, the user ID is the administrative ID for the NetApp file server).
- The password parameter is a valid password for authentication to an NDMP session with the NAS file server.
- The data format is NETAPPDUMP. This is the data format that the NetApp file server uses for tape backup. This data format must match the data format of the target storage pool.

Defining tape drives and paths for NDMP operations

Define the tape drives that you want to use in NDMP (network data management protocol) operations and the paths to those drives. Depending on your hardware and network connections, you can use the drives for only NDMP operations, or for both traditional Tivoli Storage Manager operations and NDMP operations.

Perform the following steps to define tape drives and paths for NDMP operations:

1. Define an example drive named NASDRIVE1 for the library named NASLIB by issuing the following command:

```
define drive naslib nasdrive1 element=117
```

Important: When you define SCSI drives to the Tivoli Storage Manager server, the ELEMENT parameter must contain a number if the library has more than one drive. If the drive is shared between the NAS (network attached storage) file server and the Tivoli Storage Manager server, the element address is automatically detected. If the library is connected to a NAS file server only, there is no automatic detection of the element address and you must supply it. Element numbers are available from device manufacturers. Element numbers for tape drives are also available in the device support information available on the Tivoli Web site at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

2. Define a path for the drive. For example, if the drive is to be used only for NDMP operations, issue the following command:

```
define path nasnode1 nasdrive1 srctype=datamover desttype=drive  
library=naslib device=rst01
```

Attention: For a drive connected only to the NAS file server, do not specify ASNEEDED for the CLEANFREQUENCY parameter of the DEFINE DRIVE command.

For example, if a drive is to be used for both Tivoli Storage Manager and NDMP operations, issue the following commands:

```
define path server1 nasdrive1 srctype=server desttype=drive
  library=naslib device=/dev/rmt0
define path nasnode1 nasdrive1 srctype=datamover desttype=drive
  library=naslib device=rst01
```

Defining a path to a library

Define a path to the SCSI library from either the Tivoli Storage Manager or the NAS (network attached storage) file server.

1. For a SCSI Library connected to Tivoli Storage Manager, issue the following example command to define a path from the server, named SERVER1, to the SCSI library named TSMLIB:

```
define path server1 tsmlib srctype=server desttype=library
  device=/dev/tsm SCSI/lb1
```

2. For a SCSI library connected to a NAS file server, issue the following example command to define a path between a NetApp NAS data mover named NASNODE1 and a library named NASLIB:

```
define path nasnode1 naslib srctype=datamover desttype=library device=mc0
```

3. For a 349X library, define a path to the library from the Tivoli Storage Manager server. For example, issue the following command to define a path from the server, named SERVER1, to the 349X library named TSMLIB:

```
define path server1 tsmlib srctype=server desttype=library
  device=library1
```

Attention: DEFINE PATH is not needed for an automated cartridge system library software (ACSL) library.

Obtaining special file names for path definitions:

When you are creating paths, you must provide special file names for tape libraries and drives.

For paths from a NAS data mover, the value of the DEVICE parameter in the DEFINE PATH command is the name by which the NAS (network attached storage) file server knows a library or drive. You can obtain these names, known as special file names, by querying the NAS file server. 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.

1. To obtain the special file names for tape libraries on a Netapp Release ONTAP 10.0 GX, or later, file server, connect to the file server using telnet and issue the SYSTEM HARDWARE TAPE LIBRARY SHOW command. To obtain the special file names for tape drives on a Netapp Release ONTAP 10.0 GX, or later, file server, connect to the file server using telnet and issue the SYSTEM HARDWARE TAPE DRIVE SHOW command. For details about these commands, see the Netapp ONTAP GX file server product documentation.
2. For releases earlier than Netapp Release ONTAP 10.0 GX, continue to use the SYSCONFIG command. For example, to display the device names for tape libraries, connect to the file server using telnet and issue the following command:

```
sysconfig -m
```

To display the device names for tape drives, issue the following command:
sysconfig -t

3. For the Celerra file server, connect to the Celerra control workstation using telnet. To see the devices attached to a particular data mover, use the “server_devconfig” command on the control station:

```
server_devconfig server_# -p -s -n
```

The SERVER_# is the data mover on which the command should be run.

Labeling and checking tapes into the library

You must label the tapes and check them into the tape library.

These tasks are the same as for other libraries. For more information, see:

“Labeling removable media volumes” on page 134

Scheduling NDMP operations

You can schedule the backup or restore of images produced by NDMP (network data management protocol) operations by using administrative schedules that process the BACKUP NODE or RESTORE NODE administrative commands.

The BACKUP NODE and RESTORE NODE commands can be used only for nodes of TYPE=NAS. See “Backing up and restoring NAS file servers using NDMP” on page 189 for information about the commands.

For example, to create an administrative schedule called NASSCHED to back up all file systems for a node named NASNODE1, enter the following:

```
define schedule nassched type=administrative cmd='backup node nasnode1' active=yes  
starttime=20:00 period=1 perunits=days
```

The schedule is active, and is set to run at 8:00 p.m. every day. See Chapter 20, “Automating server operations,” on page 571 for more information.

Defining virtual file spaces

Use a virtual file space definition to perform NAS (network attached storage) directory level backups. In order to reduce backup and restore times for large file systems, map a directory path from a NAS file server to a virtual file space name on the Tivoli Storage Manager server.

To create a virtual file space name for the directory path on the NAS device, issue the DEFINE VIRTUALFSMAPPING command:

```
define virtualfsmapping nas1 /mikesdir /vol/vol1 /mikes
```

This command defines a virtual file space name of /MIKESDIR on the server which represents the directory path of /VOL/VOL1/MIKES on the NAS file server represented by node NAS1. See “Directory-level backup and restore for NDMP operations” on page 195 for more information.

Tape-to-tape copy to back up data

When using NDMP (network data management protocol) tape-to-tape function to back up data, the library type can be SCSI, 349X, or ACSLS (automated cartridge system library software). Drives can be shared between the NAS (network attached storage) devices and the Tivoli Storage Manager server.

Note: When using the NDMP tape-to-tape copy function, your configuration setup could affect the performance of the Tivoli Storage Manager back end data movement.

To have one NAS device with paths to four drives in a library, use the MOVE DATA command after you are done with your configuration setup. This moves data on the volume VOL1 to any available volumes in the same storage pool as VOL1:

```
move data vol1
```

Tape-to-tape copy to move data

In order to move data from an old tape technology to a new tape technology, using NDMP (network data management protocol) tape-to-tape copy operation, perform the steps below as well as the regular steps in your configuration setup.

Note: When using the NDMP tape-to-tape copy function, your configuration setup could affect the performance of the Tivoli Storage Manager back end data movement.

1. Define one drive in the library, lib1, that has old tape technology:

```
define drive lib1 drv1 element=1035
```
2. Define one drive in the library, lib2, that has new tape technology:

```
define drive lib2 drv1 element=1036
```
3. Move data on volume vol1 in the primary storage pool to the volumes in another primary storage pool, nasprimpool2:

```
move data vol1 stgpool=nasprimpool2
```

Backing up and restoring NAS file servers using NDMP

After you have completed the steps to configure Tivoli Storage Manager for NDMP (network data management protocol) operations, you are ready to begin using NDMP.

Use either a client interface or an administrative interface to perform a file system image backup. For example, to use the Windows backup-archive client interface to back up a file system named /vol/vol1 on a NAS (network attached storage) file server named NAS1, issue the following command:

```
dsmc backup nas -nasnodename=nas1 {/vol/vol1}
```

For more information on the command, see the *Tivoli Storage Manager Backup-Archive Clients Installation and User's Guide*.

Tip: Whenever you use the client interface, you are asked to authenticate yourself as a Tivoli Storage Manager administrator before the operation can begin. The administrator ID must have at least client owner authority for the NAS node. You can perform the same backup operation with a server interface. For example, from the administrative command-line client, back up the file system named /vol/vol1 on a NAS file server named NAS1, by issuing the following command:

```
backup node nas1 /vol/vol1
```

Note: The BACKUP NAS and BACKUP NODE commands do not include snapshots. To back up snapshots see “Backing up and restoring with snapshots” on page 195.

You can restore the image using either interface. Backups are identical whether they are backed up using a client interface or a server interface. For example, suppose you want to restore the image backed up in the previous examples. For this example the file system named /vol/vol1 is being restored to /vol/vol2. Restore the file system with the following command, issued from a Windows backup-archive client interface:

```
dsmc restore nas -nasnodename=nas1 {/vol/vol1} {/vol/vol2}
```

You can choose to restore the file system, using a server interface. For example, to restore the file system name /vol/vol1 to file system /vol/vol2, for a NAS file server named NAS1, enter the following command:

```
restore node nas1 /vol/vol1 /vol/vol2
```

You can restore data from one NAS vendor system to another NAS vendor system when you use the NDMPDUMP data format, but you should either verify compatibility between systems or maintain a separate storage pool for each NAS vendor.

NAS file servers; backups to a single Tivoli Storage Manager server

If you have several NAS (network attached storage) file servers located in different locations, you might prefer to send the backup data to a single Tivoli Storage Manager server rather than attaching a tape library to each NAS device.

When you store NAS backup data in the Tivoli Storage Manager server's storage hierarchy, you can apply Tivoli Storage Manager back end data management functions. Migration, reclamation, and disaster recovery are among the supported features when using the NDMP file server to Tivoli Storage Manager server option.

In order to back up a NAS device to a Tivoli Storage Manager native storage pool, set the destination storage pool in the copy group to point to the desired native storage pool. The destination storage pool provides the information about the library and drives used for backup and restore. You should ensure that there is sufficient space in your target storage pool to contain the NAS data, which can be backed up to sequential, disk, or file type devices. Defining a separate device class is not necessary.

If you are creating a table of contents, a management class should be specified with the TOCDESTINATION parameter in the DEFINE and UPDATE COPYGROUP commands. When backing up a NAS file server to Tivoli Storage Manager native pools, the TOCDESTINATION may be the same as the destination of the NDMP (network data management protocol) data.

Firewall considerations are more stringent than they are for filer-to-attached-library because communications can be initiated by either the Tivoli Storage Manager server or the NAS file server. NDMP tape servers run as threads within the Tivoli Storage Manager server and the tape server accepts connections on port of 10001. This port number can be changed through the following option in the Tivoli Storage Manager server options file: NDMPPORTRANGE port-number-low, port-number-high.

During NDMP filer-to-server backup operations, you can use the `NDMPPREFDATAINTERFACE` option to specify which network interface the Tivoli Storage Manager server uses to receive NDMP backup data. The value for this option is a hostname 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.

Before using this option, verify that your NAS device supports NDMP operations that use a different network interface for NDMP control and NDMP data connections. NDMP control connections are used by Tivoli Storage Manager to authenticate with an NDMP server and monitor an NDMP operation while NDMP data connections are used to transmit and receive backup data during NDMP operations. You must still configure your NAS device to route NDMP backup and restore data to the appropriate network interface.

When enabled, the `NDMPPREFDATAINTERFACE` option affects all subsequent NDMP filer-to-server operations. It does not affect NDMP control connections because they use the system's default network interface. You can update this server option without stopping and restarting the server by using the `SETOPT` command (Set a server option for dynamic update).

NetApp file servers provide an NDMP option (`ndmpd.preferred_interface`) to change the interface used for NDMP data connections. Refer to the documentation that came with your NAS device for more information.

See “Performing NDMP filer to Tivoli Storage Manager server backups” for steps on how to perform NDMP filer-to-server backups.

See the Administrator's Reference for server option information.

Performing NDMP filer to Tivoli Storage Manager server backups

You can back up data to a single Tivoli Storage Manager server rather than attaching a tape library to each NAS device.

For a filer-to-server backup of a NAS file system, perform the following steps:

1. Set up a native storage pool for the NAS data by issuing the following command:

```
define stgpool naspool disk f=100g
```

Or, select an existing native storage pool with enough available space to hold your NAS backup data.

2. Set the copy destination to the storage pool defined previously and activate the associated policy set.

```
update copygroup standard standard standard destination=naspool  
tocdestination=naspool  
activate policysset standard standard
```

The destination for NAS data is determined by the destination in the copy group. The storage size estimate for NAS differential backups uses the occupancy of the file space, the same value that is used for a full backup. You can use this size estimate as one of the considerations in choosing a storage pool. One of the attributes of a storage pool is the `MAXSIZE` value, which indicates that data be sent to the `NEXT` storage pool if the `MAXSIZE` value is exceeded by the estimated size. Because NAS differential backups to Tivoli

Storage Manager native storage pools use the base file space occupancy size as a storage size estimate, differential backups end up in the same storage pool as the full backup. Depending on collocation settings, differential backups may end up on the same media as the full backup.

3. Set up a node and data mover for the NAS device. The data format signifies that the backup images created by this NAS device are a dump type of backup image in a NetApp specific format.

```
register node nas1 nas1 type=nas domain=standard
define datamover nas1 type=nas hla=nas1 user=root
password=***** dataformat=netappdump
```

The NAS device is now ready to be backed up to a Tivoli Storage Manager server storage pool. Paths may be defined to local drives, but the destination specified by the management class determines the target location for this backup operation.

4. Back up the NAS device to the Tivoli Storage Manager storage pool by issuing the following command:

```
backup node nas1 /vol/vol0
```

5. Restore a NAS device from the Tivoli Storage Manager storage pool by issuing the following command:

```
restore node nas1 /vol/vol0
```

File-level backup and restore for NDMP operations

When you do a backup via NDMP (network data management protocol), you can specify that the Tivoli Storage Manager server collect and store file-level information in a table of contents (TOC).

If you specify this option at the time of backup, you can later display the table of contents of the backup image. Through the backup-archive Web client, you can select individual files or directories to restore directly from the backup images generated.

Collecting file-level information requires additional processing time, network resources, storage pool space, temporary database space, and possibly a mount point during the backup. You should consider dedicating more space in the Tivoli Storage Manager server database. You must set up policy so that the Tivoli Storage Manager server stores the table of contents in a different storage pool from the one where the backup image is stored. The table of contents is treated like any other object in that storage pool.

You also have the option to do a backup via NDMP without collecting file-level restore information.

To allow creation of a table of contents for a backup via NDMP, you must define the TOCDESTINATION attribute in the backup copy group for the management class to which this backup image is bound. You cannot specify a copy storage pool or an active-data pool as the destination. The storage pool you specify for the TOC destination must have a data format of either NATIVE or NONBLOCK, so it cannot be the tape storage pool used for the backup image.

If you choose to collect file-level information, specify the TOC parameter in the BACKUP NODE server command. Or, if you initiate your backup using the client, you can specify the TOC option in the client options file, client option set, or client command line. You can specify NO, PREFERRED, or YES. When you specify

PREFERRED or YES, the Tivoli Storage Manager server stores file information for a single NDMP-controlled backup in a table of contents (TOC). The table of contents is placed into a storage pool. After that, the Tivoli Storage Manager server can access the table of contents so that file and directory information can be queried by the server or client. Use of the TOC parameter allows a table of contents to be generated for some images and not others, without requiring different management classes for the images.

See the *Administrator's Reference* for more information about the BACKUP NODE command.

To avoid mount delays and ensure sufficient space, use random access storage pools (DISK device class) as the destination for the table of contents. For sequential access storage pools, no labeling or other preparation of volumes is necessary if scratch volumes are allowed.

See “Managing table of contents” on page 172 for more information.

Interfaces for file-level restore

When you restore individual files and directories, you have the choice of using one of two interfaces to initiate the restore: the backup-archive Web client or the server interface.

Restore Using Backup-Archive Web Client

The backup-archive Web client requires that a table of contents exist in order to restore files and directories. The Web client must be on a Windows system. The Tivoli Storage Manager server accesses the table of contents from the storage pool and loads TOC information into a temporary database table. Then, you can use the backup-archive Web client to examine directories and files contained in one or more file system images, and select individual files or directories to restore directly from the backup images generated.

Restore Using Server Interface

- If you have a table of contents, use the QUERY NASBACKUP command to display information about backup images generated by NDMP (network data management protocol), and to see which images have a corresponding table of contents. Then, use the RESTORE NODE command with the FILELIST parameter.
- If you did not create a table of contents, the contents of the backup image cannot be displayed. You can restore individual files, directories, or both if you know the name of the file or directory, and in which image the backup is located. Use the RESTORE NODE command with the FILELIST parameter.

International characters for NetApp file servers

All systems that create or access data on a particular NAS (network attached storage) file server volume must do so in a manner compatible with the volume language setting.

You should install Data ONTAP 6.4.1 or later, if it is available, on your NetApp NAS file server in order to garner full support of international characters in the names of files and directories.

If your level of Data ONTAP is earlier than 6.4.1, you must have one of the following two configurations in order to collect and restore file-level information. Results with configurations other than these two are unpredictable. The Tivoli Storage Manager server will print a warning message (ANR4946W) during backup operations. The message indicates that the character encoding of NDMP file history messages is unknown, and UTF-8 will be assumed in order to build a table of contents. It is safe to ignore this message only for the following two configurations.

- Your data has directory and file names that contain only English (7-bit ASCII) characters.
- Your data has directory and file names that contain non-English characters and the volume language is set to the UTF-8 version of the proper locale (for example, de.UTF-8 for German).

If your level of Data ONTAP is 6.4.1 or later, you must have one of the following three configurations in order to collect and restore file-level information. Results with configurations other than these three are unpredictable.

- Your data has directory and file names that contain only English (7-bit ASCII) characters and the volume language is either not set or is set to one of these:
 - C (POSIX)
 - en
 - en_US
 - en.UTF-8
 - en_US.UTF-8
- Your data has directory and file names that contain non-English characters, and the volume language is set to the proper locale (for example, de.UTF-8 or de for German).

Tip: Using the UTF-8 version of the volume language setting is more efficient in terms of Tivoli Storage Manager server processing and table of contents storage space.

- You only use CIFS to create and access your data.

File level restore from a directory-level backup image

File-level restore is supported for directory-level backup images.

As with a NAS (network attached storage) file system backup, a table of contents (TOC) is created during a directory-level backup and you are able to browse the files in the image, using the Web client. The default is that the files are restored to the original location. During a file-level restore from a directory-level backup, however, you can either select a different file system or another virtual file space name as a destination.

For a TOC of a directory level backup image, the path names for all files are relative to the directory specified in the virtual file space definition, not the root of the file system.

Directory-level backup and restore

If you have a large NAS (network attached storage) file system, initiating a backup at a directory level will reduce backup and restore times and provide more flexibility in configuring your NAS backups. By defining virtual file spaces, a file system backup can be partitioned among several NDMP backup operations and multiple tape drives. You can also use different backup schedules to back up sub-trees of a file system.

The virtual file space name cannot be identical to any file system on the NAS node. If a file system is created on the NAS device with the same name as a virtual file system, a name conflict will occur on the Tivoli Storage Manager server when the new file space is backed up. See the Administrator's Reference for more information about virtual file space mapping commands.

Note: Virtual file space mappings are only supported for NAS nodes.

Directory-level backup and restore for NDMP operations

The DEFINE VIRTUALFSMAPPING command maps a directory path of a NAS (network attached storage) file server to a virtual file space name on the Tivoli Storage Manager server. After a mapping is defined, you can conduct NAS operations such as BACKUP NODE and RESTORE NODE, using the virtual file space names as if they were actual NAS file spaces.

To start a backup of the directory, issue the BACKUP NODE command specifying the virtual file space name instead of a file space name. To restore the directory subtree to the original location, run the RESTORE NODE command and specify the virtual file space name.

Virtual file space definitions can also be specified as the destination in a RESTORE NODE command. This allows you restore backup images (either file system or directory) to a directory on any file system of the NAS device.

You can use the Web client to select files for restore from a directory-level backup image because the Tivoli Storage Manager client treats the virtual file space names as NAS file spaces.

Backing up and restoring with snapshots

NDMP directory level backup gives you the ability to back up user created snapshots of a NAS file system; those are then stored as subdirectories. The snapshots can be taken at any time, and the backup to tape can be deferred to a more convenient time.

For example, to backup a snapshot created for a NetApp file system, perform the following:

1. On the console for the NAS device, issue the command to create the snapshot. SNAP CREATE is the command for a NetApp device.

```
snap create vol2 february17
```

This command creates a snapshot named FEBRUARY 17 of the `/vol/vol2` file system. The physical location for the snapshot data is in the directory `/vol/vol2/.snapshot/february17`. The stored location for snapshot data is dependent on the NAS vendor implementation. For NetApp, the SNAP LIST command can be used to display all snapshots for a given file system.

2. Define a virtual file space mapping definition on the Tivoli Storage Manager server for the snapshot data created in the previous step.

```
define virtualfilemapping nas1 /feb17snapshot /vol/vol2 /.snapshot/february17
```

This creates a virtual file space mapping definition named `/feb17snapshot`.

3. Back up the virtual file space mapping.

```
backup node nas1 /feb17snapshot mode=full toc=yes
```

4. After the backup is created, you can either restore the entire snapshot image or restore an individual file. Before restoring the data you can create a virtual file space mapping name for the target directory. You can select any file system name as a target. The target location in this example is the directory `/feb17snaprestore` on the file system `/vol/vol1`.

```
define virtualfilemapping nas1 /feb17snaprestore /vol/vol1 /feb17snaprestore
```

5. Issue the restore of the snapshot backup image.

```
restore node nas1 /feb17snapshot /feb17snaprestore
```

This restores a copy of the `/vol/vol2` file system to the directory `/vol/vol1/feb17snaprestore` in the same state as when the snapshot was created in the first step.

Backup and restore using NetApp SnapMirror to Tape feature

You can back up very large NetAppfile systems using the NetAppSnapMirror to Tape feature. Using a block-level copy of data for backup, the SnapMirror to Tape method is faster than a traditional Network Data Management Protocol (NDMP) full backup and can be used when NDMP full backups are impractical.

Use the NDMP SnapMirror to Tape feature as a disaster recovery option for copying very large NetAppfile systems to secondary storage. For most NetAppfile systems, use the standard NDMP full or differential backup method.

Using a parameter option on the BACKUP and RESTORE NODE commands, you can back up and restore file systems using SnapMirror to Tape. There are several limitations and restrictions on how SnapMirror images can be used. Consider the following guidelines before you use it as a backup method:

- You cannot initiate a SnapMirror to Tape backup or restore operation from the Tivoli Storage Manager Administration Center, Web client, or command-line client.
- You cannot perform differential backups of SnapMirror images.
- You cannot perform a directory-level backup using SnapMirror-to-Tape, thus Tivoli Storage Manager does not permit an SnapMirror to Tape backup operation on a server virtual filespace.
- You cannot perform an NDMP file-level restore operation from SnapMirror to Tape images. Therefore, a table of contents is never created during SnapMirror to Tape image backups.
- At the start of a SnapMirror to Tape copy operation, the file server generates a snapshot of the file system. NetApp provides an NDMP environment variable to

control whether this snapshot should be removed at the end of the SnapMirror to Tape operation. Tivoli Storage Manager always sets this variable to remove the snapshot.

- After a SnapMirror to Tape image is retrieved and copied to a NetAppfile system, the target file system is left configured as a SnapMirror partner. NetApp provides an NDMP environment variable to control whether this SnapMirror relationship should be broken. Tivoli Storage Manager always "breaks" the SnapMirror relationship during the retrieval. After the restore operation is complete, the target file system is in the same state as that of the original file system at the point-in-time of backup.

See the BACKUP NODE and RESTORE NODE commands in the *Administrator's Reference* for more information on SnapMirror to Tape feature.

NDMP backup operations using Celerra file server integrated checkpoints

When the Tivoli Storage Manager server initiates an NDMP backup operation on a Celerra data mover, the backup of a large file system might take several hours to complete. Without Celerra integrated checkpoints enabled, any changes occurring on the file system are written to the backup image.

As a result, the backup image includes changes made to the file system during the entire backup operation and is not a true point-in-time image of the file system.

If you are performing NDMP backups of Celerra file servers, you should upgrade the operating system of your data mover to Celerra file server version T5.5.25.1 or later. This version of the operating system allows enablement of integrated checkpoints for all NDMP backup operations from the Celerra Control Workstation. Enabling this feature ensures that NDMP backups represent true point-in-time images of the file system that is being backed up.

Refer to the Celerra file server documentation for instructions on enabling integrated checkpoints during all NDMP backup operations.

If your version of the Celerra file server operating system is earlier than version T5.5.25.1 and if you use NDMP to back up Celerra data movers, you should manually generate a snapshot of the file system using Celerra's command line checkpoint feature and then initiate an NDMP backup of the checkpoint file system rather than the original file system.

Refer to the Celerra file server documentation for instructions on creating and scheduling checkpoints from the Celerra control workstation.

Chapter 9. Defining device classes

A device class represents a device type that Tivoli Storage Manager can use to determine which types of devices and volumes are available to store client-node data in primary storage pools, copy storage pools and active-data pools. Device classes are also important for storing database backups and for exporting and importing data.

Sequential-access device types include tape, optical, and sequential-access disk. For random access storage, Tivoli Storage Manager supports only the DISK device class, which is defined by Tivoli Storage Manager.

To define a device class, use the DEFINE DEVCLASS command and specify the DEVTYPE parameter. The DEVTYPE parameter assigns a device type to the device class. You can define multiple device classes for each device type. For example, you might need to specify different attributes for different storage pools that use the same type of tape drive. Variations may be required that are not specific to the device, but rather to how you want to use the device (for example, mount retention or mount limit). For all device types other than FILE or SERVER, you must define libraries and drives to Tivoli Storage Manager before you define the device classes.

To update an existing device class definition, use the UPDATE DEVCLASS command. You can also delete a device class and query a device class using the DELETE DEVCLASS and QUERY DEVCLASS commands, respectively.

Task	Required Privilege Class
Define, update, or delete device classes	System or unrestricted storage
Request information about device classes	Any administrator

Remember:

- One device class can be associated with multiple storage pools, but each storage pool is associated with only one device class.
- If you include the DEVCONFIG option in the dsmserv.opt file, the files that you specify with that option are automatically updated with the results of the DEFINE DEVCLASS, UPDATE DEVCLASS, and DELETE DEVCLASS commands.
- Tivoli Storage Manager now allows SCSI libraries to include tape drives of more than one device type. When you define the device class in this environment, you must declare a value for the FORMAT parameter.

See the following topics:

Tasks
"Defining tape and optical device classes" on page 201
"Defining 3592 device classes" on page 204
"Defining device classes for removable media devices" on page 207
"Defining sequential-access disk (FILE) device classes" on page 207
"Defining LTO device classes" on page 211

Tasks
“Defining SERVER device classes” on page 214
“Defining device classes for StorageTek VolSafe devices” on page 215

For details about commands and command parameters, see the *Administrator's Reference*.

See “Mixed device types in libraries” on page 58 and “Mixed device types in libraries” on page 58 for additional information.

The examples in topics show how to perform tasks using the Tivoli Storage Manager command-line interface. For information about the commands, see the *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Sequential-access device types

Tivoli Storage Manager supports tape devices, magnetic disk devices, optical devices, removable media devices, and virtual volumes.

The following tables list supported devices, media types, and Tivoli Storage Manager device types.

For details and updates, see the following Web site: http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_Linux.html

Table 18. Tape devices

Examples	Media type	Device Type
IBM 3590, 3590E drives	IBM 3590 cartridges	3590
IBM 3592 drives	IBM 3592 cartridges	3592
IBM 7206-005	4 mm cartridges	4MM
IBM 7208-001 and 7208-011	8 mm cartridges	8MM
DLT2000, DLT4000, DLT7000 and DLT8000 drives	Digital linear tape (DLT) cartridges	DLT
Sun StorageTek SD-3, 9490, 9840, 9940, and T10000 drives	Tape cartridges	ECARTRIDGE
IBM 3580	LTO Ultrium cartridges	LTO
Tape drives supported by the NAS file server for backups	Unknown	NAS
IBM 7207	Quarter-inch tape cartridges	QIC
Sun StorageTek 9840 drives	Write-once read-many (WORM) tape cartridges	VOLSAFE

Table 19. Magnetic disk devices

Examples	Media type	Device type
Sequential-access disk	File system or storage volumes	FILE

Table 20. Removable media (file system) devices

Examples	Media type	Device Type
Removable media devices that are attached as local, removable file systems	Iomega Zip or Jaz, or CD media	REMOVABLEFILE

Table 21. Virtual volumes

Examples	Media type	Device type
Tivoli Storage Manager target server	Storage volumes or files archived in another Tivoli Storage Manager server	SERVER

Defining tape and optical device classes

Device class definitions for tapes include parameters that let you control storage operations.

Specifying the estimated capacity of tape and optical volumes

Tivoli Storage Manager also uses estimated capacity to determine when to begin reclamation storage pool volumes.

For tape and optical device classes, the default values selected by the server depend on the recording format used to write data to the volume. You can either accept the default for a given device type or specify a value.

To specify estimated capacity for tape volumes, use the ESTCAPACITY parameter when you define the device class or update its definition.

For more information about how Tivoli Storage Manager uses the estimated capacity value, see “How Tivoli Storage Manager fills volumes” on page 217.

Specifying recording formats for tape and optical media

You can specify the recording format used by Tivoli Storage Manager when writing data to tape and optical media.

To specify a recording format, use the FORMAT parameter when you define the device class or update its definition.

If all drives associated with that device class are identical, specify FORMAT=DRIVE. The server selects the highest format that is supported by the drive on which a volume is mounted.

If some drives associated with the device class support a higher density format than others, specify a format that is compatible with all drives. If you specify FORMAT=DRIVE, mount errors can occur. For example, suppose a device class uses two incompatible devices such as an IBM 7208-2 and an IBM 7208-12. The server might select the high-density recording format of 8500 for each of two new volumes. Later, if the two volumes are to be mounted concurrently, one fails because only one of the drives is capable of the high-density recording format.

If drives in a single SCSI library use different tape technologies (for example, DLT and LTO Ultrium), specify a unique value for the FORMAT parameter in each device class definition.

For a configuration example, see “Configuration with multiple drive device types” on page 93.

The recording format that Tivoli Storage Manager uses for a given volume is selected when the first piece of data is written to the volume. Updating the FORMAT parameter does not affect media that already contain data until those media are rewritten from the beginning. This process might happen after a volume is reclaimed or deleted, or after all of the data on the volume expires.

Associating library objects with device classes

A library contains the drives that can be used to mount the volume. Only one library can be associated with a given device class. However, multiple device classes can reference the same library.

To associate a device class with a library, use the LIBRARY parameter when you define a device class or update its definition.

Controlling media-mount operations for tape and optical devices

Using device class definitions, you can control the number of mounted volumes, the amount of time a volume remains mounted, and the amount of time that the Tivoli Storage Manager server waits for a drive to become available.

Controlling the number of simultaneously mounted volumes

When setting a mount limit for a device class, you need to consider the number of storage devices connected to your system, whether you are using the simultaneous-write function, whether you are associating multiple device classes with a single library, and the number of processes that you want to run at the same time.

When selecting a mount limit for a device class, consider the following issues:

- How many storage devices are connected to your system?

Do not specify a mount limit value that is greater than the number of associated available drives in your installation. If the server tries to mount as many volumes as specified by the mount limit and no drives are available for the required volume, an error occurs and client sessions may be terminated. (This does not apply when the DRIVES parameter is specified.)

- Are you using the simultaneous-write function to primary storage pools, copy storage pools, and active-data pools?

Specify a mount limit value that provides a sufficient number of mount points to support writing data simultaneously to the primary storage pool and all associated copy storage pools and active-data pools.

- Are you associating multiple device classes with a single library?

A device class associated with a library can use any drive in the library that is compatible with the device class' device type. Because you can associate more than one device class with a library, a single drive in the library can be used by more than one device class. However, Tivoli Storage Manager does not manage how a drive is shared among multiple device classes.

- How many Tivoli Storage Manager processes do you want to run at the same time, using devices in this device class?

Tivoli Storage Manager automatically cancels some processes to run other, higher priority processes. If the server is using all available drives in a device class to complete higher priority processes, lower priority processes must wait until a drive becomes available. For example, Tivoli Storage Manager cancels the process for a client backing up directly to tape if the drive being used is needed for a server migration or tape reclamation process. Tivoli Storage Manager cancels a tape reclamation process if the drive being used is needed for a client restore operation. For additional information, see “Preemption of client or server operations” on page 566.

If processes are often canceled by other processes, consider whether you can make more drives available for Tivoli Storage Manager use. Otherwise, review your scheduling of operations to reduce the contention for drives.

This consideration also applies to the simultaneous-write function. You must have enough drives available to allow for a successful simultaneous-write operation.

Best Practice: If the library associated with this device class is EXTERNAL type, explicitly specify the mount limit instead of using MOUNTLIMIT=DRIVES.

To specify the maximum number of volumes that can be simultaneously mounted, use the MOUNTLIMIT parameter when you define the device class or update its definition.

Controlling the amount of time that a volume remains mounted

You can control the amount of time that a mounted volume remains mounted after its last I/O activity. If a volume is used frequently, you can improve performance by setting a longer mount retention period to avoid unnecessary mount and dismount operations.

If mount operations are being handled by manual, operator-assisted activities, you might want to specify a long mount retention period. For example, if only one operator supports your entire operation on a weekend, then define a long mount retention period so that the operator is not being asked to mount volumes every few minutes.

To control the amount of time a mounted volume remains mounted, use the MOUNTRETENTION parameter when you define the device class or update its definition. For example, if the mount retention value is 60, and a mounted volume remains idle for 60 minutes, then the server dismounts the volume.

While Tivoli Storage Manager has a volume mounted, the drive is allocated to Tivoli Storage Manager and cannot be used for anything else. If you need to free the drive for other uses, you can cancel Tivoli Storage Manager operations that are using the drive and then dismount the volume. For example, you can cancel server migration or backup operations. For information on how to cancel processes and dismount volumes, see:

- “Canceling server processes” on page 565
- “Dismounting idle volumes” on page 154

Controlling the amount of time that the server waits for a drive

You can specify the maximum amount of time, in minutes, that the Tivoli Storage Manager server waits for a drive to become available for the current mount request.

To control wait time, use the MOUNTWAIT parameter when you define the device class or update its definition.

This parameter is not valid for EXTERNAL library types.

Write-once, read-many devices

The WORM parameter specifies whether the drive being defined is a write-once, read-many WORM device. This parameter is not supported for all device classes. You cannot change the value of the WORM parameter using the UPDATE DEVCLASS command.

For an example that shows how to configure a VolSafe device using the WORM parameter, see “Defining device classes for StorageTek VolSafe devices” on page 215

Defining 3592 device classes

Device class definitions for 3592 devices include parameters for faster volume-access speeds and drive encryption. Particular methods are required to prevent or minimize problems when mixing different generations of 3592 drives in a library.

Mixing generations of 3592 media in a single library

For optimal performance, do not mix generations of 3592 media in a single library. Media problems can result when different drive generations are mixed. For example, Tivoli Storage Manager might not be able to read a volume’s label.

The following table shows read-and-write interoperability for the three generations.

Drives	Generation 1 format	Generation 2 format	Generation 3 format
Generation 1	Read and write	n/a	n/a
Generation 2	Read and write	Read and write	n/a
Generation 3	Read only	Read and write	Read and write

If you must mix generations of drives, use one of the following methods in the following table to prevent or minimize the potential for problems.

Mixing generations of drives
<p>(349X, ACSLS, and SCSI libraries) Force all 3592 generation 3 drives to always write in the generation 2 density. Do this by explicitly setting the FORMAT parameter on the device class to either 3592-2 or 3592-2C.</p> <p>Both generation 2 and generation 3 drives can read media written in the generation 2 format. All drives can verify labels and read all data written on the media. However, this configuration does not allow the generation 3 drives to write or read in their optimal format.</p> <p>Generation 3 drives can read generation 1 format, but cannot write with it. So, mark all media previously written in generation 1 format to read-only. Generation 3 drives can both read and write with generation 2 formats.</p>

Mixing generations of drives

(349X and ACSLS libraries only) Logically partition the generations without partitioning the hardware. Define two or three new library objects for each drive generation that the physical library contains. For example, if you have a physical library with 3592-2 drives and 3592-3 drives, define two new library objects.

Specify a path with the same special file name for each new library object. In addition, for 349X libraries, specify disjoint scratch categories (including the WORMSCRATCH category, if applicable) for each library object. Specify a new device class and a new storage pool that points to each new library object.

(SCSI libraries only) Define a new storage pool and device class for the generation 3 drives. Set the FORMAT parameter to 3592-3 or 3592-3C. (Do not specify DRIVE.) The original device class will have a FORMAT parameter set to 3592, 3592C, 3952-2, or 3592-2C (not DRIVE). Update the MAXSCRATCH parameter to 0 for the storage pool that will contain all the media written in generation 1 or generation 2 formats, for example: UPDATE STGPOOL UPDATE STGPOOL *genpool1* MAXSCRATCH=0.

This method allows both generations to use their optimal format and minimizes potential media problems that can result from mixing generations. However, it does not resolve all media issues. For example, competition for mount points and mount failures might result. (To learn more about mount point competition in the context of LTO drives and media, see “Defining LTO device classes” on page 211.) The following list describes media restrictions:

- CHECKIN LIBVOL: The issue resides with using the CHECKLABEL=YES option. If the label is currently written in a generation 3 format, and you specify the CHECKLABEL=YES option, drives of previous generations fail using this command. As a best practice, use CHECKLABEL=BARCODE.
- LABEL LIBVOL: When the server tries to use drives of a previous generation to read the label written in a generation 3 format, the LABEL LIBVOL command fails unless OVERWRITE=YES is specified. Verify that the media being labeled with OVERWRITE=YES does not have any active data.
- CHECKOUT LIBVOL: When Tivoli Storage Manager verifies the label (CHECKLABEL=YES), as a generation 3 format, and read operations by drives of previous generations, the command fails. As a best practice, use CHECKLABEL=NO.

Controlling data-access speeds for 3592 volumes

Tivoli Storage Manager lets you reduce media capacity to create volumes with faster data-access speeds. The benefit is that can partition data into storage pools that have volumes with faster data-access speeds.

To reduce media capacity, use the SCALECAPACITY parameter when you define the device class or update its definition.

Specify a percentage value of 20, 90 or 100. A value of 20 percent provides the fastest access time, and 100 percent provides the largest storage capacity. For example, If you specify a scale capacity of 20 for a 3592 device class without compression, a 3592 volume in that device class would store 20 percent of its full capacity of 300 GB, or about 60 GB.

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

For information about setting up storage pool hierarchies, see “Setting up a storage pool hierarchy” on page 241.

Encrypting data with 3592 generation 2 and generation 3 drives

With Tivoli Storage Manager, you can use the following types of drive encryption with 3592 generation 2 and generation 3 drives: Application, System, and Library. These methods are defined through the hardware.

Enabling 3592 drive encryption:

The `DRIVEENCRYPTION` parameter specifies whether drive encryption is enabled or can be enabled for 3592 generation 2 (3592-2 and 3592-C) formats and 3592 generation 3 formats (3592-3 and 3592-3C). Use this parameter to ensure Tivoli Storage Manager compatibility with hardware encryption settings for empty volumes.

- To use the Application method, in which Tivoli Storage Manager generates and manages encryption keys, set the `DRIVEENCRYPTION` parameter to `ON`. This permits the encryption of data for empty volumes. If the parameter is set to `ON` and if the hardware is configured for another encryption method, backup operations will fail.
- To use the Library or System methods of encryption, set the parameter to `ALLOW`. This specifies that Tivoli Storage Manager is not the key manager for drive encryption, but will allow the hardware to encrypt the volume's data through one of the other methods. Specifying this parameter does not automatically encrypt volumes. Data can only be encrypted by specifying the `ALLOW` parameter and configuring the hardware to use one of these methods.

The following simplified example shows how to permit the encryption of data for empty volumes in a storage pool, using Tivoli Storage Manager as the key manager:

1. Define a library. For example:
`define library 3584 libtype=SCSI`
2. Define a device class, `3592_ENCRYPT`, and specify the value `ON` for the `DRIVEENCRYPTION` parameter. For example:
`define devclass 3592_encrypt library=3584 devtype=3592 driveencryption=on`
3. Define a storage pool. For example:
`define stgpool 3592_encrypt_pool 3592_encrypt`

The `DRIVEENCRYPTION` parameter is optional. The default value is to allow the Library or System methods of encryption.

For more information about using drive encryption, refer to “Encrypting data on tape” on page 496.

Disabling 3592 drive encryption:

To disable any method of encryption on new volumes, set the `DRIVEENCRYPTION` parameter to `OFF`. If the hardware is configured to encrypt data through either the Library or System method and `DRIVEENCRYPTION` is set to `OFF`, backup operations will fail.

Defining device classes for removable media devices

To access volumes that belong to this device class, the server requests that the removable media be mounted in drives. The server then opens a file on the media and reads or writes the file data.

Removable file devices include:

Iomega Zip drives, Iomega Jaz drives, and CD drives

To define a device class for removable media, use the `DEVTYPE=REMOVABLEFILE` parameter in the device class definition.

Tivoli Storage Manager REMOVABLEFILE device class supports only single-sided media. Therefore, if a data cartridge that is associated with a REMOVABLEFILE device class has two sides, the Tivoli Storage Manager server treats each side as a separate Tivoli Storage Manager volume.

When using CD-ROM media for the REMOVABLEFILE device type, the library type must be specified as `MANUAL`. Access this media through a mount point, for example, `/mnt/cdrom`.

For more information, see:

“Removable file device configuration” on page 118

Defining sequential-access disk (FILE) device classes

FILE device classes are used for storing data on disk in *simulated* storage volumes. The storage volumes are actually files. Data is written sequentially into the file system of the server machine. Because each volume in a FILE device class is actually a file, a volume name must be a fully qualified file name.

To define a FILE device class, use the `DEVTYPE=FILE` parameter in the device class definition.

Concurrent access to FILE volumes

Concurrent access improves restore performance by allowing two or more clients to access the same volume at the same time.

The Tivoli Storage Manager server allows multiple client sessions (archive, retrieve, backup, and restore) or server processes, for example, storage pool backup, to concurrently read a volume in a storage pool associated with a FILE-type device class. In addition, one client session can write to the volume while it is being read.

The following server processes are allowed shared read access to FILE volumes:

- BACKUP DB
- BACKUP STGPOOL
- COPY ACTIVATEDATA
- EXPORT/IMPORT NODE
- EXPORT/IMPORT SERVER
- GENERATE BACKUPSET
- RESTORE STGPOOL
- RESTORE VOLUME

The following server processes are not allowed shared read access to FILE volumes:

- AUDIT VOLUME
- DELETE VOLUME
- MIGRATION
- MOVE DATA
- MOVE NODEDATA
- RECLAMATION

Mitigating performance degradation when backing up or archiving to FILE volumes

The minimum I/O to a volume associated with a FILE device class is 256 KB, regardless how much data is being written to the volume. For example, if you are backing up one 500-byte object, it takes 256 KB of I/O to store it on the volume. The size of the I/O for a volume associated with a FILE device class has the greatest impact when backing up or archiving a large number of small objects, for example, small files or small directories.

To reduce the potential for performance degradation, increase the size of aggregates created by the server. (An aggregate is an object that contains multiple logical files that are backed up or archived from a client in a single transaction.) To increase the size of aggregates, do one of the following

- Increase the value of the TXNGROUPMAX option in the server options file (dsmserv.opt).
- Increase the value of the TXNGROUPMAX parameter on the REGISTER NODE or UPDATE NODE server commands.

In addition to increasing the TXNGROUPMAX value, you might also need to increase the values for the following options:

- The client option TXNBYTELIMIT in the client options file (dsm.opt)
- The server options MOVEBATCHSIZE and MOVESIZETHRESH

For details about the client option TXNBYTELIMIT, refer to the *Backup-Archive Clients Installation and User's Guide*. For details about server commands and options, refer to the *Administrator's Reference*.

Specifying directories in FILE device-class definitions

The directory name in a FILE device-class definition identifies the location where the server places the files that represent storage volumes for the device class. When processing the DEFINE DEVCLASS command, the server expands the specified directory name into its fully qualified form, starting from the root directory.

You can specify one or more directories as the location of the files used in the FILE device class. The default is the current working directory of the server at the time the command is issued.

Attention: Do not specify multiple directories from the same file system. Doing so can cause incorrect space calculations. For example, if the directories /usr/dir1 and /usr/dir2 are in the same file system, the space check, which does a preliminary evaluation of available space during store operations, will count each directory as a separate file system. If space calculations are incorrect, the server could commit to a FILE storage pool, but not be able to obtain space, causing the operation to fail. If the space check is accurate, the server can skip the FILE pool in the storage hierarchy and use the next storage pool if one is available.

If the server needs to allocate a scratch volume, it creates a new file in the specified directory or directories. (The server can choose any of the directories in which to create new scratch volumes.) To optimize performance, ensure that multiple directories correspond to separate physical volumes.

The following table lists the file name extension created by the server for scratch volumes depending on the type of data that is stored.

For scratch volumes used to store this data:	The file extension is:
Client data	.BFS
Export	.EXP
Database backup	.DBV

Avoiding data-integrity problems when using disk subsystems and file systems

Tivoli Storage Manager supports the use of remote file systems or drives for reading and writing storage pool data, database backups, and other data operations. Disk subsystems and file systems must not report successful write operations when they can fail after a successful write report to Tivoli Storage Manager.

A write failure after a successful notification constitutes a data-integrity problem because the data that was reported as successfully written is unavailable for retrieval. In this situation, all data subsequently written is also at risk due to positioning mismatches within the target file. To avoid these problems, ensure that disk subsystems and file systems, whatever implementation you use, are *always* able to return data when the data is requested.

For important disk-related information, see “Requirements for disk subsystems” on page 69.

Giving storage agents access to 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 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 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.

If file volume /opt/tivoli1/file1.dsm is created on SERVER1, and if the following command is issued,

```
update devclass classa directory="/opt/otherdir,/opt/tivoli2,
/opt/tivoli3"
```

SERVER1 will still be able to access file volume /opt/tivoli1/file1.dsm, but the storage agent STA1 will not be able to access it because a matching directory name in the PATH directory list no longer exists. If a directory name is not available in the directory list associated with the device class, the storage agent can lose access to a FILE volume in that directory. Although the volume will still be accessible from the Tivoli Storage Manager server for reading, failure of the storage agent to access the FILE volume can cause operations to be retried on a LAN-only path or to fail.

Controlling the size of FILE volumes

You can specify a maximum capacity value that controls the size of volumes (that is, files) associated with a FILE device class.

To restrict the size of volumes, use the MAXCAPACITY parameter when you define a device class or update its definition. When the server detects that a volume has reached a size equal to the maximum capacity, it treats the volume as full and stores any new data on a different volume.

Controlling the number of concurrently open FILE volumes

Tivoli Storage Manager lets you restrict the number of mount points (volumes or files) that can be concurrently opened for access by server storage and retrieval operations. Attempts to access more volumes than the number indicated causes the requester to wait.

When selecting a mount limit for this device class, consider how many Tivoli Storage Manager processes you want to run at the same time.

Tivoli Storage Manager automatically cancels some processes to run other, higher priority processes. If the server is using all available mount points in a device class to complete higher priority processes, lower priority processes must wait until a mount point becomes available. For example, Tivoli Storage Manager cancels the process for a client backup if the mount point being used is needed for a server migration or reclamation process. Tivoli Storage Manager cancels a reclamation process if the mount point being used is needed for a client restore operation. For additional information, see "Preemption of client or server operations" on page 566.

If processes are often canceled by other processes, consider whether you can make more mount points available for Tivoli Storage Manager use. Otherwise, review your scheduling of operations to reduce the contention for resources.

To specify the number of concurrently opened mount points, use the MOUNTLIMIT parameter when you define the device class or update its definition.

Defining LTO device classes

Special consideration is required to prevent or minimize problems when mixing different generations of LTO drives and media in a single library. LTO drive encryption might also be a consideration.

Mixing LTO drives and media in a library

When mixing different generations of LTO drives and media, you need to consider the read-write capabilities of each generation. As a best practice, configure a different device class for each generation of media.

If you are considering mixing different generations of LTO media and drives, be aware of the following restrictions:

Table 22. 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 1	Read and write	n/a	n/a	n/a	n/a
Generation 2	Read and write	Read and write	n/a	n/a	n/a
Generation 3	Read only	Read and write	Read and write	n/a	n/a
Generation 4	n/a	Read only	Read and write	Read and write	Read and write
Generation 5	n/a	n/a	Read only	Read and write	Read and write

If you are mixing different types of drives and media, configure different device classes: one for each type of media. To specify the exact media type, use the FORMAT parameter in each of the device class definitions. (Do not specify FORMAT=DRIVE). For example, if you are mixing Ultrium Generation 1 and Ultrium Generation 2 drives, specify FORMAT=ULTRIUMC (or ULTRIUM) for the Ultrium Generation 1 device class, and FORMAT=ULTRIUM2C (or ULTRIUM2) for the Ultrium Generation 2 device class.

Both device classes can point to the same library in which there can be Ultrium Generation 1 and Ultrium Generation 2 drives. The drives will be shared between the two storage pools. One storage pool will use the first device class and Ultrium Generation 1 media exclusively. The other storage pool will use the second device class and Ultrium Generation 2 media exclusively. Because the two storage pools share a single library, Ultrium Generation 1 media can be mounted on Ultrium Generation 2 drives as they become available during mount point processing.

Remember:

- If you are mixing Ultrium Generation 1 with Ultrium Generation 3 drives and media in a single library, you must mark the Generation 1 media as read-only, and all Generation 1 scratch volumes must be checked out.
- If you are mixing Ultrium Generation 2 with Ultrium Generation 4 or Generation 5 drives and media in a single library, you must mark the Generation 2 media as read-only, and all Generation 2 scratch volumes must be checked out.

Mount limits in LTO mixed-media environments

In a mixed-media library, in which multiple device classes point to the same library, compatible drives are shared between storage pools. You must pay special attention to setting an appropriate value for the MOUNTLIMIT parameter in each of the device classes. In a mixed media library containing Ultrium Generation 1 and Ultrium Generation 2 drives and media, for example, Ultrium Generation 1 media can get mounted in Ultrium Generation 2 drives.

Consider the example of a mixed library: that consists of the following drives and media:

- Four LTO Ultrium Generation 1 drives and LTO Ultrium Generation 1 media
- Four LTO Ultrium Generation 2 drives and LTO Ultrium Generation 2 media

You created the following device classes:

- LTO Ultrium Generation 1 device class LTO1CLASS specifying FORMAT=ULTRIUMC
- LTO Ultrium Generation 2 device class LTO2CLASS specifying FORMAT=ULTRIUM2C

You also created the following storage pools:

- LTO Ultrium Generation 1 storage pool LTO1POOL based on device class LTO1CLASS
- LTO Ultrium Generation 2 storage pool LTO2POOL based on device class LTO2CLASS

The number of mount points available for use by each storage pool is specified in the device class using the MOUNTLIMIT parameter. The MOUNTLIMIT parameter in the LTO2CLASS device class should be set to 4 to match the number of available drives that can mount only LTO2 media. The MOUNTLIMIT parameter in the LTO1CLASS device class should be set to a value higher (5 or possibly 6) than the number of available drives to adjust for the fact that Ultrium Generation 1 media can be mounted in Ultrium Generation 2 drives. The optimum value for MOUNTLIMIT will depend on workload and storage pool access patterns.

Monitor and adjust the MOUNTLIMIT setting to suit changing workloads. If the MOUNTLIMIT for LTO1POOL is set too high, mount requests for the LTO2POOL might be delayed or fail because the Ultrium Generation 2 drives have been used to satisfy Ultrium Generation 1 mount requests. In the worst scenario, too much competition for Ultrium Generation 2 drives might cause mounts for Generation 2 media to fail with the following message:

```
ANR8447E No drives are currently available in the library.
```

If the MOUNTLIMIT for LTO1POOL is not set high enough, mount requests that could potentially be satisfied LTO Ultrium Generation 2 drives will be delayed.

Some restrictions apply when mixing Ultrium Generation 1 with Ultrium Generation 2 or Generation 3 drives because of the way in which mount points are

allocated. For example, processes that require multiple mount points that include both Ultrium Generation 1 and Ultrium Generation 2 volumes might try to reserve Ultrium Generation 2 drives only, even when one mount can be satisfied by an available Ultrium Generation 1 drive. Processes that behave in this manner include the MOVE DATA and BACKUP STGPOOL commands. These processes will wait until the needed number of mount points can be satisfied with Ultrium Generation 2 drives.

Encrypting data using LTO generation 4 tape drives

Tivoli Storage Manager supports the three types of drive encryption available with LTO generation 4 drives: Application, System, and Library. These methods are defined through the hardware.

For more information about using drive encryption, refer to “Encrypting data on tape” on page 496.

Enabling LTO drive encryption

The DRIVEENCRYPTION parameter specifies whether drive encryption is enabled or can be enabled for IBM and HP LTO generation 4, Ultrium4, and Ultrium4C formats. This parameter ensures Tivoli Storage Manager compatibility with hardware encryption settings for empty volumes.

Tivoli Storage Manager supports the Application method of encryption with IBM and HP LTO-4 drives. Only IBM LTO-4 supports the System and Library methods. The Library method of encryption is supported only if your system hardware (for example, IBM 3584) supports it.

Remember: You cannot use drive encryption with write-once, read-many (WORM) media.

The Application method is defined through the hardware. To use the Application method, in which Tivoli Storage Manager generates and manages encryption keys, set the DRIVEENCRYPTION parameter to ON. This permits the encryption of data for empty volumes. If the parameter is set to ON and the hardware is configured for another encryption method, backup operations will fail.

The following simplified example shows the steps you would take to permit the encryption of data for empty volumes in a storage pool:

1. Define a library:
`define library 3584 libtype=SCSI`
2. Define a device class, LTO_ENCRYPT, and specify Tivoli Storage Manager as the key manager:
`define devclass lto_encrypt library=3584 devtype=lto driveencryption=on`
3. Define a storage pool:
`define stgpool lto_encrypt_pool lto_encrypt`

Disabling LTO drive encryption

To disable encryption on new volumes, set the `DRIVEENCRYPTION` parameter to `OFF`. The default value is `ALLOW`. Drive encryption for empty volumes is permitted if another method of encryption is enabled.

Defining SERVER device classes

`SERVER` device classes let you create volumes for one Tivoli Storage Manager server that exist as archived files in the storage hierarchy of another server, called a target server. These virtual volumes have the characteristics of sequential-access volumes such as tape.

To define a `SERVER` device class, use the `DEFINE DEVCLASS` command with the `DEVTYPE=SERVER` parameter. For information about how to use a `SERVER` device class, see “Using virtual volumes to store data on another server” on page 712.

Controlling the size of files created on a target server

You can specify a maximum capacity value that controls the size of files that are created on the target server to store data for the source server.

To specify a file size, use the `MAXCAPACITY` parameter when you define the device class or update its definition.

The storage pool volumes of this device type are explicitly set to full when the volume is closed and dismounted.

Controlling the number of simultaneous sessions between source and target servers

You can control the number of simultaneous sessions between the source server and the target server. Any attempts to access more sessions than indicated by the mount limit causes the requester to wait.

To control the number of simultaneous sessions, use the `MOUNTLIMIT` parameter when you define the device class or update its definition.

When specifying a mount limit, consider your network load balancing and how many Tivoli Storage Manager processes you want to run at the same time.

Tivoli Storage Manager automatically cancels some processes to run other, higher priority processes. If the server is using all available sessions in a device class to complete higher priority processes, lower priority processes must wait until a session becomes available. For example, Tivoli Storage Manager cancels the process for a client backup if a session is needed for a server migration or reclamation process. Tivoli Storage Manager cancels a reclamation process if the session being used is needed for a client restore operation.

When specifying a mount limit, also consider the resources available on the target server when setting mount limits. Do not set a high mount limit value if the target cannot move enough data or access enough data to satisfy all of the requests.

If processes are often canceled by other processes, consider whether you can make more sessions available for Tivoli Storage Manager use. Otherwise, review your scheduling of operations to reduce the contention for network resources.

Controlling the amount of time a SERVER volume remains mounted

You can improve response time for SERVER media mounts by leaving previously mounted volumes online.

To specify the amount of time, in minutes, to retain an idle sequential access volume before dismounting it, use the MOUNTRETENTION parameter when you define the device class or update its definition.

A value of 1 to 5 minutes is recommended.

Defining device classes for StorageTek VolSafe devices

StorageTek VolSafe brand Ultrium drives use media that cannot be overwritten. Do not use this media for short-term backups of client files, the server database, or export tapes.

There are two methods for using VolSafe media and drives: This technology uses media that cannot be overwritten; therefore, do not use this media for short-term backups of client files, the server database, or export tapes.

- Define a device class using the DEFINE DEVCLASS command and specify DEVTYPE=VOLSAFE. You can use this device class with EXTERNAL, SCSI, and ACSLS libraries. All drives in a library must be enabled for VolSafe use.
- Define a device class using the DEFINE DEVCLASS command, and specify DEVTYPE=ECARTRIDGE and WORM=YES. For VolSafe devices, WORM=YES is required and must be specified when the device class is defined. You cannot update the WORM parameter using the UPDATE DEVCLASS command. You cannot specify DRIVEENCRYPTION=ON if your drives are using WORM media.

To enable the VolSafe function, consult your StorageTek hardware documentation. Attempting to write to VolSafe media without a VolSafe-enabled drive results in errors.

To configure a VolSafe device in a SCSI library using the DEVTYPE-ECARTRIDGE parameter, enter the following series of commands. (The values you select for the library variable, the drive variable, and so on might be different for your environment.)

1. Define a library:

```
define library volsafelib libtype=scsi
```
2. Define a drive:

```
define drive volsafelib drive01
```
3. Define a path:

```
define path server01 drive01 srctype=server destype=drive  
device=/dev/tsm SCSI/mt0 library=volsafelib
```
4. Define a device class:

```
define devclass volsafeclass library=volsafelib devtype=ecartridge  
format=drive worm=yes
```

For more information about VolSafe media, see “Write-once, read-many tape media” on page 141.

Enabling ECARTRIDGE drive encryption

The DRIVEENCRYPTION parameter specifies whether drive encryption is enabled or can be enabled for DRIVE, T10000B, and T10000B-C formats. This parameter ensures Tivoli Storage Manager compatibility with hardware encryption settings for empty volumes.

Tivoli Storage Manager supports the Application method of encryption with Sun StorageTek T10000B drives. The Library method of encryption is supported only if your system hardware supports it.

Remember: You cannot use drive encryption with write-once, read-many (WORM) media.

The Application method, in which Tivoli Storage Manager generates and manages encryption keys, is defined through the hardware. To use the Application method, set the DRIVEENCRYPTION parameter to ON. This setting permits the encryption of data for empty volumes. If the parameter is set to ON and the hardware is configured for another encryption method, backup operations fail.

The following simplified example shows the steps you would take to permit data encryption for empty volumes in a storage pool:

1. Define a library:

```
define library sl3000 libtype=scsi
```
2. Define a device class, ECART_ENCRYPT, and specify Tivoli Storage Manager as the key manager:

```
define devclass ecart_encrypt library=sl3000  
devtype=ecartridge driveencryption=on
```
3. Define a storage pool:

```
define stgpool ecart_encrypt_pool ecart_encrypt
```

Related concepts

“Choosing an encryption method” on page 497

Disabling ECARTRIDGE drive encryption

To disable encryption on new volumes, set the DRIVEENCRYPTION parameter to OFF. The default value is ALLOW. You can use drive encryption for empty volumes if another method of encryption is enabled.

Obtaining information about device classes

You can choose to view a standard or detailed report for a device class.

Task	Required Privilege Class
Request information about device classes	Any administrator

To display a standard report on device classes, enter:
query devclass

Figure 16 on page 217 provides an example of command output.

Device Class Name	Device Access Strategy	Storage Pool Count	Device Type	Format	Est/Max Capacity (MB)	Mount Limit
DISK	Random	9				
TAPE8MM	Sequential	1	8MM	8200		2
FILE	Sequential	1	FILE	DRIVE	5,000.0	1
GEN1	Sequential	2	LTO	ULTRIUM		DRIVES

Figure 16. Example of a standard device class report

To display a detailed report on the GEN1 device class, enter:
`query devclass gen1 format=detailed`

Figure 17 provides an example of command output.

```

Device Class Name: GEN1
Device Access Strategy: Sequential
Storage Pool Count: 2
Device Type: LTO
Format: ULTRIUM
Est/Max Capacity (MB):
Mount Limit: DRIVES
Mount Wait (min): 60
Mount Retention (min): 60
Label Prefix: ADSM
Drive Letter:
Library: GEN2LIB
Directory:
Server Name:
Retry Period:
Retry Interval:
TwoSided:
Shared:
High-level Address:
Minimum Capacity:
WORM:
Scaled Capacity:
Last Update by (administrator): ADMIN
Last Update Date/Time: 01/23/03 12:25:31

```

Figure 17. Example of a detailed device class report

How Tivoli Storage Manager fills volumes

The DEFINE DEVCLASS command has an optional ESTCAPACITY parameter that indicates the estimated capacity for sequential volumes associated with the device class. Tivoli Storage Manager uses the estimated capacity of volumes to determine the estimated capacity of a storage pool, and the estimated percent utilized.

If the ESTCAPACITY parameter is not specified, Tivoli Storage Manager uses a default value based on the recording format specified for the device class (FORMAT=).

If you specify an estimated capacity that exceeds the actual capacity of the volume in the device class, Tivoli Storage Manager updates the estimated capacity of the volume when the volume becomes full. When Tivoli Storage Manager reaches the end of the volume, it updates the capacity for the amount that is written to the volume.

You can either accept the default estimated capacity for a given device class, or explicitly specify an estimated capacity. An accurate estimated capacity value is not required, but is useful. Tivoli Storage Manager uses the estimated capacity of

volumes to determine the estimated capacity of a storage pool, and the estimated percent utilized. You may want to change the estimated capacity if:

- The default estimated capacity is inaccurate because data compression is being performed by the drives.
- You have volumes of nonstandard size.

Data compression

Client files can be compressed to decrease the amount of data sent over networks and the space occupied by the data in Tivoli Storage Manager storage. With Tivoli Storage Manager, files can be compressed by the Tivoli Storage Manager client before the data is sent to the Tivoli Storage Manager server, or by the device where the file is finally stored.

Use either client compression or device compression, but not both. The following table summarizes the advantages and disadvantages of each type of compression.

Type of Compression	Advantages	Disadvantages
Tivoli Storage Manager client compression	Reduced load on the network	Higher CPU usage by the client Longer elapsed time for client operations such as backup
Drive compression	Amount of compression can be better than Tivoli Storage Manager client compression on some drives	Using drive compression on files that have already been compressed by the Tivoli Storage Manager client can increase file size

Either type of compression can affect tape drive performance, because compression affects data rate. When the rate of data going to a tape drive is slower than the drive can write, the drive starts and stops while data is written, meaning relatively poorer performance. When the rate of data is fast enough, the tape drive can reach streaming mode, meaning better performance. If tape drive performance is more important than the space savings that compression can mean, you may want to perform timed test backups using different approaches to determine what is best for your system.

Drive compression is specified with the `FORMAT` parameter for the drive's device class, and the hardware device must be able to support the compression format. For information about how to set up compression on the client, see "Node compression considerations" on page 370 and "Registering nodes with the server" on page 368.

Tape volume capacity and data compression

How Tivoli Storage Manager views the capacity of the volume where the data is stored depends on whether files are compressed by the Tivoli Storage Manager client or by the storage device.

It may wrongly appear that you are not getting the full use of the capacity of your tapes, for the following reasons:

- A tape device manufacturer often reports the capacity of a tape based on an assumption of compression by the device. If a client compresses a file before it is sent, the device may not be able to compress it any further before storing it.

- Tivoli Storage Manager records the size of a file as it goes to a storage pool. If the client compresses the file, Tivoli Storage Manager records this smaller size in the database. If the drive compresses the file, Tivoli Storage Manager is not aware of this compression.

Figure 18 compares what Tivoli Storage Manager sees as the amount of data stored on tape when compression is done by the device and by the client. For this example, the tape has a physical capacity of 1.2 GB. However, the manufacturer reports the capacity of the tape as 2.4 GB by assuming the device compresses the data by a factor of two.

Suppose a client backs up a 2.4 GB file:

- When the client does *not* compress the file, the server records the file size as 2.4 GB, the file is compressed by the drive to 1.2 GB, and the file fills up one tape.
- When the client compresses the file, the server records the file size as 1.2 GB, the file cannot be compressed any further by the drive, and the file still fills one tape.

In both cases, Tivoli Storage Manager considers the volume to be full. However, Tivoli Storage Manager considers the capacity of the volume in the two cases to be different: 2.4 GB when the drive compresses the file, and 1.2 GB when the client compresses the file. Use the `QUERY VOLUME` command to see the capacity of volumes from Tivoli Storage Manager's viewpoint. See “Monitoring the use of storage pool volumes” on page 335.

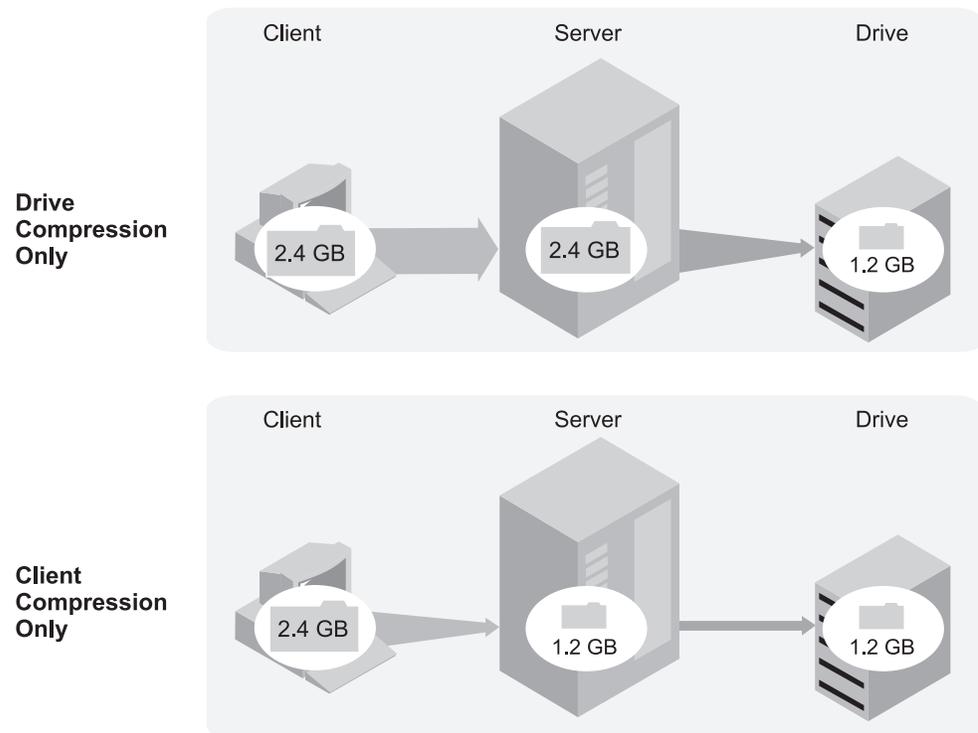


Figure 18. Comparing compression at the client and compression at the device

For how to set up compression on the client, see “Node compression considerations” on page 370 and “Registering nodes with the server” on page 368.

Chapter 10. Managing storage pools and volumes

Logical storage pools and storage volumes are the principal components in the Tivoli Storage Manager model of data storage. By manipulating the properties of these objects, you can optimize the usage of storage devices.

When you configure devices so that the server can use them to store client data, you create storage pools and storage volumes. The procedures for configuring devices use the set of defaults that provides for storage pools and volumes. The defaults can work well. However, you might have specific requirements not met by the defaults. There are three common reasons to change the defaults:

- Optimize and control storage device usage by arranging the storage hierarchy and tuning migration through the hierarchy (next storage pool, migration thresholds).
- Reuse tape volumes through reclamation. Reuse is also related to policy and expiration.
- Keep a client's files on a minimal number of volumes (collocation).

You can also make other adjustments to tune the server for your systems. See the following sections to learn more. For some quick tips, see “Task tips for storage pools” on page 232.

Concepts
“Storage pools” on page 222
“Storage pool volumes” on page 234
“Access modes for storage pool volumes” on page 239
“Storage pool hierarchies” on page 241
“Migrating files in a storage pool hierarchy” on page 252
“Caching in disk storage pools” on page 263
“Writing data simultaneously to primary, copy, and active-data pools” on page 285
“Keeping client files together using collocation” on page 310
“Reclaiming space in sequential-access storage pools” on page 319
“Estimating space needs for storage pools” on page 330

Tasks
“Defining storage pools” on page 227
“Preparing volumes for random-access storage pools” on page 236
“Preparing volumes for sequential-access storage pools” on page 236
“Defining storage pool volumes” on page 237
“Updating storage pool volumes” on page 238
“Setting up a storage pool hierarchy” on page 241
“Monitoring storage-pool and volume usage” on page 332
“Monitoring the use of storage pool volumes” on page 335
“Moving data from one volume to another volume” on page 349
“Moving data belonging to a client node” on page 354

Tasks
"Renaming storage pools" on page 357
"Defining copy storage pools and active-data pools" on page 357
"Deleting storage pools" on page 361
"Deleting storage pool volumes" on page 361

For details about devices, see:

Chapter 4, "Magnetic disk devices," on page 69

Chapter 6, "Configuring storage devices," on page 89

The examples in topics show how to perform tasks using the Tivoli Storage Manager command-line interface. For information about the commands, see the *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

You can also perform Tivoli Storage Manager tasks from the Administration Center. For more information about using the Administration Center, see Chapter 18, "Managing servers with the Administration Center," on page 547.

Storage pools

A storage pool is a collection of storage volumes. A storage volume is the basic unit of storage, such as allocated space on a disk or a single tape cartridge. The server uses the storage volumes to store backed-up, archived, or space-managed files.

The server provides three types of storage pools that serve different purposes: primary storage pools, copy storage pools, and active-data pools. You can arrange primary storage pools in a *storage hierarchy*. The group of storage pools that you set up for the Tivoli Storage Manager server to use is called *server storage*.

Primary storage pools

When a user tries to restore, retrieve, recall, or export file data, the requested file is obtained from a primary storage pool, if possible. Primary storage pool volumes are always located on-site.

The server has three default, random-access primary storage pools:

ARCHIVEPOOL

In default STANDARD policy, the destination for files that are archived from client nodes

BACKUPPOOL

In default STANDARD policy, the destination for files that are backed up from client nodes

SPACEMGPOOL

For space-managed files that are migrated from Tivoli Storage Manager for Space Management client nodes (HSM clients)

To prevent a single point of failure, create separate storage pools for backed-up and space-managed files. This also includes not sharing a storage pool in either storage pool hierarchy. Consider setting up a separate, random-access disk storage pool to give clients fast access to their space-managed files.

Restriction: Backing up a migrated, space-managed file could result in an error if the destination for the backup is the same storage pool as the storage pool where the space-managed file currently exists.

A primary storage pool can use random-access storage (DISK device class) or sequential-access storage (for example, tape or FILE device classes).

Copy storage pools

Copy storage pools contain active and inactive versions of data that is backed up from primary storage pools. Copy storage pools provide a means of recovering from disasters or media failures.

For example, when a client attempts to retrieve a file and the server detects an error in the file copy in the primary storage pool, the server marks the file as damaged. At the next attempt to access the file, the server can obtain the file from a copy storage pool.

You can move copy storage pool volumes off-site and still have the server track the volumes. Moving copy storage pool volumes off-site provides a means of recovering from an on-site disaster.

A copy storage pool can use only sequential-access storage (for example, a tape device class or FILE device class).

Remember:

- You can back up data from a primary storage pool defined with the NATIVE, NONBLOCK, or any of the NDMP formats (NETAPPDUMP, CELERRADUMP, or NDMPDUMP). The target copy storage pool must have the same data format as the primary storage pool.

For details about copy storage pools, see:

- “Restoring storage pools” on page 777
- “Backing up storage pools” on page 760
- “Recovering a lost or damaged storage pool volume” on page 796
- “Ensuring the integrity of files” on page 791
- “Backing up the data in a storage hierarchy” on page 246
- “Setting up copy storage pools and active-data pools” on page 247
- “Backing up storage pools” on page 760

Active-data pools

An active-data pool contains only active versions of client backup data. active-data pools are useful for fast client restores, reducing the number of on-site or off-site storage volumes, or reducing bandwidth when copying or restoring files that are vaulted electronically in a remote location.

Data migrated by hierarchical storage management (HSM) clients and archive data are not permitted in active-data pools. As updated versions of backup data continue to be stored in active-data pools, older versions are deactivated and removed during reclamation processing.

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. As a

best practice and to protect your inactive data, therefore, you should create a minimum of two storage pools: one active-data pool, which contains only active data, and one copy storage pool, which contains both active and inactive data. You can use the active-data pool volumes to restore critical client node data, and afterward you can restore the primary storage pools from the copy storage pool volumes. active-data pools should not be considered for recovery of a primary pool or volume unless the loss of inactive data is acceptable.

Active-data pools can use any type of sequential-access storage (for example, a tape device class or FILE device class). However, the precise benefits of an active-data pool depend on the specific device type associated with the pool. For example, active-data pools associated with a FILE device class are ideal for fast client restores because FILE volumes do not have to be physically mounted and because the server does not have to position past inactive files that do not have to be restored. In addition, client sessions restoring from FILE volumes in an active-data pool can access the volumes concurrently, which also improves restore performance.

Active-data pools that use removable media, such as tape or optical, offer similar benefits. Although tapes need to be mounted, the server does not have to position past inactive files. However, the primary benefit of using removable media in active-data pools is the reduction of the number of volumes used for on-site and off-site storage. If you vault data electronically to a remote location, an active-data pool associated with a SERVER device class lets you save bandwidth by copying and restoring only active data.

Remember:

- The server will not attempt to retrieve client files from an active-data pool during a point-in-time restore. Point-in-time restores require both active and inactive file versions. Active-data pools contain only active file versions. For optimal efficiency during point-in-time restores and to avoid switching between active-data pools and primary or copy storage pools, the server retrieves both active and inactive versions from the same storage pool and volumes.
- You cannot copy active data to an active-data pool from a primary storage pool defined with the NETAPPDUMP, the CELERRADUMP, or the NDMPDUMP data format.

For details about active-data pools, see:

- “Backing up the data in a storage hierarchy” on page 246
- “Setting up copy storage pools and active-data pools” on page 247
- “Copying active versions of client backup data to active-data pools” on page 225
- “Active-data pools as sources of active file versions for server operations” on page 225

Copying active versions of client backup data to active-data pools

To copy active versions of client backup files from primary storage pools to active-data pools, you can issue the COPY ACTIVE DATA command or you can use the simultaneous-write function. The simultaneous-write function automatically writes active backup data to active-data pools at the same time that the backup data is written to a primary storage pool.

You can issue the COPY ACTIVE DATA command either manually or in an administrative schedule or maintenance script.

Regardless whether you use the COPY ACTIVE DATA command or the simultaneous-write function, the Tivoli Storage Manager server writes data to an active-data pool only if the data belongs to a node that is a member of a policy domain that specifies the active-data pool as the destination for active data.

Restriction: You cannot use the BACKUP STGPOOL command for active-data pools.

Active-data pools as sources of active file versions for server operations

The Tivoli Storage Manager uses a search order to locate active file versions.

During client sessions and processes that require active file versions, the Tivoli Storage Manager server searches certain types of storage pools, if they exist.

1. An active-data pool associated with a FILE device class
2. A random-access disk (DISK) storage pool
3. A primary or copy storage pool associated with a FILE device class
4. A primary, copy, or active-data pool associated with on-site or off-site removable media (tape or optical)

Even though the list implies a selection order, the server might select a volume with an active file version from a storage pool lower in the order if a volume higher in the order cannot be accessed because of the requirements of the session or process, volume availability, or contention for resources such as mount points, drives, and data.

Example: Setting up server storage

All the data in four primary storage pools is backed up to one copy storage pool. Active versions of data are stored in an active-data pool.

Figure 19 on page 226 shows one way to set up server storage. In this example, the storage defined for the server includes:

- Three disk storage pools, which are primary storage pools: ARCHIVE, BACKUP, and HSM
- One primary storage pool that consists of tape cartridges
- One copy storage pool that consists of tape cartridges
- One active-data pool that consists of FILE volumes for fast client restore

Policies defined in management classes direct the server to store files from clients in the ARCHIVE, BACKUP, or HSM disk storage pools. An additional policy specifies the following:

- A select group of client nodes that requires fast restore of active backup data

- The active-data pool as the destination for the active-data belonging to these nodes
- The ARCHIVE, BACKUP, or HSM disk storage pools as destinations for archive, backup (active and inactive versions), and space-managed data

For each of the three disk storage pools, the tape primary storage pool is next in the hierarchy. As the disk storage pools fill, the server migrates files to tape to make room for new files. Large files can go directly to tape. For more information about setting up a storage hierarchy, see “Storage pool hierarchies” on page 241.

For more information about backing up primary storage pools, see “Backing up storage pools” on page 760.

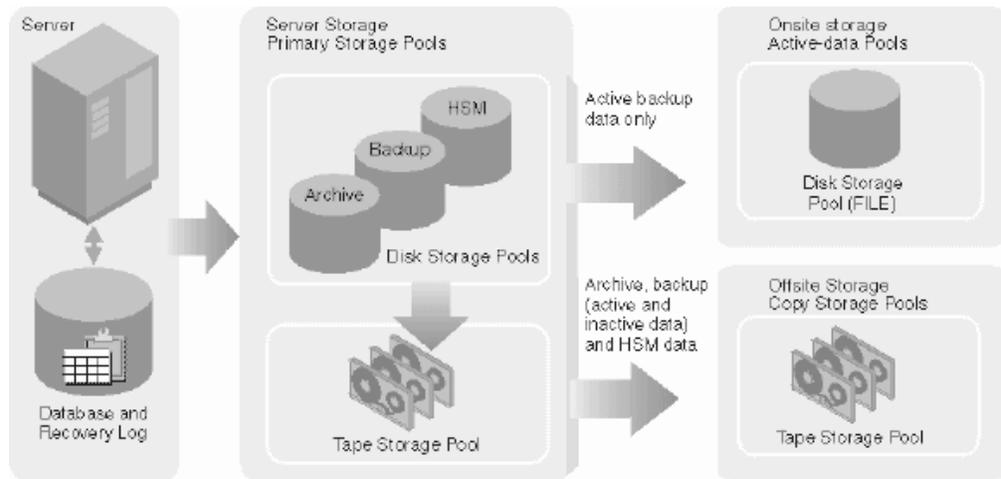


Figure 19. Example of server storage

To set up this server storage hierarchy, do the following:

1. Define the three disk storage pools, or use the three default storage pools that are defined when you install the server. Add volumes to the disk storage pools if you have not already done so.

For more information, see

“Configuring random access volumes on disk devices” on page 75

2. Define policies that direct the server to initially store files from clients in the disk storage pools. To do this, you define or change management classes and copy groups so that they point to the storage pools as destinations. Then activate the changed policy. See “Changing policy” on page 439 for details.

Define an additional policy that specifies the active-data pool that you will create as the destination for active data.

3. Assign nodes to the domains. Nodes whose active data you want to restore quickly should be assigned to the domain that specifies the active-data pool.
4. Attach one or more tape devices, or a tape library, to your server system.

To enable the server to use the device, you must enter a series of the following commands:

```
DEFINE LIBRARY
DEFINE DRIVE
DEFINE PATH
DEFINE DEVCLASS
```

DEFINE STGPOOL

For more information, see:

“Defining storage pools”

Chapter 6, “Configuring storage devices,” on page 89

5. Update the disk storage pools so that they point to the tape storage pool as the next storage pool in the hierarchy. See “Example: Updating storage pools” on page 232.
6. Define a copy storage pool and an active-data pool. The copy storage pool can use the same tape device or a different tape device as the primary tape storage pool. The active-data pool uses sequential-access disk storage (a FILE-type device class) for fast client restores. See “Defining copy storage pools and active-data pools” on page 357.
7. Set up administrative schedules or a script to back up the disk storage pools and the tape storage pool to the copy storage pool. Use the same or different schedules or scripts to copy active data to the active-data pool. Send the copy storage pool volumes off-site for safekeeping. See “Backing up storage pools” on page 760.

Defining storage pools

To optimize data storage, you can specify various properties when you define or update a storage pool using the `DEFINE STGPOOL` and `UPDATE STGPOOL` commands.

Tip: When defining or updating storage pools that use LTO Ultrium media, special considerations might apply.

Task	Required Privilege Class
Define storage pools	System
Update storage pools	System or unrestricted storage

Properties of storage pool definitions

You can define storage pools using a wide range of properties to control how data is stored. Each storage pool represents one type of media as specified in the device-class parameter.

When you define a primary storage pool, be prepared to specify some or all of the information that is shown in Table 23. Most of the information is optional. Some information applies only to random-access storage pools or only to sequential-access storage pools. Required parameters are marked.

Table 23. Information for defining a storage pool

Information	Explanation	Type of Storage Pool
Storage pool name (Required)	The name of the storage pool.	random, sequential
Device class (Required)	The name of the device class assigned for the storage pool.	random, sequential
Pool type	The type of storage pool (primary or copy). The default is to define a primary storage pool. A storage pool's type cannot be changed after it has been defined.	random, sequential

Table 23. Information for defining a storage pool (continued)

Information	Explanation	Type of Storage Pool
Maximum number of scratch volumes (Required for sequential access)	When you specify a value greater than zero, the server dynamically acquires scratch volumes when needed, up to this maximum number. For automated libraries, set this value equal to the physical capacity of the library. For details, see “Maintaining a supply of scratch volumes in an automated library” on page 150.	sequential
Access mode	Defines access to volumes in the storage pool for user operations (such as backup and restore) and system operations (such as reclamation and server migration). Possible values are: Read/Write User and system operations can read from or write to the volumes. Read-Only User operations can read from the volumes, but not write. 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. Unavailable User operations cannot get access to volumes in the storage pool. No new writes are permitted to volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool.	random, sequential
Maximum file size ¹	To exclude large files from a storage pool, set a maximum file size. The maximum file size applies to the size of a physical file (a single client file or an aggregate of client files). Do not set a maximum file size for the last storage pool in the hierarchy unless you want to exclude very large files from being stored in server storage.	random, sequential
Cyclic Redundancy Check (CRC) ¹	Specifies whether the server uses CRC to validate storage pool data during audit volume processing. For additional information see “Data validation during audit volume processing” on page 784. Using the CRC option in conjunction with scheduling audit volume processing continually ensures the integrity of data stored in your storage hierarchy. If you always want your storage pool data validated, set your primary storage pool crcdata definition to YES.	random, sequential
Name of the next storage pool ¹	Specifies the name of the next storage pool in the storage pool hierarchy, where files can be migrated or where files are stored that exceed the maximum size for this storage pool. See “Storage pool hierarchies” on page 241.	random, sequential
Migration thresholds ¹	Specifies a percentage of storage pool occupancy at which the server begins migrating files to the next storage pool (high threshold) and the percentage when migration stops (low threshold). See “Migrating files in a storage pool hierarchy” on page 252.	random, sequential
Migration processes ¹	Specifies the number of concurrent processes to use for migrating files from this storage pool. See “Migrating disk storage pools” on page 253 and “Specifying multiple concurrent migration processes” on page 261.	random, sequential

Table 23. Information for defining a storage pool (continued)

Information	Explanation	Type of Storage Pool
Migration delay ¹	Specifies the minimum number of days a file must remain in a storage pool before it is eligible for migration. See “Keeping files in a storage pool” on page 257 and “How the server migrates files from sequential-access storage pools” on page 259.	random, sequential
Continue migration process ¹	Specifies whether migration of files should continue even if files do not meet the requirement for migration delay. This setting is used only when the storage pool cannot go below the low migration threshold without moving additional files. See “Keeping files in a storage pool” on page 257 and “How the server migrates files from sequential-access storage pools” on page 259.	random, sequential
Cache	Enables or disables cache. When cache is enabled, copies of files migrated by the server to the next storage pool are left on disk after the migration. In this way, a retrieval request can be satisfied quickly. See “Caching in disk storage pools” on page 263.	random
Collocation	With collocation enabled, the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of sequential-access storage volumes. See “Keeping client files together using collocation” on page 310.	sequential
Reclamation threshold ¹	Specifies what percentage of reclaimable space can accumulate on a volume before the server initiates a space reclamation process for the volume. See “Reclamation thresholds” on page 321.	sequential
Reclamation processes ¹	Specifies the number of concurrent processes to use for reclaiming the volumes in a storage pool. See “Optimizing drive usage using multiple concurrent reclamation processes” on page 322.	sequential
Off-site reclaim limit	Specifies the number of off-site volumes to have their space reclaimed during reclamation for a storage pool. See “Reclamation of off-site volumes” on page 326.	sequential
Reclamation storage pool ¹	Specifies the name of the storage pool to be used for storing data from volumes being reclaimed in this storage pool. Use for storage pools whose device class only has one drive or mount point. See “Reclaiming volumes in a storage pool with one drive” on page 323.	sequential
Reuse delay period	Specifies the number of days that must elapse after all of the files have been deleted from a volume, before the volume can be rewritten or returned to the scratch pool. See “Delaying reuse of volumes for recovery purposes” on page 766.	sequential
Overflow location ¹	Specifies the name of a location where volumes are stored when they are ejected from an automated library by the MOVE MEDIA command. Use for a storage pool that is associated with an automated library or an external library. For details, see “Managing a full library” on page 148.	sequential
Data Format	The format in which data will be stored. NATIVE is the default data format. NETAPPDUMP and NONBLOCK are examples of other data formats.	sequential

Table 23. Information for defining a storage pool (continued)

Information	Explanation	Type of Storage Pool
Copy Storage Pools ¹	<p>Specifies the names of copy storage pools where the server simultaneously writes data when a client backup, archive, import or migration operation stores data to the primary storage pool. The server writes the data simultaneously to all listed copy storage pools. This option is restricted to primary random-access storage pools or to primary sequential-access storage pools that use the NATIVE or NONBLOCK data format. See the Copy Continue entry. See the Copy Continue entry and “Writing data simultaneously to primary, copy, and active-data pools” on page 285 for related information.</p> <p>Attention: The COPYSTGPOOLS parameter is not intended to replace the BACKUP STGPOOL command. If you use the simultaneous-write function, ensure that the copy of the primary storage pool is complete by regularly issuing the BACKUP STGPOOL command. Failure to do so could result in the inability to recover the primary storage pool data if the primary storage pool becomes damaged or lost.</p>	random, sequential
Copy Continue ¹	<p>Specifies how the server should react to a copy storage pool write failure for any of the copy storage pools listed in the COPYSTGPOOLS parameter. With a value of YES, during a write failure, the server will exclude the failing copy storage pool from any further writes while that specific client session is active. With a value of NO, during a write failure, the server will fail the entire transaction including the write to the primary storage pool.</p> <p>This option has no effect on active-data pools.</p>	sequential
Active-data pools ¹	<p>Specifies the names of active-data pools where the server simultaneously writes active versions of client node data during backups. The server writes the data simultaneously to all listed active-data pools. This option is restricted to primary random-access storage pools or to primary sequential-access storage pools that use the NATIVE or NONBLOCK data format. Nodes whose data is to be written to an active-data pool during a simultaneous-write operation must be members of a policy domain that specifies the active-data pool as the destination for active backup data.</p> <p>Attention: The ACTIVEDATAPOOLS parameter is not intended to replace the COPY ACTIVEBACKUP command. If you use the simultaneous-write function, ensure that the copy of active backup data is complete by regularly issuing the COPY ACTIVEBACKUP command. If you do not issue the COPY ACTIVEBACKUP command regularly and you do not have copy storage pools, you might not be able to recover any of the data in a primary storage pool if the primary storage pool becomes damaged or lost.</p>	random, sequential
Shredding	<p>Specifies whether data is physically overwritten when it is deleted. After client data is deleted, it might still be possible to recover it. For sensitive data, this condition is a potential security exposure. Shredding the deleted data increases the difficulty of discovering and reconstructing the data later. For more information, including how to set up shred pools and how shredding interacts with other command parameters, see “Securing sensitive client data” on page 499.</p>	random

¹ This information is not available for sequential-access storage pools that use the following data formats:

- CELERRADUMP
- NDMPDUMP
- NETAPPDUMP

Example: Defining storage pools

An engineering department requires a separate storage hierarchy. You want the department's backed-up files to go to a disk storage pool. When that pool fills, you want the files to migrate to a tape storage pool.

You want the storage pools to have the following characteristics:

- Disk primary storage pool
 - The pool named ENGBACK1 is the storage pool for the engineering department.
 - The size of the largest file that can be stored is five MB. Files larger than five MB are stored in the tape storage pool.
 - Files migrate from the disk storage pool to the tape storage pool when the disk storage pool is 85% full. File migration to the tape storage pool stops when the disk storage pool is down to 40% full.
 - The access mode is the default, read/write.
 - Cache is used.
- Tape primary storage pool
 - The name of the pool is BACKTAPE.
 - The pool uses the device class TAPE, which has already been defined.
 - No limit is set for the maximum file size, because this is the last storage pool in the hierarchy.
 - To group files from the same client on a small number of volumes, use collocation at the client node level.
 - Use scratch volumes for this pool, with a maximum number of 100 volumes.
 - The access mode is the default, read/write.
 - Use the default for reclamation: Reclaim a partially full volume (to allow tape reuse) when 60% of the volume's space can be reclaimed.

You can define the storage pools in a storage pool hierarchy from the top down or from the bottom up. Defining the hierarchy from the bottom up requires fewer steps. To define the hierarchy from the bottom up, perform the following steps:

1. Define the storage pool named BACKTAPE with the following command:

```
define stgpool backtape tape
description='tape storage pool for engineering backups'
maxsize=nolimit collocate=node maxscratch=100
```

2. Define the storage pool named ENGBACK1 with the following command:

```
define stgpool engback1 disk
description='disk storage pool for engineering backups'
maxsize=5m nextstgpool=backtape highmig=85 lowmig=40
```

Restrictions:

- You cannot establish a chain of storage pools that lead to an endless loop. For example, you cannot define StorageB as the *next* storage pool for StorageA, and then define StorageA as the *next* storage pool for StorageB.
- The storage pool hierarchy includes only primary storage pools, not copy storage pools or active-data pools.
- If a storage pool uses the data format NETAPPDUMP, CELERRADUMP, or NDMPDUMP, the server will not perform any of the following functions:
 - Migration
 - Reclamation
 - Volume audits

- Data validation
- Simultaneous-write operations

For more information about data formats, see Chapter 8, “Using NDMP for operations with NAS file servers,” on page 167.

Example: Updating storage pools

You decide to increase the maximum size of a physical file that can be stored in the ENGBACK1 disk storage pool.

In this example, the ENGBACK1 disk storage pool is defined as shown in “Example: Defining storage pools” on page 231. To increase the maximum size of a physical file that can be stored in the storage pool, use the following command:

```
update stgpool engback1 maxsize=100m
```

Restrictions:

- You cannot use this command to change the data format for a storage pool.
- For storage pools that have the NETAPPDUMP, the CELERRADUMP, or the NDMPDUMP data format, you can modify the following parameters only:
 - ACCESS
 - COLLOCATE
 - DESCRIPTION
 - MAXSCRATCH
 - REUSEDELAY

Task tips for storage pools

Tivoli Storage Manager provides many functions, such as migration and reclamation, for optimizing data-storage operations. To take advantage of these functions, you can create specialized storage pools or specify certain properties in your storage pool definitions.

Table 24 gives tips on how to accomplish some tasks that are related to storage pools.

Table 24. Task tips for storage pools

For this Goal	Do This	For More Information
Keep the data for a group of client nodes, a single client node, or a client file space on as few volumes as possible.	Enable collocation for the storage pool.	“Keeping client files together using collocation” on page 310
Reduce the number of volume mounts needed to back up multiple clients.	Disable collocation for the storage pool.	“Keeping client files together using collocation” on page 310
Write data simultaneously to a primary storage pool and to copy storage pools and active-data pools.	Provide a list of copy storage pools and active-data pools when defining the primary storage pool.	“Writing data simultaneously to primary, copy, and active-data pools” on page 285
Specify how the server reuses tapes.	Set a reclamation threshold for the storage pool. Optional: Identify a reclamation storage pool	“Reclaiming space in sequential-access storage pools” on page 319

Table 24. Task tips for storage pools (continued)

For this Goal	Do This	For More Information
Move data from disk to tape automatically as needed.	Set a migration threshold for the storage pool. Identify the next storage pool.	"Migrating disk storage pools" on page 253
Move data from disk to tape automatically based on how frequently users access the data or how long the data has been in the storage pool.	Set a migration threshold for the storage pool. Identify the next storage pool. Set the migration delay period.	"Migrating disk storage pools" on page 253
Improve client restore performance using concurrent access to FILE volumes.	Implement a storage pool associated with the FILE device type.	"Defining storage pools" on page 227 "Setting up copy storage pools and active-data pools" on page 247
Back up your storage pools.	Implement a copy storage pool.	"Setting up copy storage pools and active-data pools" on page 247
Copy active data from a primary storage pool.	Implement an active-data pool.	"Setting up copy storage pools and active-data pools" on page 247
Have clients back up directly to a tape storage pool.	Define a sequential-access storage pool that uses a tape device class. Change the policy that the clients use, so that the backup copy group points to the tape storage pool as the destination.	"Defining storage pools" on page 227 "Changing policy" on page 439
Make the best use of available tape drives and FILE volumes during reclamation and migration.	Specify multiple concurrent processes.	"Optimizing drive usage using multiple concurrent reclamation processes" on page 322 "Specifying multiple concurrent migration processes" on page 261
Ensure that reclamation completes within the desired amount of time.	Limit the number of off-site volumes to be reclaimed.	"Reclamation of off-site volumes" on page 326 "Starting reclamation manually or in a schedule" on page 322
For storage pools associated with random-access and sequential-access disk (DISK and FILE device classes), automatically create private volumes and preassign them to specified storage pools when predetermined space utilization thresholds have been reached.	Use the DEFINE SPACETRIGGER and UPDATE SPACETRIGGER commands to specify the number and size of volumes.	"Preparing volumes for random-access storage pools" on page 236 "Defining storage pool volumes" on page 237
For storage pools associated with random-access disk (DISK device class) and sequential-access disk (FILE device class), create and format volumes using one command.	Use the DEFINE VOLUME command to specify the size and number of volumes to be created.	"Preparing volumes for random-access storage pools" on page 236 "Defining storage pool volumes" on page 237

Storage pool volumes

Storage pool volumes are the physical media that are assigned to a storage pool.

Some examples of volumes are:

- Space allocated on a disk drive
- A tape cartridge
- An optical disk

Storage pools and their volumes are either random access or sequential access, depending on the device type of the device class to which the pool is assigned.

Random-access storage pool volumes

Random-access storage pools consist of volumes on disk. Random-access storage pools are always associated with the DISK device class. All volumes in this type of storage pool have the same form.

All volumes are fixed-size files on a disk. You can create these volumes using the DEFINE VOLUME command, or you can use space triggers to automatically create and assign volumes to specified storage pools. Tivoli Storage Manager does not support raw logical volumes on Linux.

For additional information, see:

- “Preparing volumes for random-access storage pools” on page 236
- “Requirements for disk subsystems” on page 69

Sequential-access storage pool volumes

Sequential-access volumes are volumes in data is accessed sequentially, one block at a time, one after the other. Each volume defined in a sequential-access storage pool must be the same type as the device class associated with the storage pool.

You can define volumes in a sequential-access storage pool or you can specify that the server dynamically acquire scratch volumes. You can also use a combination of defined and scratch volumes. What you choose depends on the amount of control you want over individual volumes.

For information about preparing sequential-access volumes, see “Preparing volumes for sequential-access storage pools” on page 236.

Types of sequential-access volumes

Each Tivoli Storage Manager sequential-access device type is associated with a particular type of storage pool volume.

Some examples of sequential-access volumes are:

- Tape cartridge
- Optical disk
- File

Table 25 on page 235 lists the types of volumes associated with each device type.

Table 25. Volume types

Device Type	Volume Description	Label Required
3590	IBM 3590 tape cartridge	Yes
3592	IBM 3592 tape cartridge	Yes
4MM	4 mm tape cartridge	Yes
8MM	8 mm tape cartridge	Yes
DLT	A digital linear tape	Yes
ECARTRIDGE	A cartridge tape that is used by a tape drive such as the StorageTek SD-3 or 9490 tape drive	Yes
FILE	A file in the file system of the server machine	No
LTO	IBM Ultrium tape cartridge	Yes
NAS	A tape drive that is used for NDMP backups by a network-attached storage (NAS) file server	Yes
OPTICAL	A two-sided 5.25-inch rewritable optical cartridge	Yes
QIC	A 1/4-inch tape cartridge	Yes
REMOVABLEFILE	A file on a removable medium. If the medium has two sides, each side is a separate volume.	Yes
SERVER	One or more objects that are archived in the server storage of another server	No
VOLSAFE	A StorageTek cartridge tape that is for write-once use on tape drives that are enabled for VolSafe function.	No
WORM	A two-sided 5.25-inch write-once optical cartridge	Yes

Defined volumes

Use defined volumes when you want to control precisely which volumes are used in the storage pool. Defined volumes can also be useful when you want to establish a naming scheme for volumes.

You can also use defined volumes to reduce potential disk fragmentation and maintenance overhead for storage pools associated with random-access and sequential-access disk.

Scratch volumes

Use scratch volumes to enable the server to define a volume when needed and delete the volume when it becomes empty. Using scratch volumes frees you from the task of explicitly defining all of the volumes in a storage pool.

The server tracks whether a volume being used was originally a scratch volume. Scratch volumes that the server acquired for a primary storage pool are deleted from the server database when they become empty. The volumes are then available for reuse by the server or other applications.

Scratch volumes in a copy storage pool or an active-data storage pool are handled in the same way as scratch volumes in a primary storage pool, except for volumes with the access value of off-site. If an off-site volume becomes empty, the server does not immediately return the volume to the scratch pool. The delay prevents the empty volumes from being deleted from the database, making it easier to determine which volumes should be returned to the on-site location. The administrator can query the server for empty off-site copy storage pool volumes or active-data pool volumes, and return them to the on-site location. The volume is

returned to the scratch pool only when the access value is changed to READWRITE, READONLY, or UNAVAILABLE.

For scratch volumes that were acquired in a FILE device class, the space that the volumes occupied is freed by the server and returned to the file system.

Preparing volumes for random-access storage pools

Volumes in random-access storage pools must be defined before the server can access them.

Task	Required Privilege Class
Define volumes in any storage pool	System or unrestricted storage
Define volumes in specific storage pools	System, unrestricted storage, or restricted storage for those pools

To prepare a volume for use in a random-access storage pool, define the volume. For example, suppose you want to define a 21 MB volume for the BACKUPPOOL storage pool. You want the volume to be located in a particular path and named stgvol.001. Enter the following command:

```
define volume backuppool /usr/lpp/admserv/bin/stgvol.001 formatsize=21
```

If you do not specify a full path name for the volume name, the command uses the path associated with the registry key of this server instance.

You can also define volumes in a single step using the DEFINE VOLUME command. For example, to define ten, 5000 MB volumes in a random-access storage pool that uses a DISK device class, you would enter the following command.

```
define volume diskpool diskvol numberofvolumes=10 formatsize=5000
```

Remember:

- Define storage pool volumes on disk drives that reside on the Tivoli Storage Manager server machine, not on remotely mounted file systems. Network-attached drives can compromise the integrity of the data that you are writing.

You can also use a space trigger to automatically create volumes assigned to a particular storage pool.

Tivoli Storage Manager does not support raw logical volumes on Linux.

Preparing volumes for sequential-access storage pools

For most purposes, in a sequential-access storage pool, the server can use dynamically acquired scratch volumes, volumes that you define, or a combination of both.

For sequential-access storage pools with a FILE or SERVER device type, no labeling or other preparation of volumes is necessary. For sequential-access storage pools associated with device types other than a FILE or SERVER, you must prepare volumes for use.

When the server accesses a sequential-access volume, it checks the volume name in the header to ensure that the correct volume is being accessed. To prepare a volume:

1. Label the volume. Table 25 on page 235 shows the types of volumes that require labels. You must label those types of volumes before the server can use them.

For details, see:

“Labeling removable media volumes” on page 134.

Tip: When you use the LABEL LIBVOLUME command with drives in an automated library, you can label and check in the volumes with one command.

2. For storage pools in automated libraries, use the CHECKIN LIBVOLUME command to check the volume into the library. For details, see:

“Checking new volumes into a library” on page 137.

3. If you have not allowed scratch volumes in the storage pool, you must identify the volume, by name, to the server. For details, see “Defining storage pool volumes.”

If you allowed scratch volumes in the storage pool by specifying a value greater than zero for the MAXSCRATCH parameter, you can let the server use scratch volumes, identify volumes by name, or do both. See “Acquiring scratch volumes dynamically” on page 238 for information about scratch volumes.

Defining storage pool volumes

Defined volumes let you control precisely which volumes are used in the storage pool. Using defined volumes can also be useful when you want to establish a naming scheme for volumes.

Task	Required Privilege Class
Define volumes in any storage pool	System or unrestricted storage
Define volumes in specific storage pools	System, unrestricted storage, or restricted storage for those pools

When you define a storage pool volume, you inform the server that the volume is available for storing backup, archive, or space-managed data.

For a sequential-access storage pool, the server can use dynamically acquired scratch volumes, volumes that you define, or a combination.

To define a volume named VOL1 in the ENGBACK3 tape storage pool, enter:

```
define volume engback3 vol1
```

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries but that are used by the same server.

For storage pools associated with FILE device classes, you can define private volumes in a single step using the DEFINE VOLUME command. For example, to define ten, 5000 MB volumes, in a sequential-access storage pool that uses a FILE device class, you would enter the following command.

```
define volume filepool filevol numberofvolumes=10 formatsize=5000
```

For storage pools associated with the FILE device class, you can also use the DEFINE SPACETRIGGER and UPDATE SPACETRIGGER commands to have the server create volumes and assign them to a specified storage pool when predetermined space-utilization thresholds have been exceeded. One volume must be predefined.

Acquiring scratch volumes dynamically

If you allow sequential-access storage pools to use scratch volumes, you do not need to define volumes. You can control the maximum number of scratch volumes that the server can request using the MAXSCRATCH parameter on the DEFINE STGPOOL and UPDATE STGPOOL command.

To allow the storage pool to acquire volumes as needed, set the MAXSCRATCH parameter to a value greater than zero. The server automatically defines the volumes as they are acquired. The server also automatically deletes scratch volumes from the storage pool when the server no longer needs them.

Before the server can use a scratch volume with a device type other than FILE or SERVER, the volume must have a label.

Restriction: Tivoli Storage Manager only accepts tapes labeled with IBM standard labels. IBM standard labels are similar to ANSI Standard X3.27 labels except that the IBM standard labels are written in EBCDIC (extended binary coded decimal interchange code). For a list of IBM media sales contacts who can provide compatible tapes, go to the IBM Web site. If you are using non-IBM storage devices and media, consult your tape-cartridge distributor.

For details about labeling, see “Preparing volumes for sequential-access storage pools” on page 236.

Updating storage pool volumes

You can update a volume to reset an error state to an access mode of read/write. You can also update a volume to change the its location in a sequential-access storage pool. or to change the access mode of the volume, for example, if a tape cartridge is moved off-site or is damaged.

Task	Required Privilege Class
Update volumes	System or operator

To change the properties of a volume that has been defined to a storage pool, issue the UPDATE VOLUME command. For example, suppose you accidentally damage a volume named VOL1. To change the access mode to unavailable so that the server does not try to write or read data from the volume, issue the following command:

```
update volume vol1 access=unavailable
```

For details about access modes, see “Access modes for storage pool volumes” on page 239.

Volume properties that you can update

Update volume properties by changing the values of those properties in the volume definition.

Table 26 lists volume properties that you can update.

Table 26. Information for updating a storage pool volume

Information	Explanation
Volume name (Required)	Specifies the name of the storage pool volume to be updated. You can specify a group of volumes to update by using wildcard characters in the volume name. You can also specify a group of volumes by specifying the storage pool, device class, current access mode, or status of the volumes you want to update. See the parameters that follow.
New access mode	Specifies the new access mode for the volume (how users and server processes such as migration can access files in the storage pool volume). See “Access modes for storage pool volumes” for descriptions of access modes. A random-access volume must be varied offline before you can change its access mode to <i>unavailable</i> or <i>destroyed</i> . To vary a volume offline, use the VARY command. See “Varying disk volumes online or offline” on page 77. If a scratch volume that is empty and has an access mode of off-site is updated so that the access mode is read/write, read-only, or unavailable, the volume is deleted from the database.
Location	Specifies the location of the volume. This parameter can be specified only for volumes in sequential-access storage pools.
Storage pool	Restricts the update to volumes in the specified storage pool.
Device class	Restricts the update to volumes in the specified device class.
Current access mode	Restricts the update to volumes that currently have the specified access mode.
Status	Restricts the update to volumes with the specified status (online, offline, empty, pending, filling, or full).
Preview	Specifies whether you want to preview the update operation without actually performing the update.

Access modes for storage pool volumes

Access to a volume in a storage pool is determined by the access mode assigned to that volume. You can manually change the access mode of a volume, or the server can change the access mode based on what happens when it tries to access a volume.

For example, if the server cannot write to a volume having read/write access mode, the server automatically changes the access mode to read-only.

The following access modes apply to storage pool volumes:

Read/write

Allows files to be read from or written to a volume in the storage pool.

If the server cannot write to a read/write access volume, the server automatically changes the access mode to read-only.

If a scratch volume that is empty and has an access mode of off-site is updated so that the access mode is read/write, the volume is deleted from the database.

Read-only

Allows files to be read from but not written to a disk or tape volume.

If a scratch volume that is empty and has an access mode of off-site is updated so that the access mode is read-only, the volume is deleted from the database.

Unavailable

Specifies that the volume is not available for any type of access by the server.

You must vary offline a random-access volume before you can change its access mode to *unavailable*. To vary a volume offline, use the VARY command. See “Varying disk volumes online or offline” on page 77.

If a scratch volume that is empty and has an access mode of off-site is updated so that the access mode is unavailable, the volume is deleted from the database.

Destroyed

Specifies that a primary storage pool volume has been permanently damaged. Neither users nor system processes (like migration) can access files stored on the volume.

This access mode is used to indicate an entire volume that should be restored using the RESTORE STGPOOL or RESTORE VOLUME command. After all files on a destroyed volume are restored to other volumes, the destroyed volume is automatically deleted from the database. See “Storage pool restore processing” on page 757 for more information.

Only volumes in primary storage pools can be updated to an access mode of destroyed.

You must vary offline a random-access volume before you can change its access mode to *destroyed*. To vary a volume offline, use the VARY command. See “Varying disk volumes online or offline” on page 77. Once you update a random-access storage pool volume to destroyed, you cannot vary the volume online without first changing the access mode.

If you update a sequential-access storage pool volume to destroyed, the server does not attempt to mount the volume.

If a volume contains no files and the UPDATE VOLUME command is used to change the access mode to destroyed, the volume is deleted from the database.

Offsite

Specifies that a copy storage pool volume or active-data pool volume is at an off-site location and therefore cannot be mounted. Use this mode to help you track volumes that are off-site. The server treats off-site volumes differently, as follows:

- Mount requests are not generated for off-site volumes.
- Data can be reclaimed or moved from off-site volumes by retrieving files from other storage pools.
- Empty, off-site scratch volumes are not deleted from the copy storage pool or from the active-data pool.

You can only update volumes in a copy storage pool or an active-data pool to off-site access mode. Volumes that have the device type of SERVER (volumes that are actually archived objects stored on another Tivoli Storage Manager server) cannot have an access mode of off-site.

Storage pool hierarchies

You can arrange storage pools in a storage hierarchies, which consist of at least one primary storage pool to which a client node backs up, archives, or migrates data. Typically, data is stored initially in a disk storage pool for fast client restores, and then moved to a tape-based storage pool, which is slower to access but which has greater capacity. The location of all data objects is automatically tracked within the server database.

You can set up your devices so that the server automatically moves data from one device to another, or one media type to another. The selection can be based on characteristics such as file size or storage capacity. A typical implementation might have a disk storage pool with a subordinate tape storage pool. When a client backs up a file, the server might initially store the file on disk according to the policy for that file. Later, the server might move the file to tape when the disk becomes full. This action by the server is called *migration*. You can also place a size limit on files that are stored on disk, so that large files are stored initially on tape instead of on disk.

For example, your fastest devices are disks, but you do not have enough space on these devices to store all data that needs to be backed up over the long term. You have tape drives, which are slower to access, but have much greater capacity. You define a hierarchy so that files are initially stored on the fast disk volumes in one storage pool. This provides clients with quick response to backup requests and some recall requests. As the disk storage pool becomes full, the server migrates, or moves, data to volumes in the tape storage pool.

Another option to consider for your storage pool hierarchy is IBM 3592 tape cartridges and drives, which can be configured for an optimal combination of access time and storage capacity. For more information, see “Controlling data-access speeds for 3592 volumes” on page 205.

Migration of files from disk to sequential storage pool volumes is particularly useful because the server migrates all the files for a group of nodes or a single node together. This gives you partial collocation for clients. Migration of files is especially helpful if you decide not to enable collocation for sequential storage pools. For details, see “Keeping client files together using collocation” on page 310.

Setting up a storage pool hierarchy

To establish a hierarchy, identify the *next* storage pool, sometimes called the subordinate storage pool. The server migrates data to the next storage pool if the original storage pool is full or unavailable.

You can set up a storage pool hierarchy when you first define storage pools. You can also change the storage pool hierarchy later.

Restrictions:

- You cannot establish a chain of storage pools that leads to an endless loop. For example, you cannot define StorageB as the *next* storage pool for StorageA, and then define StorageA as the *next* storage pool for StorageB.

- The storage pool hierarchy includes only primary storage pools, not copy storage pools or active-data pools. See “Backing up the data in a storage hierarchy” on page 246.
- A storage pool must use the NATIVE or NONBLOCK data formats to be part of a storage pool hierarchy. For example, a storage pool using the NETAPPDUMP data format cannot be part of a storage pool hierarchy.

For detailed information about how migration between storage pools works, see “Migrating files in a storage pool hierarchy” on page 252.

Example: Defining a storage pool hierarchy

You determined that an engineering department requires a separate storage hierarchy. You set up policy so that the server initially stores backed up files for this department to a disk storage pool. When that pool fills, you want the server to migrate files to a tape storage pool.

You want the storage pools to have the following characteristics:

- Primary storage pool on disk
 - Name the storage pool ENGBACK1.
 - Limit the size of the largest file that can be stored to 5 MB. The server stores files that are larger than 5 MB in the tape storage pool.
 - Files migrate from the disk storage pool to the tape storage pool when the disk storage pool is 85% full. File migration to the tape storage pool stops when the disk storage pool is down to 40% full.
 - Use caching, so that migrated files stay on disk until the space is needed for other files.
- Primary storage pool on tape:
 - Name the storage pool BACKTAPE.
 - Use the device class TAPE, which has already been defined, for this storage pool.
 - Do not set a limit for the maximum file size, because this is the last storage pool in the hierarchy.
 - Use scratch volumes for this pool, with a maximum number of 100 volumes.

You can define the storage pools in a storage pool hierarchy from the top down or from the bottom up. Defining the hierarchy from the bottom up requires fewer steps. To define the hierarchy from the bottom up:

1. Define the storage pool named BACKTAPE with the following command:

```
define stgpool backtape tape
description='tape storage pool for engineering backups'
maxsize=nolimit collocate=node maxscratch=100
```

2. Define the storage pool named ENGBACK1 with the following command:

```
define stgpool engback1 disk
description='disk storage pool for engineering backups'
maxsize=5M nextstgpool=backtape highmig=85 lowmig=40
```

Example: Updating a storage pool hierarchy

You already defined the ENGBACK1 disk storage pool. Now you decide to set up a tape storage pool to which files from ENGBACK1 can migrate.

If you have already defined the storage pool at the top of the hierarchy, you can update the storage hierarchy to include a new storage pool.

To define the new tape storage pool and update the hierarchy:

1. Define the storage pool named BACKTAPE with the following command:

```
define stgpool backtape tape
description='tape storage pool for engineering backups'
maxsize=nolimit collocate=node maxscratch=100
```
2. Update the storage-pool definition for ENGBACK1 to specify that BACKTAPE is the next storage pool defined in the storage hierarchy:

```
update stgpool engback1 nextstgpool=backtape
```

How the server groups files before storing

When client files are backed up or archived, the server can group them into an *aggregate* of files. By controlling the size of aggregates, you can control the performance of client operations.

The size of the aggregate depends on the sizes of the client files being stored, and the number of bytes and files allowed for a single transaction. Two options affect the number of files and bytes allowed for a single transaction. TXNGROUPMAX, located in the server options file, affects the number of files allowed. TXNBYTELIMIT, located in the client options file, affects the number of bytes allowed in the aggregate.

- The TXNGROUPMAX option in the server options file indicates the maximum number of logical files (client files) that a client may send to the server in a single transaction. The server might create multiple aggregates for a single transaction, depending on how large the transaction is.

It is possible to affect the performance of client backup, archive, restore, and retrieve operations by using a larger value for this option. When transferring multiple small files, increasing the TXNGROUPMAX option can improve throughput for operations to tape.

Important: 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. For more information, see “Files moved as a group between client and server” on page 610.

You can override the value of the TXNGROUPMAX server option for individual client nodes by using the TXNGROUPMAX parameter in the REGISTER NODE and UPDATE NODE commands.

- The TXNBYTELIMIT option in the client options file indicates the total number of bytes that the client can send to the server in a single transaction.

When a Tivoli Storage Manager for Space Management client (HSM client) migrates files to the server, the files are not grouped into an aggregate.

Where the server stores files

When a client file is backed up, archived, or migrated, the server verifies the management class that is bound to the file. The management class specifies the destination storage pool in which to store the file.

The server checks the destination storage pool to determine:

- If it is possible to write file data to the storage pool (access mode).
- If the size of the physical file exceeds the maximum file size allowed in the storage pool. For backup and archive operations, the physical file may be an aggregate or a single client file.
- Whether sufficient space is available on the available volumes in the storage pool.
- What the next storage pool is, if any of the previous conditions prevent the file from being stored in the storage pool that is being checked.

Using these factors, the server determines if the file can be written to that storage pool or the next storage pool in the hierarchy.

Subfile backups: When the client backs up a subfile, it still reports the size of the entire file. Therefore, allocation requests against server storage and placement in the storage hierarchy are based on the full size of the file. The server does not put a subfile in an aggregate with other files if the size of the entire file is too large to put in the aggregate. For example, the entire file is 8 MB, but the subfile is only 10 KB. The server does not typically put a large file in an aggregate, so the server begins to store this file as a stand-alone file. However, the client sends only 10 KB, and it is now too late for the server to put this 10 KB file with other files in an aggregate. As a result, the benefits of aggregation are not always realized when clients back up subfiles.

Example: How the server determines where to store files in a hierarchy

The server determines where to store a file based upon the destination storage pool specified in the copy group of the management class to which the file is bound. The server also checks the capacity utilization of the storage pool and the maximum file size allowed.

Assume a company has a storage pool hierarchy as shown in Figure 20 on page 245.

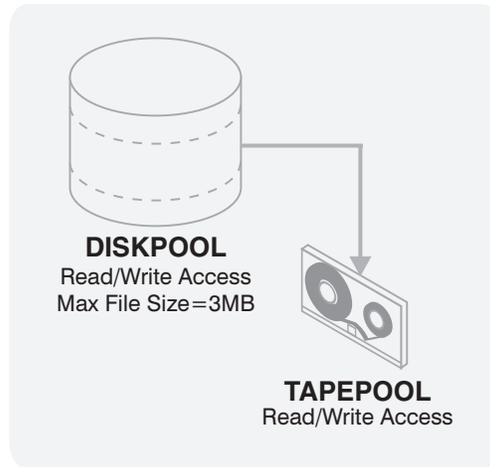


Figure 20. Storage hierarchy example

The storage pool hierarchy consists of two storage pools:

DISKPOOL

The top of the storage hierarchy. It contains fast disk volumes for storing data.

TAPEPOOL

The next storage pool in the hierarchy. It contains tape volumes accessed by high-performance tape drives.

Assume a user wants to archive a 5 MB file that is named *FileX*. *FileX* is bound to a management class that contains an archive copy group whose storage destination is DISKPOOL, see Figure 20.

When the user archives the file, the server determines where to store the file based on the following process:

1. The server selects DISKPOOL because it is the storage destination specified in the archive copy group.
2. Because the access mode for DISKPOOL is read/write, the server checks the maximum file size allowed in the storage pool.

The maximum file size applies to the physical file being stored, which may be a single client file or an aggregate. The maximum file size allowed in DISKPOOL is 3 MB. *FileX* is a 5 MB file and therefore cannot be stored in DISKPOOL.

3. The server searches for the next storage pool in the storage hierarchy. If the DISKPOOL storage pool has no maximum file size specified, the server checks for enough space in the pool to store the physical file. If there is not enough space for the physical file, the server uses the next storage pool in the storage hierarchy to store the file.
4. The server checks the access mode of TAPEPOOL, which is the next storage pool in the storage hierarchy. The access mode for TAPEPOOL is read/write.
5. The server then checks the maximum file size allowed in the TAPEPOOL storage pool. Because TAPEPOOL is the last storage pool in the storage hierarchy, no maximum file size is specified. Therefore, if there is available space in TAPEPOOL, *FileX* can be stored in it.

Backing up the data in a storage hierarchy

You can use copy storage pools and active-data pools to protect the data in primary storage pools. Copy storage pools can contain any combination of active and inactive data, archive data, or space-migrated data. Active-data pools contain only active versions of client backup data.

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.

As a best practice, therefore, and to prevent the permanent loss of inactive versions of client backup data, you should create a minimum of one active-data pool, which contains active-data only, and one copy storage pool, which contains both active and inactive data. To recover from a disaster, use the active-data pool to restore critical client node data, and then restore the primary storage pools from the copy storage pool. Do not use active-data pools for recovery of a primary pool or volume unless the loss of inactive data is acceptable.

“Setting up copy storage pools and active-data pools” on page 247 describes the high-level steps for implementation.

Neither copy storage pools nor active-data pools are part of a storage hierarchy, which, by definition, consists only of primary storage pools. Data can be stored in copy storage pools and active-data pools using the following methods:

- Including the `BACKUP STGPOOL` and `COPY ACTIVEDATA` commands in administrative scripts or schedules so that data is automatically backed up or copied at regular intervals.
- Enabling the simultaneous-write function so that data is written to primary storage pools, copy storage pools, and active-data pools during the same transaction. Writing data simultaneously to copy storage pools is supported for backup, archive, space-management, and import operations. Writing data simultaneously to active-data pools is supported only for client backup operations and only for active backup versions.
- (*copy storage pools only*) Manually issuing the `BACKUP STGPOOL` command, specifying the primary storage pool as the source and a copy storage pool as the target. The `BACKUP STGPOOL` command backs up whatever data is in the primary storage pool (client backup data, archive data, and space-managed data).
- (*active-data pools only*) Manually issuing the `COPY ACTIVEDATA` command, specifying the primary storage pool as the source and an active-data pool as the target. The `COPY ACTIVEDATA` command copies only the active versions of client backup data. If an aggregate being copied contains all active files, then the entire aggregate is copied to the active-data pool during command processing. If an aggregate being copied contains some inactive files, the aggregate is reconstructed during command processing into a new aggregate without the inactive files.

For efficiency, you can use a single copy storage pool and a single active-data pool to back up all primary storage pools that are linked in a storage hierarchy. By backing up all primary storage pools to one copy storage pool and one active-data pool, you do not need to repeatedly copy a file when the file migrates from its original primary storage pool to another primary storage pool in the storage hierarchy.

In most cases, a single copy storage pool and a single active-data pool can be used for backup of all primary storage pools. However, the number of copy storage pools and active-data pools you actually need depends on whether you have more than one primary storage pool hierarchy and on the type of disaster recovery protection you want to implement. Multiple copy storage pools and active-data pools might be needed to handle particular situations, including the following:

- Special processing of certain primary storage hierarchies (for example, archive storage pools or storage pools dedicated to priority clients)
- Creation of multiple copies for multiple locations (for example, to keep one copy on-site and one copy off-site)
- Rotation of full storage pool backups (See “Backing up storage pools” on page 760.)

Inactive files in volumes in an active-data pool are deleted by reclamation processing. The rate at which reclaimable space accumulates in active-data pool volumes is typically faster than the rate for volumes in non-active-data pools. If reclamation of volumes in an active-data pool is occurring too frequently, requiring extra resources such as tape drives and libraries to mount and dismount volumes, you can adjust the reclamation threshold until the rate of reclamation is acceptable. The default reclamation threshold for active-data pools is 60 percent, which means that reclamation begins when the storage pool reaches 60 percent of capacity. Note that accelerated reclamation of volumes has more of an effect on active-data pools that use removable media and, in particular, on removable media that is taken off-site.

Setting up copy storage pools and active-data pools

To back up the data in primary storage pools, use copy storage pools, active-data pools, or combination of the two.

To set up a copy storage pool or an active-data pool:

1. Define a copy storage pool or active-data pool. For details, see “Defining copy storage pools and active-data pools” on page 357.
2. (*active-data pools only*) Create a policy domain, and specify the name of the active-data pool as the value of the `ACTIVEDATAPOOL` parameter. To learn more about creating domains and the `ACTIVEDATAPOOL` parameter, see “Defining and updating a policy domain” on page 458.
3. (*active-data pools only*) Identify the nodes whose active backup data is to be stored in the active-data pool, and then assign the nodes to the domain defined in step 2. For details about assigning nodes to a domain, see “Assigning client nodes to a policy domain” on page 472.
4. (*optional*) If you want to use the simultaneous-write function, update the primary storage pool definition, specifying the name of the copy storage pool and active-data pool as the values of the `COPYSTGPOOLS` and `ACTIVEDATAPOOLS` parameters, respectively. For details about the simultaneous-write function, see “Writing data simultaneously to primary, copy, and active-data pools” on page 285.
5. Set up administrative schedules or scripts to automatically issue the `BACKUP STGPOOL` and `COPY ACTIVEDATA` commands. See “Automating a basic administrative command schedule” on page 572 and “IBM Tivoli Storage Manager server scripts” on page 578.

Example: Setting up an active-data pool for fast client restore:

A sequential-access disk (FILE) device class is used to set up an active-data pool for fast restore of client-node data.

decide which client nodes have data that needs to be restored quickly if a disaster occurs. Only the data belonging to those nodes should be stored in the active-data pool.

For the purposes of this example, the following definitions already exist on the server:

- The default STANDARD domain, STANDARD policy set, STANDARD management class, and STANDARD copy group.
- A primary storage pool, BACKUPPOOL, and a copy storage pool, COPYPOOL. BACKUPPOOL is specified in the STANDARD copy group as the storage pool in which the server initially stores backup data. COPYPOOL contains copies of all the active and inactive data in BACKUPPOOL.
- Three nodes that are assigned to the STANDARD domain (NODE1, NODE2, and NODE 3).
- A FILE device class named FILECLASS.

You have identified NODE2 as the only high-priority node, so you need to create a new domain to direct the data belonging to that node to an active-data pool. To set up and enable the active-data pool, follow these steps:

1. Define the active-data pool:

```
DEFINE STGPOOL ADPPool FILECLASS POOLTYPE=ACTIVEDATA MAXSCRATCH=1000
```

2. Define a new domain and specify the active-data pool in which you want to store the data belonging to NODE2:

```
DEFINE DOMAIN ACTIVEDOMAIN ACTIVEDESTINATION=ADPPool
```

3. Define a new policy set:

```
DEFINE POLICYSET ACTIVEDOMAIN ACTIVEPOLICY
```

4. Define a new management class:

```
DEFINE MGMTCLASS ACTIVEDOMAIN ACTIVEPOLICY ACTIVEMGMT
```

5. Define a backup copy group:

```
DEFINE COPYGROUP ACTIVEDOMAIN ACTIVEPOLICY ACTIVEMGMT DESTINATION=BACKUPPOOL
```

This command specifies that the active and inactive data belonging to client nodes that are members of ACTIVEDOMAIN will be backed up to BACKUPPOOL. Note that this is the destination storage pool for data backed up from nodes that are members of the STANDARD domain.

6. Assign the default management class for the active-data pool policy set:

```
ASSIGN DEFMGMTCLASS ACTIVEDOMAIN ACTIVEPOLICY ACTIVEMGMT
```

7. Activate the policy set for the active-data pool:

```
ACTIVATE POLICYSET ACTIVEDOMAIN ACTIVEPOLICY
```

8. Assign the high-priority node, NODE2, to the new domain:

```
UPDATE NODE NODE2 DOMAIN=ACTIVEDOMAIN
```

A node can belong to only one domain. When you update a node by changing its domain, you remove it from its current domain.

9. (optional) Update the primary storage pool, BACKUPPOOL, with the name of the active-data pool, ADPPPOOL, where the server simultaneously will write data during a client backup operation:

```
UPDATE STGPOOL BACKUPPOOL ACTIVEDATAPOOLES=ADPPPOOL
```

Only active versions of backup data can be simultaneously written to active-data pools.

10. To ensure that copies of active data are complete, define a schedule to copy active data from BACKUPPOOL to ADPPPOOL every day at 8:00 p.m.:

```
DEFINE SCHEDULE COPYACTIVE_BACKUPPOOL TYPE=ADMINISTRATIVE  
  CMD="COPY ACTIVE DATA BACKUPPOOL ADPPPOOL" ACTIVE=YES  
  STARTTIME=20:00 PERIOD=1
```

Instead of defining a schedule, you can issue the COPY ACTIVE DATA command manually whenever it is convenient to copy the active data.

Every time NODE2 stores data into BACKUPPOOL, the server simultaneously writes the data to ADPPPOOL. The schedule, COPYACTIVE_BACKUPPOOL, ensures that any data that was not stored during simultaneous-write operations is copied to the active-data pool. When client nodes NODE1 and NODE3 are backed up, their data is stored in BACKUPPOOL only, and not in ADPPPOOL. When the administrative schedule runs, only the data belonging to NODE2 is copied to the active-data pool.

Remember: If you want all the nodes belonging to an existing domain to store their data in the active-data pool, then you can skip steps 2 through 8. Use the UPDATE DOMAIN command to update the STANDARD domain, specifying the name of the active-data pool, ADPPPOOL, as the value of the ACTIVEDESTINATION parameter.

Example: Setting up an active-data pool to reduce media resources:

Backup data is simultaneously written to an active-data pool so that volumes in the pool can be taken off-site.

In addition to using active-data pools for fast restore of client-node data, you can also use active-data pools to reduce the number of tape volumes that are stored either on-site or off-site for the purpose of disaster recovery. This example assumes that, in your current configuration, all data is backed up to a copy storage pool and taken off-site. However, your goal is to create an active-data pool, take the volumes in that pool off-site, and maintain the copy storage pool on-site to recover primary storage pools.

Attention: Active-data pools should not be considered for recovery of a primary pool or volume unless the loss of inactive data is acceptable.

The following definitions already exist on the server:

- The default STANDARD domain, STANDARD policy set, STANDARD management class, and STANDARD copy group.
- A primary storage pool, BACKUPPOOL, and a copy storage pool, COPYPOOL. BACKUPPOOL is specified in the STANDARD copy group as the storage pool in which the server initially stores backup data. COPYPOOL contains copies of all the active and inactive data in BACKUPPOOL.

- An administrative schedule, named `BACKUP_BACKUPPOOL`, that issues a `BACKUP STGPOOL` command to back up the data in `BACKUPPOOL` to `COPYPOOL`. The schedule runs every day at 10:00 p.m.
- Three nodes that are assigned to the `STANDARD` domain (`NODE1`, `NODE2`, and `NODE 3`).
- A device class of type 3592 named `3592CLASS`.

To set up and enable an active-data pool, follow these steps:

1. Define the active-data pool:

```
DEFINE STGPOOL ADPPool 3592CLASS POOLTYPE=ACTIVEDATA MAXSCRATCH=1000
```

2. Update the `STANDARD` domain to allow data from all nodes to be stored in the active-data pool:

```
UPDATE DOMAIN STANDARD ACTIVEDESTINATION=ADPPool
```

3. (*optional*) Update the primary storage pool, `BACKUPPOOL`, with the name of the active-data pool, `ADPPool`, where the server will write data simultaneously during client backup operations:

```
UPDATE STGPOOL BACKUPPOOL ACTIVEDATAPool=ADPPool
```

Only active versions of backup data can be simultaneously written to active-data pools.

4. To ensure that copies of active data are complete, define a schedule to copy active data from `BACKUPPOOL` to `ADPPool` every day at 8:00 p.m.:

```
DEFINE SCHEDULE COPYACTIVE_BACKUPPOOL TYPE=ADMINISTRATIVE
  CMD="COPY ACTIVEDATA BACKUPPOOL ADPPool" ACTIVE=YES STARTTIME=20:00 PERIOD=1
```

Instead of defining a schedule, you can issue the `COPY ACTIVEDATA` command manually whenever it is convenient to copy the active data.

Every time data is stored into `BACKUPPOOL`, the data is simultaneously written to `ADPPool`. The schedule, `COPYACTIVE_BACKUPPOOL`, ensures that any data that was not stored during a simultaneous-write operation is copied to the active-data pool. You can now move the volumes in the active-data pool to a safe location off-site.

If your goal is to replace the copy storage pool with the active-data pool, follow the steps below. As a best practice and to protect your inactive data, however, you should maintain the copy storage pool so that you can restore inactive versions of backup data if required. If the copy storage pool contains archive or files that were migrated by a Tivoli Storage Manager for Space Management client, do not delete it.

1. Stop backing up to the copy storage pool:

```
DELETE SCHEDULE BACKUP_BACKUPPOOL
UPDATE STGPOOL BACKUPPOOL COPYSTGPOOLS=""
```

2. After all data has been copied to the active-data pool, delete the copy storage pool and its volumes.

Staging client data from disk to tape

Typically, client backup data is stored initially in disk-based storage pools. To make room for additional backups, you can migrate the older data to tape. If you are using copy storage pools or active-data pools, store data in those pools before beginning the migration process.

Typically you need to ensure that you have enough disk storage to process one night's worth of the clients' incremental backups. While not always possible, this guideline proves to be valuable when considering storage pool backups.

For example, suppose you have enough disk space for nightly incremental backups for clients, but not enough disk space for a FILE-type, active-data pool. Suppose also that you have tape devices. With these resources, you can set up the following pools:

- A primary storage pool on disk, with enough volumes assigned to contain the nightly incremental backups for clients
- A primary storage pool on tape, which is identified as the next storage pool in the hierarchy for the disk storage pool
- An active-data pool on tape
- A copy storage pool on tape

You can then schedule the following steps every night:

1. Perform an incremental backup of the clients to the disk storage pool.
2. After clients complete their backups, back up the active and inactive versions in the disk primary storage pool (now containing the incremental backups) to the copy storage pool. Then copy the active backup versions to the active-data pool.

Backing up disk storage pools before migration processing allows you to copy as many files as possible while they are still on disk. This saves mount requests while performing your storage pool backups. If the migration process starts while active data is being copied to active-data pools or while active and inactive data is being backed up to copy storage pools, some files might be migrated before they are copied or backed up.

3. Start the migration of the files in the disk primary storage pool to the tape primary storage pool (the next pool in the hierarchy) by lowering the high migration threshold. For example, lower the threshold to 40%.

When this migration completes, raise the high migration threshold back to 100%.

4. To ensure that all files are backed up, back up the tape primary storage pool to the copy storage pool. In addition, copy the active backup data in the tape primary storage pool to the active-data pool.

The tape primary storage pool must still be backed up (and active files copied) to catch any files that might have been missed in the backup of the disk storage pools (for example, large files that went directly to sequential media).

For more information about storage pool space, see “Estimating space needs for storage pools” on page 330

Migrating files in a storage pool hierarchy

To maintain free space in primary storage pools, the Tivoli Storage Manager server can automatically migrate data from one primary pool to the next storage pool in the hierarchy. You can control when migration begins and ends, which files to migrate, and whether to run concurrent migration processes.

The migration process helps to ensure that there is sufficient free space in the storage pools at the top of the hierarchy, where faster devices can provide the most benefit to clients. For example, the server can migrate data stored in a random-access disk storage pool to a slower but less expensive sequential-access storage pool.

You can control:

When migration begins and ends

Migration thresholds are used to control when migration begins and ends. Thresholds are set as levels of the space that is used in a storage pool, and expressed as a percent of total space available in the storage pool. For random-access and sequential-access disk storage pools, the server compares the threshold to the amount of data stored in the pool as a percent of the total data capacity of the volumes in the pool. Total data capacity for sequential-access disk storage pools includes the capacity of all scratch volumes specified for the pool. For tape and optical storage pools, the server compares the threshold to the number of volumes containing data as a percent of the total number of volumes available to the pool, including scratch volumes.

You can also schedule migration activities to occur when they are most convenient to you. In addition, you can specify how long migration will run before being automatically canceled, whether the server attempts reclamation before migration, and whether the migration process runs in the background or foreground.

How the server chooses files to migrate

By default, the server does not consider how long a file has been in a storage pool or how long since a file was accessed before choosing files to migrate. Optional parameters allow you to change the default. You can ensure that files remain in a storage pool for a minimum number of days before the server migrates them to another pool. To do this, you set a migration delay period for a storage pool. Before the server can migrate a file, the file must be stored in the storage pool at least as long as the migration delay period. For random-access disk storage pools, the last time the file was accessed is also considered for migration delay. For sequential-access storage pools, including sequential-access disk storage pools associated with a FILE device class, all files on a volume must exceed the value specified as a migration delay before the server migrates all of the files on the volume.

The number of concurrent migration processes

You can specify a single migration process or multiple concurrent migration processes for a random-access or sequential-access storage pool. Multiple concurrent processes let you make better use of your available tape drives and FILE volumes. However, because you can perform migration concurrently on different storage pools during auto-migration, you must carefully consider the resources (for example, drives) you have available for the operation.

If simultaneous-write operations during migration are enabled during multiple concurrent-migration processing, each process has the following requirements:

- A mount point
- A volume for each copy storage pool and active-data pool that is defined to the target storage pool and the primary pool

For details about the simultaneous-write function, see “Writing data simultaneously to primary, copy, and active-data pools” on page 285.

Migration processing can differ for disk storage pools versus sequential-access storage pools. If you plan to modify the default migration parameter settings for storage pools or want to understand how migration works, read the following topics:

- “Migrating disk storage pools”
- “Migrating sequential-access storage pools” on page 258
- “Starting migration manually or in a schedule” on page 261

Remember:

- If you receive an error message during the migration process, refer to *IBM Tivoli Storage Manager Messages*, which can provide useful information for diagnosing and fixing problems.

Migrating disk storage pools

Migration thresholds specify when the server should begin and stop migrating data to the next storage pool in the storage hierarchy. Migration thresholds are defined as a percentage of total storage-pool data capacity.

You can use the defaults for the migration thresholds, or you can change the threshold values to identify the maximum and minimum amount of space for a storage pool.

To control how long files must stay in a storage pool before they are eligible for migration, specify a migration delay for a storage pool. For details, see “Keeping files in a storage pool” on page 257.

If you decide to enable cache for disk storage pools, files can temporarily remain on disks even after migration. When you use cache, you might want to set lower migration thresholds.

For more information about migration thresholds, see “How the server selects files to migrate” on page 254 and “Migration thresholds” on page 256. For information about using the cache, see “Minimizing access time to migrated files” on page 258 and “Caching in disk storage pools” on page 263.

How the server selects files to migrate

When data in a storage pool comprises a percentage of the pool's capacity that is equal to the high migration threshold, the server migrates files from the pool to the next storage pool. The process for selecting files to migrate is based on the space consumed by a client node's files and on the setting for migration delay.

The server selects the files to migrate as follows:

1. The server checks for the client node that has backed up or migrated the largest single file space or has archived files that occupy the most space.
2. For *all* files from *every* file space belonging to the client node that was identified, the server examines the number of days since the files were stored in the storage pool and last retrieved from the storage pool. The server compares the number (whichever is less) to the migration delay that is set for the storage pool. The server migrates any of these files for which the number is more than the migration delay set for the storage pool.
3. After the server migrates the files for the first client node to the next storage pool, the server checks the low migration threshold for the storage pool. If the amount of space that is used in the storage pool is now below the low migration threshold, migration ends. If not, the server chooses another client node by using the same criteria as described above, and the migration process continues.

The server may not be able to reach the low migration threshold for the pool by migrating only files that have been stored longer than the migration delay period. When this happens, the server checks the storage pool characteristic that determines whether migration should stop even if the pool is still above the low migration threshold. For more information, see "Keeping files in a storage pool" on page 257.

If multiple migration processes are running (controlled by the MIGPROCESS parameter of the DEFINE STGPOOL command), the server may choose the files from more than one node for migration at the same time.

For example, Table 27 displays information that is contained in the database that is used by the server to determine which files to migrate. This example assumes that the storage pool contains no space-managed files. This example also assumes that the migration delay period for the storage pool is set to zero, meaning any files can be migrated regardless of time stored in the pool or the last time of access.

Table 27. Database information on files stored in DISKPOOL

Client Node	Backed-Up File Spaces and Sizes		Archived Files (All Client File Spaces)
TOMC	TOMC/C	200 MB	55 MB
	TOMC/D	100 MB	
CAROL	CAROL	50 MB	5 MB
PEASE	PEASE/home	150 MB	40 MB
	PEASE/temp	175 MB	

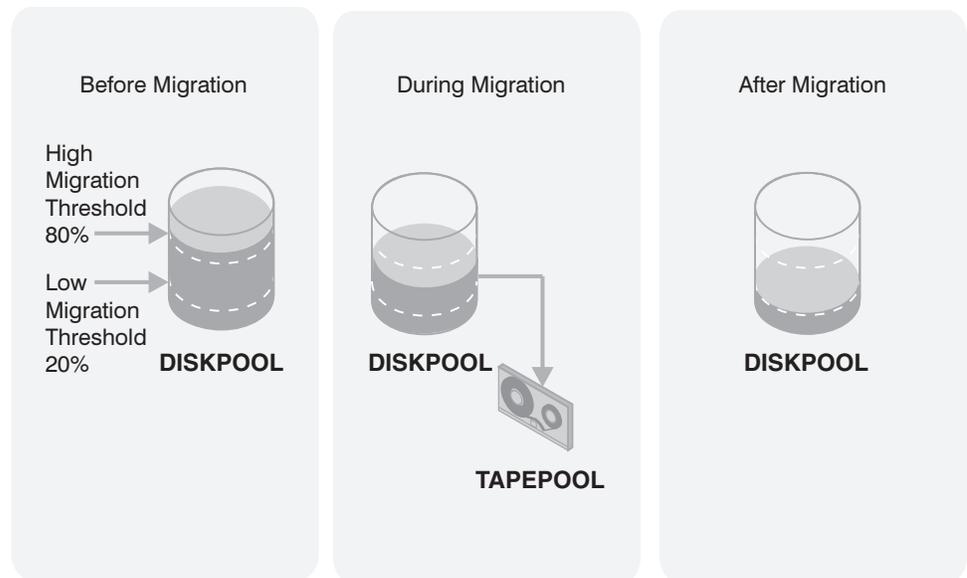


Figure 21. The migration process and migration thresholds

Figure 21 shows what happens when the high migration threshold defined for the disk storage pool DISKPOOL is exceeded. When the amount of migratable data in DISKPOOL reaches 80%, the server performs the following tasks:

1. Determines that the TOMC/C file space is taking up the most space in the DISKPOOL storage pool, more than any other single backed-up or space-managed file space and more than any client node's archived files.
2. Locates all data belonging to node TOMC stored in DISKPOOL. In this example, node TOMC has backed up or archived files from file spaces TOMC/C and TOMC/D stored in the DISKPOOL storage pool.
3. Migrates all data from TOMC/C and TOMC/D to the next available storage pool. In this example, the data is migrated to the tape storage pool, TAPEPOOL.

The server migrates all of the data from both file spaces belonging to node TOMC, even if the occupancy of the storage pool drops below the low migration threshold before the second file space has been migrated.

If the cache option is enabled, files that are migrated remain on disk storage (that is, the files are *cached*) until space is needed for new files. For more information about using cache, see “Caching in disk storage pools” on page 263.

4. After all files that belong to TOMC are migrated to the next storage pool, the server checks the low migration threshold. If the low migration threshold has not been reached, then the server again determines which client node has backed up or migrated the largest single file space or has archived files that occupy the most space. The server begins migrating files belonging to that node.

In this example, the server migrates *all* files that belong to the client node named PEASE to the TAPEPOOL storage pool.

5. After all the files that belong to PEASE are migrated to the next storage pool, the server checks the low migration threshold again. If the low migration threshold has been reached or passed, then migration ends.

Migration thresholds

Migration thresholds specify when migration for a storage pool begins and ends. Setting migration thresholds for disk storage pools ensures sufficient free space on faster devices, which can lead to better performance.

Choosing thresholds appropriate for your situation takes some experimenting. Start by using the default high and low values. You need to ensure that migration occurs frequently enough to maintain some free space but not so frequently that the device is unavailable for other use.

High-migration thresholds:

Before changing the high-migration threshold, you need to consider the amount of storage capacity provided for each storage pool and the amount of free storage space needed to store additional files, without having migration occur.

If you set the high-migration threshold too high, the pool may be just under the high threshold, but not have enough space to store an additional, typical client file. Or, with a high threshold of 100%, the pool may become full and a migration process must start before clients can back up any additional data to the disk storage pool. In either case, the server stores client files directly to tape until migration completes, resulting in slower performance.

If you set the high-migration threshold too low, migration runs more frequently and can interfere with other operations.

Keeping the high-migration threshold at a single value means that migration processing could start at any time of day, whenever that threshold is exceeded. You can control when migration occurs by using administrative command schedules to change the threshold. For example, set the high-migration threshold to 95% during the night when clients run their backup operations. Lower the high-migration threshold to 50% during the time of day when you want migration to occur. By scheduling when migration occurs, you can choose a time when your tape drives and mount operators are available for the operation.

Low-migration thresholds:

Before setting the low-migration threshold, you need to consider the amount of free disk storage space needed for normal daily processing, whether you use cache on disk storage pools, how frequently you want migration to occur, and whether data in the next storage pool is being collocated by group.

To choose the low-migration threshold, consider:

- The amount of free disk storage space needed for normal daily processing. If you have disk space to spare, you can keep more data on the disk (a larger low threshold). If clients' daily backups are enough to fill the disk space every day, you may need to empty the disk (a smaller low threshold).

If your disk space is limited, try setting the threshold so that migration frees enough space for the pool to manage the amount of client data that is typically stored every day. Migration then runs about every day, or you can force it to run every day by lowering the high-migration threshold at a time you choose.

You may also want to identify clients that are transferring large amounts of data daily. For these clients, you may want to set up policy (a new copy group or a new policy domain) so that their data is stored directly to tape. Using a separate policy in this way can optimize the use of disk for the majority of clients.

- Whether you use cache on disk storage pools to improve how quickly some files are retrieved. If you use cache, you can set the low threshold lower, yet still maintain faster retrieval for some data. Migrated data remains cached on the disk until new client data pushes the data off the disk. Using cache requires more disk space for the database, however, and can slow backup and archive operations that use the storage pool.

If you do not use cache, you may want to keep the low threshold at a higher number so that more data stays on the disk.

- How frequently you want migration to occur, based on the availability of sequential-access storage devices and mount operators. The larger the low threshold, the shorter time that a migration process runs (because there is less data to migrate). But if the pool refills quickly, then migration occurs more frequently. The smaller the low threshold, the longer time that a migration process runs, but the process runs less frequently.

You may need to balance the costs of larger disk storage pools with the costs of running migration (drives, tapes, and either operators or automated libraries).

- Whether data in the next storage pool is being collocated by group. During migration from a disk storage pool, all the data for all nodes belonging to the same collocation group are migrated by the same process. Migration will continue regardless whether the low migration threshold has been reached or the amount of data that the group has to migrate.

Keeping files in a storage pool

For some applications, you might want to delay the migration of files in the storage pool where they were initially stored by the server. You can delay migration of files for a specified number of days.

For example, you might have backups of monthly summary data that you want to keep in your disk storage pool for faster access until the data is 30 days old. After the 30 days, the server moves the files to a tape storage pool.

To delay file migration of files, set the `MIGDELAY` parameter when you define or update a storage pool. The number of days is counted from the day that a file was stored in the storage pool or accessed by a client, whichever is more recent. You can set the migration delay separately for each storage pool. When you set the delay to zero, the server can migrate any file from the storage pool, regardless of how short a time the file has been in the storage pool. When you set the delay to greater than zero, the server checks how long the file has been in the storage pool and when it was last accessed by a client. If the number of days exceeds the migration delay, the server migrates the file.

Note: If you want the number of days for migration delay to be counted based only on when a file was stored and not when it was retrieved, use the `NORETRIEVEDATE` server option. For more information about this option, see the *Administrator's Reference*.

If you set migration delay for a pool, you must decide what is more important: either ensuring that files stay in the storage pool for the migration delay period, or ensuring that there is enough space in the storage pool for new files. For each storage pool that has a migration delay set, you can choose what happens as the server tries to move enough data out of the storage pool to reach the low migration threshold. If the server cannot reach the low migration threshold by moving only files that have been stored longer than the migration delay, you can choose one of the following:

- Allow the server to move files out of the storage pool even if they have not been in the pool for the migration delay (MIGCONTINUE=YES). This is the default. Allowing migration to continue ensures that space is made available in the storage pool for new files that need to be stored there.
- Have the server stop migration without reaching the low migration threshold (MIGCONTINUE=NO). Stopping migration ensures that files remain in the storage pool for the time you specified with the migration delay. The administrator must ensure that there is always enough space available in the storage pool to hold the data for the required number of days.

If you allow more than one migration process for the storage pool and allow the server to move files that do not satisfy the migration delay time (MIGCONTINUE=YES), 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 storage pool to meet the low migration threshold.

Minimizing access time to migrated files

Caching is a method of minimizing access time to files on disk storage, even if the server has migrated files to a tape storage pool. However, cached files are removed from disk when the space they occupy is required. The files must then be obtained from the storage pool to which they were migrated

Important: For information about the disadvantages of using cache, see “Caching in disk storage pools” on page 263.

To ensure that files remain on disk storage and do not migrate to other storage pools, use one of the following methods:

- Do not define the *next* storage pool.

A disadvantage of using this method is that if the file exceeds the space available in the storage pool, the operation to store the file fails.

- Set the high-migration threshold to 100%.

When you set the high migration threshold to 100%, files will not migrate at all. You can still define the *next* storage pool in the storage hierarchy, and set the maximum file size so that large files are stored in the next storage pool in the hierarchy.

A disadvantage of setting the high threshold to 100% is that after the pool becomes full, client files are stored directly to tape instead of to disk. Performance may be affected as a result.

Migrating sequential-access storage pools

You can set up migration thresholds for sequential-access storage pools. Migrating data from one sequential-access storage pool to another might be appropriate in some cases, for example, when you install a tape drive that uses a different type of tape and want to move data to that tape.

You probably will not want the server to migrate sequential-access storage pools on a regular basis. An operation such as tape-to-tape migration has limited benefits compared to disk-to-tape migration, and requires at least two tape drives.

You can migrate data from a sequential-access storage pool only to another sequential-access storage pool. You cannot migrate data from a sequential-access

storage pool to a disk storage pool. If you need to move data from a sequential-access storage pool to a disk storage pool, use the MOVE DATA command. See “Moving data from one volume to another volume” on page 349.

To control the migration process, set migration thresholds and migration delays for each storage pool using the DEFINE STGPOOL and UPDATE STGPOOL commands. You can also specify multiple concurrent migration processes to better use your available tape drives or FILE volumes. (For details, see “Specifying multiple concurrent migration processes” on page 261.) Using the MIGRATE STGPOOL command, you can control the duration of the migration process and whether reclamation is attempted prior to migration. For additional information, see “Starting migration manually or in a schedule” on page 261.

Tip: Data in storage pools that have an NDMP format (NETAPPDUMP, CELERRADUMP, or NDMPDUMP) cannot be migrated. However, in primary storage pools that have an NDMP format, you can make space available by using the MOVE DATA command. The target storage pool must have the same data format as the source storage pool.

How the server migrates files from sequential-access storage pools

The server migrates files by volume from sequential-access storage pools. Volumes that exceed the reclamation threshold are migrated first. Files in the least frequently referenced volumes are migrated next. Before files are migrated, the server checks the migration delay for the storage pool.

For tape and optical storage pools, the server begins the migration process when the ratio of volumes containing data to the total number of volumes in the storage pool, including scratch volumes, reaches the high migration threshold. For sequential-access disk (FILE) storage pools, the server starts the migration process when the ratio of data in a storage pool to the pool's total estimated data capacity reaches the high migration threshold. The calculation of data capacity includes the capacity of all the scratch volumes specified for the pool.

Tip: When Tivoli Storage Manager calculates the capacity for a sequential-access disk storage pool, it takes into consideration the amount of disk space available in the file system. For this reason, be sure that you have enough disk space in the file system to hold all the defined and scratch volumes specified for the storage pool. For example, suppose that the capacity of all the scratch volumes specified for a storage pool is 10 TB. (There are no predefined volumes.) However, only 9 TB of disk space is available in the file system. The capacity value used in the migration threshold is 9 TB, not 10 TB. If the high migration threshold is set to 70%, migration will begin when the storage pool contains 6.3 TB of data, not 7 TB.

When migrating files by volume from sequential-access storage pools, including sequential-access disk storage pools associated with a FILE device class, the server performs the following procedure:

1. The server first reclaims volumes that have exceeded the reclamation threshold. Reclamation is a server process of consolidating files from several volumes onto one volume. (See “Reclaiming space in sequential-access storage pools” on page 319.)
2. After reclamation processing, the server compares the space used in the storage pool to the low migration threshold.

3. If the space used is now below the low migration threshold, the server stops processing. If the space used is still above the low migration threshold, the server determines which volume is the least recently referenced volume.
4. If the amount of time a file has been in the storage pool exceeds the amount of time specified as the migration delay for the storage pool, the file is eligible for migration. The server selects the volume for migration only when all files on the volume are eligible for migration.
5. The server repeats steps 3 and 4 until the storage pool reaches the low migration threshold.

Because migration delay can prevent volumes from being migrated, the server can migrate files from all eligible volumes but still find that the storage pool is above the low migration threshold. If you set migration delay for a pool, you need to decide what is more important: either ensuring that files stay in the storage pool for as long as the migration delay, or ensuring there is enough space in the storage pool for new files. For each storage pool that has a migration delay set, you can choose what happens as the server tries to move enough files out of the storage pool to reach the low migration threshold. If the server cannot reach the low migration threshold by migrating only volumes that meet the migration delay requirement, you can choose one of the following:

- Allow the server to migrate volumes from the storage pool even if they do not meet the migration delay criteria (MIGCONTINUE=YES). This is the default. Allowing migration to continue ensures that space is made available in the storage pool for new files that need to be stored there.
- Have the server stop migration without reaching the low migration threshold (MIGCONTINUE=NO). Stopping migration ensures that volumes are not migrated for the time you specified with the migration delay. The administrator must ensure that there is always enough space available in the storage pool to hold the data for the required number of days.

Migration criteria for sequential-access storage pools

If you are planning to use migration for sequential-access storage pools, you need to consider a number of factors, including the time required to mount tapes into drives and whether collocation is enabled.

When defining migration criteria for sequential-access storage pools, consider:

- The capacity of the volumes in the storage pool
- The time required to migrate data to the next storage pool
- The speed of the devices that the storage pool uses
- The time required to mount media, such as tape volumes, into drives
- Whether operator presence is required
- The number of concurrent migration processes

If you decide to migrate data from one sequential-access storage pool to another, ensure that:

- Two drives (mount points) are available, one in each storage pool.
- The access mode for the next storage pool in the storage hierarchy is set to read/write.

For information about setting an access mode for sequential-access storage pools, see “Defining storage pools” on page 227.

- Collocation is set the same in both storage pools. For example, if collocation is set to NODE in the first storage pool, then collocation should be set to NODE in the next storage pool.

When you enable collocation for a storage pool, the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. For information about collocation for sequential-access storage pools, see “Keeping client files together using collocation” on page 310.

- You have sufficient resources (for example, staff) available to manage any necessary media mount and dismount operations. (This is especially true for multiple concurrent processing. For details, see “Specifying multiple concurrent migration processes.”) More mount operations occur because the server attempts to reclaim space from sequential-access storage pool volumes before it migrates files to the next storage pool.

If you want to limit migration from a sequential-access storage pool to another storage pool, set the high-migration threshold to a high percentage, such as 95%.

For information about setting a reclamation threshold for tape storage pools, see “Reclaiming space in sequential-access storage pools” on page 319.

There is no straightforward way to selectively migrate data for a specific node from one sequential storage pool to another. You can use the MOVE NODEDATA command to move file spaces for a node from one storage pool to another. See “Moving data belonging to a client node” on page 354.

Starting migration manually or in a schedule

To gain more control over how and when the migration process occurs, you can use the MIGRATE STGPOOL command. Issuing this command starts migration from one storage pool to the next storage pool in the hierarchy, regardless of the value of the HIGHMIG parameter of the storage pool definition.

You can specify the maximum number of minutes the migration will run before automatically cancelling. If you prefer, you can include this command in a schedule to perform migration when it is least intrusive to normal production needs.

For example, to migrate data from a storage pool named ALTPPOOL to the next storage pool, and specify that it end as soon as possible after one hour, issue the following command:

```
migrate stgpool altpool duration=60
```

Do not use this command if you are going to use automatic migration. To prevent automatic migration from running, set the HIGHMIG parameter of the storage pool definition to 100. For details about the MIGRATE STGPOOL command, refer to the *Administrator's Reference*.

Specifying multiple concurrent migration processes

Running multiple migration processes concurrently lets you make better use of your available tape drives or FILE volumes. When calculating the number of concurrent processes to run, you must carefully consider available resources.

Each migration process requires at least two simultaneous volume mounts (at least two mount points) and, if the device type is not FILE, at least two drives. One of the drives is for the input volume in the storage pool from which files are being migrated. The other drive is for the output volume in the storage pool to which files are being migrated.

When calculating the number of concurrent processes to run, carefully consider the resources you have available, including the number of storage pools that will be

involved with the migration, the number of mount points, the number of drives that can be dedicated to the operation, and (if appropriate) the number of mount operators available to manage migration requests. 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 storage pools that are involved in the migration. For more information about mount limit, see:

“Controlling the number of simultaneously mounted volumes” on page 202

For example, suppose that you want to migrate data on volumes in two sequential storage pools simultaneously and that all storage pools involved have the same device class. Each process requires two mount points and, if the device type is not FILE, two drives. To run four migration processes simultaneously (two for each storage pool), you need a total of at least eight mount points and eight drives if the device type is not FILE. The device class must have a mount limit of at least eight.

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 indefinitely or until the other migration processes complete and mount points or drives become available.

To specify one or more migration processes for each primary sequential-access storage pool, use the MIGPROCESS parameter on the DEFINE STGPOOL and UPDATE STGPOOL commands.

The Tivoli Storage Manager server starts 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.

Multiple concurrent migration processing does not affect collocation. If you specify collocation and multiple concurrent processes, the Tivoli Storage Manager server attempts to migrate the files for each collocation group, client node, or client file space onto as few volumes as possible. If files are collocated by group, each process can migrate only one group at a single time. In addition, if files belonging to a single collocation group (or node or file space) are on different volumes and are being migrated at the same time by different processes, the files could be migrated to separate output volumes.

If simultaneous-write operations during migration are enabled during multiple concurrent-migration processing, each process has the following requirements:

- A mount point
- A volume for each copy storage pool and active-data pool that is defined to the target storage pool and the primary pool

For details about the simultaneous-write function, see “Writing data simultaneously to primary, copy, and active-data pools” on page 285.

The effect of migration on copy storage pools and active-data pools

Files in copy storage pools and active-data pools cannot be migrated. Migration of files between primary storage pools does not affect copy storage pool files or active-data pool files. Neither copy storage pool files nor active-data pool files move when primary storage pool files move.

For example, suppose a copy of a file is made while it is in a disk storage pool. The file then migrates to a primary tape storage pool. If you then back up the primary tape storage pool to the same copy storage pool, a new copy of the file is not needed. The server knows it already has a valid copy of the file.

The only way to store files in copy storage pools is by backing up (the `BACKUP STGPOOL` command) or by using the simultaneous-write function. The only way to store files in active-data pools is by copying active data (the `COPY ACTIVEDATA` command) or by using the simultaneous-write function.

Caching in disk storage pools

When cache is enabled, the migration process leaves behind duplicate copies of files after the server migrates these files to the next storage pool in the storage hierarchy. Using cache can improve the speed with which the server retrieves some files. Consider enabling cache for space-managed files that are frequently accessed by clients.

If space is needed to store new data in the disk storage pool, cached files are erased and the space they occupied is used for the new data.

Using cache has some important disadvantages:

- Using cache can increase the time required for client backup operations to complete. Performance is affected because, as part of the backup operation, the server must erase cached files to make room for storing new files. The effect can be severe when the server is storing a very large file and must erase cached files. For the best performance for client backup operations to disk storage pools, do not use cache.
- Using cache can require more space for the server database. When you use cache, more database space is needed because the server has to keep track of both the cached copy of the file and the new copy in the next storage pool.
- If you want to use caching, you cannot also enable shredding for that disk storage pool. See “Securing sensitive client data” on page 499 for more information about shredding.

When cache is disabled and migration occurs, the server migrates the files to the next storage pool and erases the files from the disk storage pool. By default, the system disables caching for each disk storage pool because of the potential effects of cache on backup performance. If you leave cache disabled, consider higher migration thresholds for the disk storage pool. A higher migration threshold keeps files on disk longer because migration occurs less frequently.

If fast restores of active client data is your objective, you can also use active-data pools, which are storage pools containing only active versions of client backup data. For details, see “Active-data pools” on page 223.

To enable cache, specify `CACHE=YES` when defining or updating a storage pool.

How the server removes cached files

When space is needed, the server reclaims space occupied by cached files. Files that have the oldest retrieval date are overwritten first.

For example, assume that two files, File A and File B, are cached files that are the same size. If File A was last retrieved on 05/16/08 and File B was last retrieved on 06/19/08, then File A is deleted to reclaim space first.

If you do not want the server to update the retrieval date for files when a client restores or retrieves the file, specify the server option `NORETRIEVEDATE` in the server options file. If you specify this option, the server removes copies of files in cache regardless how recently the files were retrieved.

Effect of caching on storage pool statistics

The space-utilization statistic for the pool (Pct Util) includes the space used by any cached copies of files in the storage pool. The migratable-data statistic (Pct Migr) does not include space occupied by cached copies of files.

The server compares the migratable-data statistic with migration-threshold parameters to determine when migration should begin or end. For more information about storage pool statistics, see “Monitoring storage-pool and volume usage” on page 332.

Deduplicating data

Data deduplication is a method for eliminating redundant data. Only one instance of the data is retained on storage media, such as disk or tape. Other instances of the same data are replaced with a pointer to the retained instance.

Deduplicated data must be stored in sequential-access disk (FILE) primary, copy, and active-data storage pools. that you enable for data deduplication. Because you can store more data on disk than on tape, data deduplication can reduce the overall amount of time that is required to retrieve data.

Restriction: When a client backs up or archives a file, the data is written to the primary storage pool specified by the copy group of the management class that is bound to the data. To deduplicate the client data, the primary storage pool must be a sequential-access disk (FILE) storage pool that is enabled for data deduplication.

Data deduplication location

In server-side data deduplication, processing takes place exclusively on the server. In client-side data deduplication, the processing is distributed between the server and the backup-archive client.

The ability to deduplicate data on either the backup-archive client or the server provides flexibility in terms of resource utilization, policy management, and security. You can also combine both client-side and server-side data deduplication in the same production environment. For example, you can specify certain nodes for client-side data deduplication and certain nodes for server-side data deduplication. You can store the data for both sets of nodes in the same deduplicated storage pool.

Server-side data deduplication is available only with IBM Tivoli Storage Manager V6.1 or later servers. For optimal efficiency when using server-side data

deduplication, upgrade to the V6.1 or later backup-archive client. Client-side data deduplication is available only with servers, V6.2 or later, and backup-archive clients, V6.2 or later. Backup-archive clients that can deduplicate data can also access data that was deduplicated by server-side processes. Similarly, data that was deduplicated by client-side processes can be accessed by the server.

You enable client-side data deduplication using a combination of settings on the client node and the server. See Table 28.

Table 28. Data deduplication settings: Client and server

Value of the DEDUPLICATION parameter for the REGISTER NODE or UPDATE NODE server commands	Value of the client DEDUPLICATION option	Data deduplication location
SERVERONLY	Yes	Server The Yes value of the client option is ignored by the server.
SERVERONLY	No	Server
CLIENTORSERVER	Yes	Client
CLIENTORSERVER	No	Server

You can set the DEDUPLICATION option in the client options file, in the preference editor of the Tivoli Storage Manager client GUI, or in the client option set on the Tivoli Storage Manager server. Use the DEFINE CLIENTOPT command to set the DEDUPLICATION option in a client option set. To prevent the client from overriding the value in the client option set, specify **FORCE=YES**.

Server-side data deduplication: Overview

Server-side data deduplication is a two-phase process. In the first phase, the server identifies duplicate data. In the second phase, duplicate data is removed by certain server processes.

In addition to whole files, IBM Tivoli Storage Manager can also deduplicate parts of files that are common with parts of other files. Data becomes eligible for duplicate identification as volumes in the storage pool are filled. A volume does not have to be full before duplicate identification starts.

Duplicate data is removed by one of the following processes:

- Reclaiming volumes in the primary storage pool, copy storage pool, or active-data pool
- Backing up a primary storage pool to a copy storage pool that is also set up for data deduplication
- Copying active data in the primary storage pool to an active-data pool that is also set up for data deduplication
- Migrating data from the primary storage pool to another primary storage pool that is also set up for data deduplication
- Moving data from the primary storage pool to a different primary storage pool that is also set up for data deduplication
- Moving data within the same copy storage pool or moving data within the same active-data pool

Related concepts

“Client-side data deduplication: Overview”

Client-side data deduplication: Overview

In client-side data deduplication, the backup-archive client and the server work together to identify duplicate data.

Benefits

Client-side data deduplication provides several advantages:

- It can reduce the amount of data that is sent over the local area network (LAN).
- The processing power that is required to identify duplicate data is offloaded from the server to client nodes. Server-side data deduplication is always enabled for deduplication-enabled storage pools. However, files that are in the deduplication-enabled storage pools and that were deduplicated by the client, do not require additional processing.
- The processing power that is required to remove duplicate data on the server is eliminated, allowing space savings on the server to occur immediately.

Client-side data deduplication has a possible disadvantage. The server does not have whole copies of client files *until* you back up the primary storage pools that contain client extents to a non-deduplicated copy storage pool. (*Extents* are parts of a file that are created during the data-deduplication process.) During storage pool backup to non-deduplicated storage pool, client extents are reassembled into contiguous files.

Server-side data deduplication offers more protection against data loss. By default, primary sequential-access storage pools that are set up for data deduplication must be backed up to non-deduplicated copy storage pools before they can be reclaimed and before duplicate data can be removed. The default ensures that the server has copies of whole files at all times, in either a primary storage pool or a copy storage pool.

Client-side data-deduplication process

Client-side data deduplication is a three-phase process:

1. The client creates extents.
2. The client and server work together to identify duplicate extents.
3. The client sends non-duplicate extents to the server.

Subsequent client data-deduplication operations create new extents. Some or all of those extents might match the extents that were created in previous data-deduplication operations and sent to the server. Matching extents are not sent to the server again.

Prerequisites

The following prerequisites apply to client-side data deduplication:

- When a client backs up or archives a file, the data is written to the primary storage pool that is specified by the copy group of the management class that is bound to the data. To deduplicate the client data, the primary storage pool must be a sequential-access disk (FILE) storage pool that is enabled for data deduplication.

- The value of the DEDUPLICATION option on the client must be set to YES. You can set the DEDUPLICATION option in the client options file, in the preference editor of the IBM Tivoli Storage Manager client GUI, or in the client option set on the Tivoli Storage Manager server. Use the DEFINE CLIENTOPT command to set the DEDUPLICATION option in a client option set. To prevent the client from overriding the value in the client option set, specify **FORCE=YES**.
- Client-side data deduplication must be enabled on the server. To enable client-side data deduplication, use the **DEDUPLICATION** parameter on the REGISTER NODE or UPDATE NODE server command. Set the value of the parameter to CLIENTORSERVER.
- Files on the client must not be excluded from client-side data deduplication. By default, all files are included. You can optionally exclude specific files from data deduplication.
- Files on the client must not be encrypted. Encrypted files and files from encrypted file systems cannot be deduplicated.
- Files must be more than 2 KB. Files that are 2 KB or less are not deduplicated.

Restoring or retrieving files

When restoring or retrieving files, the client node queries for and displays files as it typically does. If a user selects a file that exists in a deduplicated storage pool, the server reconstructs the file.

Enhancements

With client-side data deduplication, you can:

- Exclude specific files on a client from data deduplication.
- Enable a data deduplication cache that reduces network traffic between the client and the server. The cache contains extents that were sent to the server in previous incremental backup operations. Instead of querying the server for the existence of an extent, the client queries its cache.
Specify a size and location for a client cache. If an inconsistency between the server and the local cache is detected, the local cache is removed and repopulated.
- Enable both client-side data deduplication and compression to reduce the amount of data that is stored by the server. Each extent is compressed before being sent to the server. The trade-off is between storage savings and the processing power that is required to compress client data. In general, if you compress and deduplicate data on the client system, you are using approximately twice as much processing power as data deduplication alone. The server can work with deduplicated, compressed data. In addition, backup-archive clients earlier than V6.2 can restore deduplicated, compressed data.

Client-side data deduplication and next storage pools

If client-side data deduplication is enabled and the primary destination storage pool is full, but there is a next storage pool in the hierarchy, the server stops the transaction. Client-side data deduplication is disabled, and the client tries the transaction again with files that are not deduplicated.

If the backup operation is successful and if the next storage pool is enabled for data deduplication, the files are deduplicated by the server. If the next storage pool is not enabled for data deduplication, the files are not deduplicated.

To ensure that client-side data deduplication proceeds without interruption, maintain sufficient free storage in your primary destination storage pool.

For details about client-side data deduplication, including options for controlling deduplication, see the *Backup-Archive Clients Installation and User's Guide*.

LAN-free access to storage pools containing client-side deduplicated data

Only V6.2 storage agents can use LAN-free data movement to access storage pools that contain data that was deduplicated by clients. V6.1 storage agents or earlier cannot use LAN-free data movement to access these storage pools. Using a V6.1 storage agent or earlier to access a storage pool that contains client-side deduplicated data causes restore operations and retrieve operations to go over the LAN. See Table 29.

Table 29. Paths for data movement

	Storage pool contains only client-side deduplicated data	Storage pool contains a mixture of client-side and server-side deduplicated data	Storage pool contains only server-side deduplicated data
V6.1 or earlier storage agent	Over the LAN	Over the LAN	LAN-free
V6.2 storage agent	LAN-free	LAN-free	LAN-free

V6.2 backup-archive clients are compatible with V6.2 storage agents, and provide LAN-free access to storage pools that contain client-side deduplicated data.

As part of the planning process, decide whether you want to use LAN-free data movement and whether you want to use client-side data deduplication, server-side deduplication, or both. If you decide to use LAN-free data movement and both client-side and server-side data deduplication, take one of the following steps:

- For V6.1 or earlier storage agents, store client-side deduplicated data in a separate storage pool. Restore and retrieve deduplicated data from this storage pool over the LAN. Use LAN-free data movement to restore and retrieve data from storage pools that contain data that was deduplicated only by the server.
- Upgrade to V6.2 storage agents. Upgrading to V6.2 storage agents provides LAN-free access to any storage pool that contains client-side deduplicated data, server-side deduplicated data, or both.

Related concepts

"Server-side data deduplication: Overview" on page 265

Data deduplication limitations

Before implementing data deduplication, be aware that certain limitations apply.

Version support

Server-side data deduplication is available only with IBM Tivoli Storage Manager V6.1 or later servers. For optimal efficiency when using server-side data deduplication, upgrade to the backup-archive client V6.1 or later.

Client-side data deduplication is available only with Tivoli Storage Manager V6.2 or later servers and backup-archive clients V6.2 or later.

Eligible storage pools

Data on random-access disk or on tape cannot be deduplicated. Only data in storage pools that are associated with sequential-access disk devices (FILE) can be deduplicated. You must enable FILE storage pools for data deduplication.

Client files must be bound to a management class that specifies a deduplication-enabled storage pool.

Encrypted files

The Tivoli Storage Manager server and the backup-archive client cannot deduplicate encrypted files. If an encrypted file is encountered during data deduplication processing, the file is not deduplicated, and a message is logged.

Tip: You do not have to process encrypted files separately from files that are eligible for client-side data deduplication. Both types of files can be processed in the same operation. However, they are sent to the server in different transactions.

As a security precaution, you can take one or more of the following steps:

- Enable storage-device encryption together with client-side data deduplication.
- Use client-side data deduplication only for nodes that are secure.
- If you are uncertain about network security, enable Secure Sockets Layer (SSL).
- If you do not want certain objects (for example, image objects) to be processed by client-side data deduplication, you can exclude them on the client. If an object is excluded from client-side data deduplication and it is sent to a storage pool that is set up for data deduplication, the object is deduplicated on server.
- Use the SET DEDUPVERIFICATIONLEVEL command to detect possible security attacks on the server during client-side data deduplication. Using this command, you can specify a percentage of client extents for the server to verify. If the server detects a possible security attack, a message is displayed.

File size

Only files that are more than 2 KB are deduplicated. Files that are 2 KB or less are not deduplicated.

Operations that preempt client-side data deduplication

The following operations take precedence over client-side data deduplication:

- LAN-free data movement
- Subfile backup operations

- Simultaneous-write operations

Do not schedule or enable any of those operations during client-side data deduplication. If any of those operations occur during client-side data deduplication, client-side data deduplication is turned off, and a message is issued to the error log.

Data deduplication of hierarchical storage management data

HSM data from UNIX and Linux clients is ignored by client-side data deduplication. Server-side deduplication of HSM data from UNIX and Linux clients is allowed.

Collocation

You can use collocation for storage pools that are set up for data deduplication. However, collocation might not have the same benefit as it does for storage pools that are not set up for data deduplication.

By using collocation with storage pools that are set up for data deduplication, you can control the placement of data on volumes. However, the physical location of duplicate data might be on different volumes. No-query-restore, and other processes remain efficient in selecting volumes that contain non-deduplicated data. However, the efficiency declines when additional volumes are required to provide the duplicate data.

Related tasks

“Keeping client files together using collocation” on page 310

“Detecting possible security attacks on the server during client-side deduplication” on page 271

Planning guidelines for data deduplication

Planning for data deduplication is important because there are many factors, such as data deduplication location and storage pool setup, to consider. A set of guidelines is provided to structure your planning activities.

As part of planning, you must make the following decisions:

- Determine which client nodes have data that you want to deduplicate.
- Determine whether you want to implement server-side data deduplication, client-side data deduplication, or a combination of both.
- If you choose client-side data deduplication, decide what, if any, security precautions to take. Steps that you can take to protect data and the server include:
 - Specifying client-side data deduplication for only nodes that are secure
 - Enabling Secure Sockets Layer (SSL)
 - Excluding certain client files from data deduplication
 - Using the SET DEDUPVERIFICATIONLEVEL command to detect possible security attacks on the server during client-side data deduplication
 - Using storage-device encryption together with client-side data deduplication
- Decide whether you want to define a new storage pool exclusively for data deduplication or update an existing storage pool. The storage pool must be a sequential-access disk (FILE) pool. Data deduplication occurs at the storage-pool level, and all data within a storage pool, except encrypted data, is deduplicated.

- If you want to implement server-side data deduplication, decide how best to control duplicate-identification processes. For example, you might want to run duplicate-identification processes automatically all the time. Alternatively, you might want to start and stop duplicate-identification processes manually. You can also start duplicate-identification processes automatically and then increase or decrease the number of processes depending on your server workload. Whatever you decide, you can always change the settings later, after the initial setup, to meet the requirements of your operations.

The following table can help in the planning process.

Table 30. Options for controlling duplicate-identification processes

If you create a storage pool for data deduplication...	If you update an existing storage pool...
<p>You can specify 1 - 20 duplicate-identification processes to start automatically. The IBM Tivoli Storage Manager server does not start any processes if you specify zero.</p> <p>If you are creating a primary sequential-access storage pool and you do not specify a value, the server starts one process automatically. If you are creating a copy storage pool or an active-data pool and you do not specify a value, the server does not start any processes automatically.</p> <p>After the storage pool has been created, you can increase and decrease the number of duplicate-identification processes manually. You can also start, stop, and restart duplicate-identification processes manually.</p>	<p>You can specify 0 - 20 duplicate-identification processes to start automatically. If you do not specify any duplicate-identification processes, you must start and stop processes manually.</p> <p>The Tivoli Storage Manager server does not start any duplicate-identification processes automatically by default.</p>

- Decide whether to define or update a storage pool for data deduplication, but not actually perform data deduplication. For example, suppose that you have a primary sequential-access disk storage pool and a copy sequential-access disk storage pool. Both pools are set up for data deduplication. You might want to run duplicate-identification processes for only the primary storage pool. In this way, only the primary storage pool reads and deduplicates data. However, when the data is moved to the copy storage pool, the data deduplication is preserved, and no duplicate identification is required.

Related tasks

“Detecting possible security attacks on the server during client-side deduplication”

Detecting possible security attacks on the server during client-side 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, setting of the node **DEDUPLICATION** parameter 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 the client-side data deduplication feature 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.

To detect a possible security attack when client-side data deduplication is enabled, issue the SET DEDUPVERIFICATIONLEVEL command. Specify an integer value 1 - 100 to indicate the percentage of client extents to be verified. The default value is 0. This value indicates that no extents are verified.

Tip: Verifying extents consumes processing power and adversely affects server performance. For optimal performance, do not specify values greater than 10 for the SET DEDUPVERIFICATIONLEVEL command. Other methods for protecting the server include:

- Enabling client-side data deduplication only for clients that are secure. If you choose this method, do not change the default setting of SET DEDUPVERIFICATIONLEVEL command.
- Creating automated scripts to enable client-side data deduplication only during certain time periods.
- Deduplicating data using only server-side data deduplication. Server-side data deduplication does not expose the server to security attacks from the client.

To display the current value for SET DEDUPVERIFICATIONLEVEL, issue the QUERY STATUS command. Check the value in the **Client-side Deduplication Verification Level** field.

Evaluating data deduplication in a test environment

Testing can give you important information about the possible benefits of server-side and client-side data deduplication in your production environment. Space savings and restore-and-retrieve times are two key indicators that you can test.

Restore and retrieve operations from server-side and client-side deduplicated storage pools

Restore-and-retrieve operations from a sequential-access disk (FILE) storage pool that is set up for data deduplication have different performance characteristics than restore-and-retrieve operations from a FILE storage pool that is not set up for data deduplication. To ensure that performance objectives can be met, test your restore scenarios.

In a FILE storage pool that is not set up for data deduplication, files on a volume that are being restored or retrieved are read sequentially from the volume before the next volume is mounted. This process ensures optimal I/O performance and eliminates the need to mount a volume multiple times.

In a FILE storage pool that is set up for data deduplication, however, extents that comprise a single file can be distributed across multiple volumes. To restore or retrieve the file, each volume containing a file extent must be mounted. As a result, the I/O is more random, which can lead to slower restore-and-retrieve times. These results occur more often with small files that are less than 100 KB. In

addition, more processor resources are consumed when restoring or retrieving from a deduplicated storage pool. The additional consumption occurs because the data is checked to ensure that it has been reassembled properly.

Although small-file, restore-and-retrieve operations from a deduplicated storage pool might be relatively slow, these operations are still typically faster than small-file restore-and-retrieve operations from tape because of the added tape mount-and-locate time. If you have data for which fastest-possible restore-and-retrieval time is critical, you can use a sequential-access disk storage pool that is not set up for data deduplication.

Tips:

- To reduce the mounting and removing of FILE storage pool volumes, the server allows for multiple volumes to remain mounted until they are no longer needed. The number of volumes that can be mounted at a time is controlled by the NUMOPENVOLSALLOWED option.
- For optimal efficiency when deduplicating, upgrade to the backup-archive client version 6.1 or version 6.2.

Related tasks

“Improving performance when reading from deduplicated storage pools” on page 284

Estimating space savings from server-side data deduplication

Before setting up data deduplication in your production environment, you can estimate the amount of storage space that can be saved. Directions are provided for backing up the data in a primary storage pool to a temporary copy storage pool that is set up for data deduplication.

To estimate space savings:

1. Create a sequential-access disk (FILE) copy storage pool and enable the pool for data deduplication.
2. Back up the contents of the primary storage pool that you want to test to the copy storage pool.
3. Run the duplicate-identification processes against the volumes in the copy storage pool.

If you specified one or more duplicate-identification processes when you created the copy storage pool, those processes start automatically. If you did not specify any processes, you must specify and start duplicate-identification processes manually.

4. After all the data in the copy storage pool is identified, start reclamation by changing the reclamation percentage on the copy storage pool to 1%.
5. When reclamation finishes, use the QUERY STGPOOL command to check the copy storage-pool statistics to determine the amount of space that was saved.

If the results are satisfactory, complete one of the following tasks:

- If the primary storage pool is a sequential-access disk storage pool, update the storage, specifying data deduplication.
- If the primary storage pool is not a sequential-access disk storage pool, create a new primary sequential-access disk storage pool, specifying data deduplication. Move the data or migrate the data from the original storage pool to the new storage pool.

Managing deduplication-enabled storage pools

You can create a storage pool for data deduplication or you can upgrade an existing storage pool. If you are implementing sever-side data deduplication, IBM Tivoli Storage Manager provides the option of running duplicate-identification processes automatically or manually.

Before setting up a storage pool:

- Determine which client nodes have data that you want to deduplicate. Decide whether you want to deduplicate data on a node-by-node basis, on either the client or the server.
- Decide whether you want to define a new storage pool exclusively for data deduplication or update an existing storage pool. If you update a storage pool for data deduplication, Tivoli Storage Manager deduplicates the data that has already been stored. No additional backup, archive, or migration is required. You can also define or update a storage pool for data deduplication, but not actually deduplicate data.
- Decide how you want to control duplicate-identification processes.

You can create a storage pool for data deduplication or update an existing storage pool for data deduplication. To set up a storage pool for data deduplication:

- If you are defining a new storage pool:
 1. Use the `DEFINE STGPPOOL` command and specify the `DEDUPLICATE=YES` parameter.
 2. Define a new policy domain to direct eligible client-node data to the storage pool.
- If you are updating an existing storage pool:
 1. Determine whether the storage pool contains data from one or more client nodes that you want to exclude from data deduplication. If it does:
 - a. Using the `MOVE DATA` command, move the data belonging to the excluded nodes from the storage pool to be converted to another storage pool.
 - b. Direct data belonging to the excluded nodes to the other storage pool. The easiest way to complete this task is to create another policy domain and designate the other storage pool as the destination storage pool.
 2. Change the storage-pool definition using the `UPDATE STGPPOOL` command. Specify the `DEDUPLICATE` and `NUMPROCESSES` parameters.

As data is stored in the pool, the duplicates are identified. When the reclamation threshold for the storage pool is reached, reclamation begins, and the space that is occupied by duplicate data is reclaimed.

In the storage pool definition, you can specify as many as 20 duplicate-identification processes to start automatically. If you do not specify any duplicate-identification processes in the storage pool definition, you must control data deduplication manually. Duplicate identification requires extra disk I/O and processor resources. To mitigate the effects on server workload, you can manually increase or decrease the number of duplicate-identification processes, as well as their duration.

Attention: By default, the Tivoli Storage Manager server requires that you back up deduplication-enabled primary storage pools before volumes in the storage pool are reclaimed and before duplicate data is discarded. The copy storage pools and active-data pools to which you back up data and copy active data must not be set up for data deduplication. To prevent possible data loss, do not change the default. If you do change the default, reclamation criteria remains unchanged.

Protecting data in primary storage pools set up for data deduplication

By default, primary sequential-access storage pools that are set up for data deduplication must be backed up to a copy storage pool before they can be reclaimed and duplicate data can be removed. To minimize the potential of data loss, do not change the default setting.

To protect the data in primary storage pools, issue the `BACKUP STGPOOL` command to copy the data to copy storage pools. Ensure that the copy storage pools are *not* configured for data deduplication.

During storage pool backup to a non-deduplicated storage pool, server-side and client side extents are reassembled into contiguous files.

Copying active data to an active-data pool does not qualify as a valid backup for protecting data. Data must be backed up to a copy storage pool that is not set up for data deduplication.

Attention: You can change the default setting to permit reclamation of primary storage pools that are not backed up. However, there is a remote possibility that changing the default can result in unrecoverable data loss if a data-integrity error occurs. To change the default and permit reclamation of primary sequential-access storage pools that are not backed up, set the value of the `DEDUPREQUIRESBACKUP` server option to `NO`. Changing the default does not change the reclamation criteria that you specified for a storage pool.

The `DEDUPREQUIRESBACKUP` server option applies only to primary storage pools. The option does not apply to copy storage pools or active-data pools.

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 was 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.

Effects on data deduplication when moving or copying data

You can move or copy data between storage pools regardless of whether they are set up for data deduplication.

The following table illustrates what happens to data deduplication when data objects are moved or copied.

Table 31. Effects when moving or copying data

If the source storage pool is...	...and you move or copy data to a target storage pool that is...	The result is...
Set up for data deduplication	Set up for data deduplication	All data objects in the source pool are examined for existence in the target pool. If an object exists in the target pool, information about data deduplication is preserved so that the data does not need to be deduplicated again. If an object does not exist in the target pool, it is moved or copied.
	Not set up for data deduplication	The data is not deduplicated in the target storage pool. This rule applies to any type of storage pool, including storage pools that use virtual volumes.
Not set up for data deduplication	Set up for data deduplication	Normal data deduplication processing takes place after the data is moved or copied.
	Not set up for data deduplication	No data deduplication occurs.

Turning data deduplication off for a storage pool

If you turn data deduplication off for a storage pool by updating the storage pool definition, new data that enters the storage pool is not deduplicated.

Deduplicated data, which was in the storage pool before you turned off data deduplication, is not reassembled. Deduplicated data continues to be removed due to normal reclamation and deletion. All information about data deduplication for the storage pool is retained.

To turn off data deduplication for a storage pool, use the UPDATE STGPOOL command and specify DEDUPLICATE=NO.

If you turn data deduplication on for the same storage pool, duplicate-identification processes resume, skipping any files that have already been processed. You can change the number of duplicate-identification processes. When calculating the number of duplicate-identification processes to specify, consider the workload on the server and the amount of data requiring data deduplication.

Controlling data deduplication

If client files are bound to a management class that specifies a deduplication-enabled storage pool, the files are, by default, deduplicated on the server. Client-side data deduplication is enabled using a combination of settings on the client and the server.

The following table shows how the data deduplication settings on the client interact with the data deduplication settings on the Tivoli Storage Manager server.

Table 32. Data deduplication settings: Client and server

Value of the DEDUPLICATION parameter for REGISTER NODE or UPDATE NODE	Value of the client DEDUPLICATION option in the client options file	Data deduplication location
SERVERONLY	Yes	Server The Yes value of the client option is ignored by the server.
CLIENTORSERVER	Yes	Client
CLIENTORSERVER	No	Server
SERVERONLY	No	Server

You can set the DEDUPLICATION option in the client options file, in the preference editor of the Tivoli Storage Manager client GUI, or in the client option set on the Tivoli Storage Manager server. Use the DEFINE CLIENTOPT command to set the DEDUPLICATION option in a client option set. To prevent the client from overriding the value in the client option set, specify **FORCE=YES**.

Controlling server-side data deduplication

If client files are bound to a management class that specifies a deduplication-enabled storage pool, the files are, by default, deduplicated on the server.

To enable server-side data deduplication, specify SERVERONLY as the value of the DEDUPLICATION parameter on the REGISTER NODE or UPDATE NODE command.

If you specify SERVERONLY, the values of the client DEDUPLICATION option are ignored.

Related concepts

“Server-side data deduplication: Overview” on page 265

Controlling duplicate-identification processes:

In server-side data deduplication, client data is deduplicated on the server. When you define or update a storage pool for data deduplication, you can specify 0 - 20 duplicate-identification processes to start automatically and run indefinitely. To avoid resource impacts during server operations (for example, client backups), you can also control data deduplication processing manually.

For example, suppose you specify eight duplicate-identification processes in your storage pool definition. These processes start automatically and run indefinitely. However, you decide that you want to reduce the number of processes during client backups, which take 60 minutes. You can manually reduce the number of duplicate-identification processes to four and set a duration of 60 minutes. After the backup is complete, the IBM Tivoli Storage Manager server automatically restarts four processes so that the eight processes are running again.

Alternatively, you can identify duplicates manually. Specify 0 as the number of duplicate-identification processes in your storage pool definition. With this setting, the Tivoli Storage Manager server does not automatically start any duplicate-identification processes. Then, depending on your schedule and server workload, specify the number of duplicate-identification processes and their

duration for one or more storage pools.

Duplicate-identification processing states:

Duplicate-identification processes are different from other server processes. When other server processes finish a task, they end. When duplicate-identification processes finish processing available files, they go into an idle state.

Duplicate-identification processes can be either active or idle. Processes that are currently 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. Processes end only when canceled or when you change the number of duplicate-identification processes for the storage pool to a value less than the number that is currently specified.

The output of the QUERY PROCESS command for a duplicate-identification process includes the total number of bytes and files that have been processed since the process first started. For example, if a duplicate-identification process processes four files, idles, and then processes five more files, the total number of files processed is nine.

Interaction of manual data deduplication controls:

You can change the number of duplicate-identification processes used during server-side data deduplication. You can also change the length of time that processes are allowed to run. You can adjust these settings as often as you want.

Table 33 on page 279 shows how these two controls, the number and duration of processes, interact for a particular storage pool.

Remember:

- When the amount of time that you specify as a duration expires, the number of duplicate-identification processes always reverts to the number of processes specified in the storage pool definition.
- 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 value that you specify as a duration.
- To change the number of duplicate-identification processes, you can also update the storage pool definition using the UPDATE STGPOOL command. However, when you update a storage pool definition, you cannot specify a duration. The processes that you specify in the storage pool definition run indefinitely, or until you issue the IDENTIFY DUPLICATES command, update the storage pool definition again, or cancel a process.

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 33. Controlling duplicate-identification processes manually

Using the IDENTIFY DUPLICATES command, you specify...	...and a duration of...	The result is...
2 duplicate-identification processes	None specified	One duplicate-identification processes finishes the file it is working on, if any, and then stops. Two processes run indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.
	60 minutes	One duplicate-identification process finishes the file it is working on, if any, and then stops. After 60 minutes, the server starts one process so that three are running.
4 duplicate-identification processes	None specified	The server starts one duplicate-identification process. Four processes run indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.
	60 minutes	The server starts one duplicate-identification process. At the end of 60 minutes, one process finishes the file it is working on, if any, and then stops. The additional process started by this command might not be the one that stops when the duration has expired.
0 duplicate-identification processes	None specified	All duplicate-identification processes finish the files that they are working on, if any, and stop. This change lasts indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.
	60 minutes	All duplicate-identification processes finish the files that they are working on, if any, and stop. At the end of 60 minutes, the server starts three processes.
None specified	Not available	The number of duplicate-identification processes resets to the number of processes specified in the storage pool definition. This change lasts indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.

The following example illustrates how you can control data deduplication using a combination of automatic and manual duplicate-identification processes. Suppose you create two new storage pools for data deduplication, A and B. When you create the pools, you specify two duplicate-identification processes for A and one process for B. The IBM Tivoli Storage Manager server is set by default to run those processes automatically. As data is stored in the pools, duplicates are identified and marked for removal. When there is no data to deduplicate, the duplicate-identification processes go into an idle state, but remain active.

Suppose you want to avoid resource impacts on the server during client-node backups. You must reduce the number of duplicate-identification processes manually. For A, you specify a value of 1 for the number of duplicate-identification process. For B, you specify a value of 0. You also specify that these changes remain in effect for 60 minutes, the duration of your backup window.

Specifying these values causes two of the three running processes to finish the files on which they are working and to stop. One duplicate-identification process is now running for A. No duplicate-identification processes are running for B. After

60 minutes, the Tivoli Storage Manager server automatically resets the data-deduplication processes to the values specified in the storage pool definition. One process starts for A, for a total of two running processes. One process also starts for B.

Starting and stopping duplicate-identification processes:

In server-side data deduplication, you can start additional duplicate-identification processes and stop some or all active processes. You can also specify an amount of time that the change remains in effect. If you did not specify any duplicate-identification processes in the storage pool definition, you can start new processes and stop them manually.

To specify the number and duration of duplicate-identification processes for a storage pool, issue the IDENTIFY DUPLICATES command.

For example, suppose that you have four storage pools: stgpoolA, stgpoolB, stgpoolC, and stgpoolD. All the storage pools are associated with a particular IBM Tivoli Storage Manager server. Storage pools A and B are each running one duplicate-identification process, and storage pools C and D are each running two. A 60-minute client backup is scheduled to take place, and you want to reduce the server workload from these processes by two-thirds.

To accomplish this task, issue the following commands:

```
IDENTIFY DUPLICATES STGPOOLA DURATION=60 NUMPROCESS=0
IDENTIFY DUPLICATES STGPOOLB DURATION=60 NUMPROCESS=0
IDENTIFY DUPLICATES STGPOOLC DURATION=60 NUMPROCESS=1
IDENTIFY DUPLICATES STGPOOLD DURATION=60 NUMPROCESS=1
```

Now two processes are running for 60 minutes, one third of the number running before the change. At the end of 60 minutes, the Tivoli Storage Manager server automatically restarts one duplicate-identification process in storage pools A and B, and one process in storage pools C and D.

Controlling client-side data deduplication

The combination of client option and server parameter values determines whether data deduplication occurs on the client or the server. The default data deduplication setting is server-side data deduplication.

To enable client-side data deduplication, complete both of the following steps:

- Specify **YES** as the value of the **DEDUPLICATION** option for the client.
You can set the **DEDUPLICATION** option in the client options file, in the preference editor of the IBM Tivoli Storage Manager client GUI, or in the client option set on the Tivoli Storage Manager server. Use the **DEFINE CLIENTOPT** command to set the **DEDUPLICATION** option in a client option set. To prevent the client from overriding the value in the client option set, specify **FORCE=YES**.
- Specify **DEDUPLICATION=CLIENTORSERVER** on the **REGISTER NODE** or **UPDATE NODE** command.

For details about client-side data deduplication options, see the *Backup-Archive Clients Installation and User's Guide*.

Related concepts

“Client-side data deduplication: Overview” on page 266

Enabling client-side data deduplication for a single node:

In this example, you enable client-side data deduplication for a single node. You have a policy domain that you use to manage deduplicated data.

The name of the domain that you use to manage deduplicated data is dedupdomain1. The primary storage pool specified by the copy group of the default management class is a deduplication-enabled storage pool. The client, MATT, that you want to enable for data deduplication uses a default management class for backup operations.

To enable client-side data deduplication for a single node:

- On the server, assign client node MATT to dedupdomain1:

```
update node matt domain=dedupdomain1 deduplication=clientorserver
```

The setting of the **DEDUPLICATION** parameter must be CLIENTORSERVER

- Add the following option to the dsm.sys file:

```
deduplication yes
```

You can set the DEDUPLICATION option in the preference editor of the IBM Tivoli Storage Manager client GUI or in the client option set on the Tivoli Storage Manager server. If you set the DEDUPLICATION option in the client option set, you can also use the **FORCE** parameter to specify whether the server forces the client to use the value in the option set or whether the client can override the value.

To determine the amount of data that was deduplicated, start a backup or archive operation. At the end of the operation, check the backup or archive report.

Enabling client-side data deduplication for multiple client nodes:

In this example, you enable client-side data deduplication for more than one client node.

Complete the following steps on the server. No configuration is necessary on the client.

1. Assign all the client nodes to a domain (DEDUPDOMAIN1) that has a deduplication-enabled destination storage pool:

```
update node node1 domain=dedupdomain1 deduplication=clientorserver
...
update node noden domain=dedupdomain1 deduplication=clientorserver
```

You can automate this step with a script or a macro.

2. To enable client-side data deduplication, define a client option set:

```
define cloptset client_dedup
define clientopt client_dedup deduplication=yes yes force=yes
```

You can also add the following client-side data deduplication options to the client option set:

- ENABLEDEDUPCACHE
- DEDUPCACHEPATH
- DEDUPCACHESIZE
- INCLUDE.DEDUP
- EXCLUDE.DEDUP

3. Assign the client option set to the client nodes:

```
update node node1 cloptset=client_dedup
...
update node noden cloptset=client_dedup
```

You can automate this step with a script or a macro.

Changing data deduplication location for a single client:

In this scenario, a client is deduplicating data. However, because of security concerns, you want the IBM Tivoli Storage Manager server to deduplicate the data belonging to the client.

The data belonging client MATT is bound to a management class with a copy group that specifies a deduplication-enabled destination storage pool.

To change the data deduplication location from the client to the server, issue the following command:

```
update node matt deduplication=serveronly
```

No configuration is necessary on the client.

Specifying transaction and object size for deduplication

When you deduplicate large objects, intensive database activity can result from long-running transactions that are required to update the database. Server options are available to limit the size of transactions when client-side deduplicated data is backed up or archived and to limit the size of objects that are deduplicated by the server.

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 processed, the intensity and type of concurrent operations taking place on the IBM Tivoli Storage Manager server, and the Tivoli Storage Manager server configuration.

With the `SERVERDEDUPTXNLIMIT` server option, you can limit the size of objects that can be deduplicated on the server. With the `CLIENTDEDUPTXNLIMIT` server option, you can limit the size of transactions when client-side deduplicated data is backed up or archived.

Tip: To control which objects are deduplicated, you can also use the `MAXSIZE` parameter of the `DEFINE STGPOOL` and `UPDATE STGPOOL` commands. Using the `MAXSIZE` parameter, you can force large objects to the `NEXT` storage pool for storage.

For details about these options, see the *Administrator's Reference*.

Displaying statistics about server-side data deduplication

Important statistics about data deduplication are available by querying the server for information about storage pools or duplicate-identification processes.

You can also obtain statistics about client-side data deduplication. For details, see Backup-Archive Clients Installation and User's Guide.

Querying a storage pool for statistics about data deduplication

You can query a storage pool for important statistics about data deduplication.

Querying a storage pool provides the following statistics:

- Whether the storage pool has been set up for data deduplication
- The number of duplicate-identification processes specified when the storage pool was created
- The amount of data that was removed from the storage pool by reclamation processing

To query a storage pool for statistics about data deduplication, issue the `QUERY STGPOOL` command.

You might notice a discrepancy between the number of duplicate-identification processes specified as the default for a storage pool and the number of duplicate-identification processes currently running. This discrepancy occurs when you manually increase or decrease the number of duplicate-identification processes for the storage pool.

Remember: Querying a storage pool displays storage-pool utilization as a percentage of its assigned capacity. (Storage-pool utilization is expressed as *Pct Util* in the command output.) 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 percentage utilized is not displayed until all duplicate data is removed from the storage pool.

Querying a duplicate-identification process

Querying a duplicate-identification process displays the total number of bytes and total number of files processed.

To query a duplicate-identification process, issue the `QUERY PROCESS` command.

Querying a volume for information about linked files

You can query a volume for information about client files that link to files on other volumes. This information is useful when file extents created by data deduplication are distributed on different volumes.

You can display information only about files that are linked to a volume or only about files that are stored on a volume. You can also display information about both stored files and linked files.

To display information about files on a volume, issue the `QUERY CONTENT` command and specify the `FOLLOWLINKS` parameter.

For example, suppose a volume in a deduplicated storage pool is physically destroyed. You must restore this volume. Before you do, you want to determine whether other volumes in the storage pool have files that are linked to files in the destroyed volume. With that information, you can decide whether to restore the other volumes. To identify links, you issue the `QUERY CONTENT` command for

the destroyed volume and specify the **FOLLOWLINKS** parameter to list all the files with links to files on the destroyed volume.

Improving performance when reading from deduplicated storage pools

To obtain the different extents that make up a file from a deduplicated storage pool, client restore operations and certain server processes might require opening and closing FILE volumes multiple times. The frequency with which FILE volumes are opened and closed during a session can severely affect performance.

Opening and closing volumes multiple times can affect the following server processes that read data from a deduplicated storage pool:

- Volume reclamation
- MOVE DATA or MOVE NODEDATA
- EXPORT
- AUDIT VOLUME
- Storage-pool restore operation
- Volume restore operation
- Data migration

To reduce the number of times a volume is opened and closed, IBM Tivoli Storage Manager allows multiple input FILE volumes in a deduplicated storage pool to remain open at the same time during a session. To specify the number of open FILE volumes in deduplicated storage pools that can remain open, use the **NUMOPENVOLSALLOWED** server option. Set this option in the server options file or by using the **SETOPT** command.

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 standard 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.

Tip: This option can significantly increase the number of volumes and mount points in use at any one time. To optimize performance, complete the following tasks:

- To set **NUMOPENVOLSALLOWED**:
 1. Select a beginning value. The default is recommended.
 2. Monitor client sessions and server processes.
 3. Note the highest number of volumes open for a single session or process. If the highest number of open volumes is equal to the value specified by **NUMOPENVOLSALLOWED**, increase the setting of **NUMOPENVOLSALLOWED**.
- To prevent sessions or processes from having to wait for a mount point:
 1. Increase the value of the **MOUNTLIMIT** parameter in the device-class definition.

2. 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 volumes specified by the **NUMOPENVOLSALLOWED** option.
3. Check the following results:
 - 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.
- For any node backing up or archiving data into a deduplicated storage pool, 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 client operations because the **MAXNUMMP** value is being exceeded.

Writing data simultaneously to primary, copy, and active-data pools

With IBM Tivoli Storage Manager, you can write data simultaneously to a primary storage pool, copy storage pools, and active-data pools. The simultaneous-write function increases your level of data protection and reduces the amount of time required for storage pool backup.

You can write data simultaneously during any of the following operations:

- Client store sessions, for example:
 - Backup and archive sessions by Tivoli Storage Manager backup-archive clients.
 - Backup and archive sessions by application clients using the Tivoli Storage Manager API.
 - Migration processes by hierarchical storage management (HSM) clients. Migrated data is simultaneously written only to copy storage pools. Migrated data is not permitted in active-data pools.
- Server migration of data within a storage pool hierarchy.
- Server import processes that involve copying exported file data from external media to a primary storage pool that is configured for the simultaneous-write function. Imported data is simultaneously written to copy storage pools. Imported data is not simultaneously written to active-data pools. To store newly imported data into an active-data pool, use the **COPY ACTIVEDATA** command.

The maximum number of copy storage pools and active-data pools to which data can be simultaneously written is three. For example, you can write data simultaneously to three copy storage pools, or you can write data simultaneously to two copy storage pools and one active-data pool.

Attention: Do not use the simultaneous-write function to replace the task of regularly backing up storage pools. If you use the function to simultaneously write to copy storage pools, active-data pools, or both, ensure that the copy of each primary storage pool is complete by regularly issuing the BACKUP STGPOOL command and the COPY ACTIVE DATA command. If you fail to regularly back up storage pools, you can lose the ability to recover primary storage pool data. For example, if a copy storage pool fails during a write operation and the COPYCONTINUE parameter is set to YES, the Tivoli Storage Manager server removes the failed copy storage pool from the copy pool list for the remainder of the client session. After the copy storage pool is removed, the Tivoli Storage Manager server continues to write to the primary storage pool and to any remaining copy storage pools and active-data pools. If these pools become damaged or lost, and if you did not issue the BACKUP STGPOOL command for the copy storage pool that failed, you might not be able to recover your data.

Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied when primary storage pools are backed up or when active data is copied.

Guidelines for using the simultaneous-write function

The goal of the simultaneous-write function is to minimize the amount of time that is required for storage-pool backup operations. IBM Tivoli Storage Manager provides several options for accomplishing this goal. Your choice depends on how you want to manage your environment.

You can specify the simultaneous-write function for a primary storage pool if it is the target for client store sessions, server import processes, or server data-migration processes. You can also specify the simultaneous-write function for a primary storage pool when it is the target for *all* of the eligible operations.

Writing data simultaneously during client store sessions might be the logical choice if you have sufficient time for mounting and removing tapes during the client store session. However, if you choose this option you must ensure that a sufficient number of mount points and drives are available to accommodate all the client nodes that are storing data.

As a best practice, you are probably issuing the BACKUP STGPOOL and COPY ACTIVE DATA commands for all the storage pools in your storage pool hierarchy. If you are, and if you migrate only a small percentage of data from the primary storage pool daily, writing data simultaneously during client store sessions is the most efficient option. This method is efficient because data is stored in copy storage pools and active-data pools when the client stores the data. Little or no data is left to copy during backup storage pool or copy active-data operations.

Writing data simultaneously during server data-migration processes is another option. It is the most efficient method of writing data simultaneously if you migrate *all* the data in your primary storage pool nightly and then back up the primary storage pools. The reason for the efficiency is that data written simultaneously to copy storage pools or active-data pools during migration is not copied during backup storage pool or copy active-data operations.

Use the simultaneous-write function during migration if you have many client nodes and the number of mount points that are required to write data simultaneously during client store sessions is unacceptable. Similarly, mounting

and removing tapes when writing data simultaneously during client store sessions might be taking too much time. If so, consider writing data simultaneously during migration.

Another option is to specify the simultaneous-write function for a primary storage pool if it is the target for any of the eligible operations (client store, server import, server migration). This choice is preferable if, for example, you have large files to back up (for example, image files, database backup files, or Tivoli Data Protection files). Instead of backing up these files to the random-access disk storage pool at the top of the storage hierarchy, you can back them up to the sequential-access disk storage pool that is the next storage pool in the hierarchy. If you specify the simultaneous-write function for any of the eligible operations for the next storage pool, the following events occur:

- Large files that are backed up to the next storage pool are simultaneously written to copy storage pools (and active-data pools, if you have them).
- Other files that migrate to the next storage pool are simultaneously written to the same copy storage pools (and active-data pools, if you have them).

By default, the Tivoli Storage Manager server writes data simultaneously during client store sessions if you have copy storage pools or active-data pools defined to the target storage pool.

You can also disable the simultaneous-write function. This option is useful if you have copy storage pools or active-data pools defined, but you want to disable the simultaneous-write function without deleting and redefining the pools.

Limitations that apply to the simultaneous-write function

Using the simultaneous-write function requires consideration of factors such as storage devices and network configuration.

The following limitations apply:

- Data cannot be written simultaneously to copy storage pools and active-data pools during server data movements such as reclamation, moving data from one storage pool to another storage pool, or backing up a storage pool.
- Simultaneous-write operations take precedence over LAN-free data movement. The operations go over the LAN, and the simultaneous-write configuration is honored.
- You can back up or archive copies of files that were migrated by a Tivoli Storage Manager for Space Management client to the same Tivoli Storage Manager server to which they were migrated. However, the files are stored only in the primary storage pool. As a best practice, create current backup and archive versions of the files before the Tivoli Storage Manager for Space Management client migrates them.
- Target storage pools used for simultaneous-write operations can have different device classes. Performance is limited by the speed of the slowest device.
- The **COPYSTGPOOLS** and **ACTIVEDATAPOOLS** parameters are 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

- Limitations apply when a NAS backup operation is writing a TOC file. If the primary storage pool that is specified in the TOCDESTINATION in the copy group of the management class has copy storage pools or active-data pools defined, the copy storage pools and active-data pools are ignored. The data is stored only in the primary storage pool.

Controlling the simultaneous-write function

You control the simultaneous-write function by specifying certain parameters when you define or update primary storage pools. You can control when data is written simultaneously. You can also specify the copy storage pools and active-data pools to which data is simultaneously written.

Specifying when the simultaneous-write operations occur

You can specify simultaneous-write operations for any primary storage pool that is the target of client store sessions, server import processes, or server data-migration processes. You can also disable the simultaneous-write function.

To control the simultaneous-write function, use the **AUTOCOPY** parameter on the **DEFINE STGPOOL** or **UPDATE STGPOOL** commands for primary storage pools.

Remember:

- Specify a value for the **AUTOCOPY** parameter on the primary storage pool that is the target of data movement. (The default is to write data simultaneously during client store sessions and server import processes.) For example, if you want to write data simultaneously only during server data-migration processes, specify **AUTOCOPY=MIGRATION** in the definition of the next storage pool in the storage pool hierarchy.
- The **AUTOCOPY** parameter is not available for copy storage pools or active-data pools.

IBM Tivoli Storage Manager provides the following options for controlling when simultaneous-write operations occur:

- To disable the simultaneous-write function, specify **AUTOCOPY=NONE**.
This option is useful, if, for example, you have copy storage pools or active-data pools defined, and you want to temporarily disable the simultaneous-write function without having to delete and then redefine the pools.
- To specify simultaneous-write operations only during client store sessions and server import processes, specify **AUTOCOPY=CLIENT**.
During server import processes, data is simultaneously written only to copy storage pools. Data is not written to active-data pools during import processes.
- To specify that simultaneous-write operations take place only during server data-migration processes, specify **AUTOCOPY=MIGRATION**.
During server data migration, data is simultaneously written to copy storage pools and active-data pools only if the data does not exist in those pools.
- To specify that simultaneous-write operations take place during client store sessions, server data-migration processes, and server import processes, specify **AUTOCOPY=ALL**.

A primary storage pool can be the target for more than one type of data movement. For example, the next storage pool in a storage pool hierarchy can be the target for data migration from the primary storage pool at the top of the hierarchy. The next storage pool can also be the target for direct backup of certain types of client files (for example, image files). The **AUTOCOPY=ALL**

setting on a primary storage pool ensures that data is written simultaneously during both server data-migration processes and client store sessions.

The following table provides examples of **AUTOCOPY** settings for some common scenarios in which the simultaneous-write function is used.

Table 34. AUTOCOPY settings

If your goal is...	Set the AUTOCOPY parameter for the primary storage pool at the top of the storage hierarchy to...	Set the AUTOCOPY parameter for the next primary storage pool to...
To disable the simultaneous-write function	NONE	NONE
To enable simultaneous-write operations only during client store sessions and server import processes	CLIENT	NONE
To enable simultaneous-write operations only during server data-migration processes	NONE	MIGRATION
To simultaneously write client files to copy storage pools during migration. You also want simultaneous-write operations to occur for files that are directly backed up to the next storage pool.	NONE	ALL
To enable simultaneous-write operations during any of the following operations: client store sessions, server import processes, and server data-migration processes.	ALL	ALL

For details about the **DEFINE STGPOOL** and **UPDATE STGPOOL** commands and parameters, see the *Administrator's Reference*.

Specifying copy pools and active-data pools for simultaneous-write operations

The maximum number of copy storage pools and active-data pools to which data can be simultaneously written is three. For example, you can write data simultaneously to three copy storage pools. You can also write data simultaneously to two copy storage pools and one active-data pool, and so on.

The parameters that are used to specify copy storage pools and active-data pools are on the **DEFINE STGPOOL** and **UPDATE STGPOOL** commands.

- To specify copy storage pools, use the **COPYSTGPOOLS** parameter.
- To specify active-data pools, use the **ACTIVEDATAPOOLS** parameter.

For details about the **DEFINE STGPOOL** and **UPDATE STGPOOL** commands, refer to the *Administrator's Reference*.

Related concepts

“Rules of inheritance for the simultaneous-write function” on page 291

Specifying how the server reacts to a write failure during simultaneous-write operations

Write failures to copy or active-data pools might occur during while data is being simultaneously written. If a write failure occurs during a client store session, you can specify whether to continue or discontinue the operation.

Use the **COPYCONTINUE** parameter on the **DEFINE STGPOOL** command to specify how the server reacts to a write failure to copy storage pools during client store sessions:

- To stop writing to failing copy storage pools for the remainder of the session, but continue storing files into the primary pool and any remaining copy pools or active-data pools, specify **COPYCONTINUE=YES**.

The copy storage 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.

- To fail the transaction and discontinue the store operation, specify **COPYCONTINUE=NO**.

Restrictions:

- The setting of the **COPYCONTINUE** parameter does not affect active-data pools. If a write failure occurs for any of 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 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.

For details about the **DEFINE STGPOOL** and **UPDATE STGPOOL** commands and parameters, refer to the *Administrator's Reference*.

Related concepts

“Rules of inheritance for the simultaneous-write function” on page 291

Rules of inheritance for the simultaneous-write function

When switching primary storage pools during client store sessions or server import processes, certain rules of inheritance apply to copy storage pool lists, active-data pool lists, and the setting of the **COPYCONTINUE** parameter.

When a client backs up, archives, or migrates a file, or when the server imports data, the data is written to the primary storage pool that is specified by the copy group of the management class that is bound to the data. If a data storage operation or a server import operation switches from the primary storage pool at the top of a storage hierarchy to a next primary storage pool in the hierarchy, the next storage pool inherits the list of copy storage pools, the list of active-data pools, and the value of the **COPYCONTINUE** parameter from the primary storage pool at the top of the storage pool hierarchy.

The following rules apply during a client store session or a server import process when the server must switch primary storage pools:

- If the destination primary storage pool has one or more copy storage pools or active-data pools defined using the **COPYSTGPOOLS** or **ACTIVEDATAPOOLS** parameters, the server writes the data to the next storage pool and to the copy storage pools and active-data pools that are defined to the destination primary pool, regardless whether the next pool has copy pools defined.

The setting of the **COPYCONTINUE** of the destination primary storage pool is inherited by the next primary storage pool. The **COPYCONTINUE** parameter specifies how the server reacts to a copy storage-pool write failure for any of the copy storage pools listed in the **COPYSTGPOOLS** parameter. If the next pool has copy storage pools or active-data pools defined, they are ignored as well as the value of the **COPYCONTINUE** parameter.

- If no copy storage pools or active-data pools are defined in the destination primary storage pool, the server writes the data to the next primary storage pool. If the next pool has copy storage pools or active-data pools defined, they are ignored.

These rules apply to all the primary storage pools within the storage pool hierarchy.

Related tasks

“Specifying copy pools and active-data pools for simultaneous-write operations” on page 289

“Specifying how the server reacts to a write failure during simultaneous-write operations” on page 290

Simultaneous-write operations: Examples

Illustrated examples of simultaneous-write operations show how the function works in various storage pool configurations. Other examples show what happens when an error occurs during a simultaneous-write operation.

Examples of simultaneous-write operations during client store operations

Examples show how the simultaneous-write function works during client store operations. In all the examples, client nodes, whose files require fast restore, are members of a policy domain that specifies an active-data pool.

For these examples, assume the following conditions:

- Primary storage pools DISKPOOL and TAPEPOOL are linked to form a storage hierarchy. DISKPOOL is at the top of the storage hierarchy and TAPEPOOL is the next pool in the storage hierarchy.
- The value of the **AUTOCOPY** parameter for DISKPOOL is CLIENT. The value of the **AUTOCOPY** parameter for TAPEPOOL is NONE.
- The active backup data belonging to certain clients must be restored as quickly as possible if a disaster occurs. These clients are members of policy domain FASTRESTORE, which specifies an active-data pool as the destination for active backup data. Files A and B belong to a node in this domain and are bound to management class STANDARD. The destination specified in its backup copy group is DISKPOOL.
- The data belonging to other nodes is less critical. Restore times are flexible. These nodes are assigned to policy domain NORMAL, which does not have an active-data pool specified. Files C, D, and E belong to one of the nodes in this domain and are bound to management class STANDARD. The destination that is specified in its backup copy group is DISKPOOL.
- DISKPOOL has enough space to store only files C and D, but its next pool (TAPEPOOL) has enough space for file E.

Related concepts

Chapter 14, “Implementing policies for client data,” on page 435

Writing data simultaneously during a simple client store session:

In this example, the simultaneous-write function automatically copies client data to two copy storage pools and an active-data pool during a client store operation.

With DISKPOOL and TAPEPOOL already defined as your storage pool hierarchy, issue the following commands to enable the simultaneous-write function:

```
define stgpool cypool1 mytapedevice pooltype=copy
define stgpool cypool2 mytapedevice pooltype=copy
define stgpool activedatapool mydiskdevice pooltype=activedata
update stgpool diskpool copystgpools=cypool1,cypool2 copycontinue=yes
    activedatapools=activedatapool
```

where MYTAPEDEVICE is the device-class name associated with the copy storage pools and MYDISKDEVICE is the device-class name associated with the active-data pool.

The storage pool hierarchy and the copy storage pools and active-data pool associated with DISKPOOL are displayed in Figure 22 on page 293.

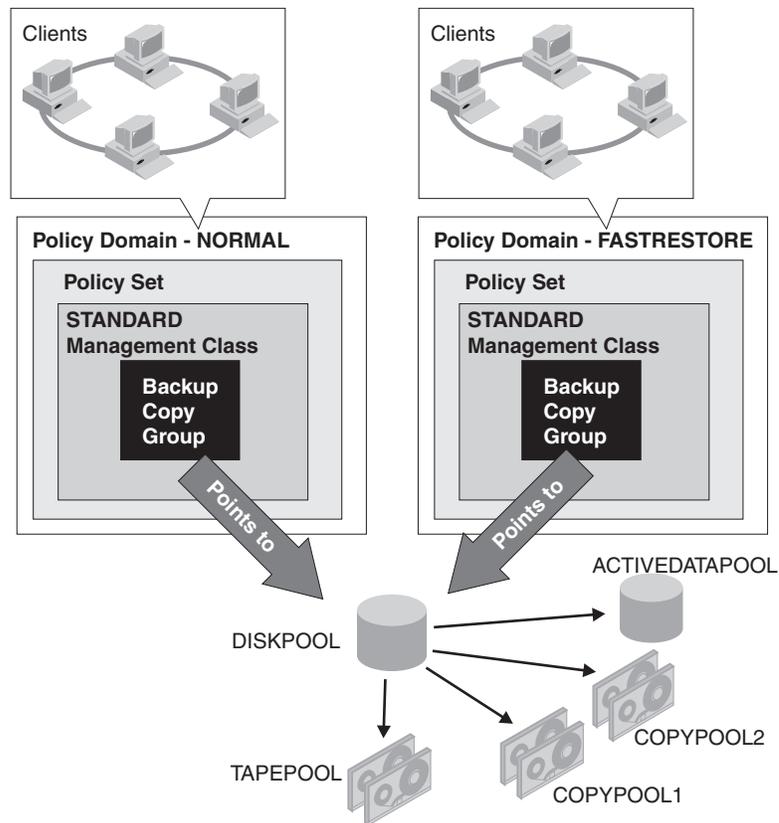


Figure 22. Example of storage pool hierarchy with copy storage pools defined for DISKPOOL

During a simultaneous-write operation, the next storage pool TAPEPOOL inherits the list of copy storage pools (COPYPOOL1 and COPYPOOL2) and the value of the COPYCONTINUE parameter from DISKPOOL, the primary pool at the top of the storage pool hierarchy. TAPEPOOL also inherits the list of active-data pools (ACTIVEDATAPOOL). When files A, B, C, D, and E are backed up, the following events occur:

- A and B are written to DISKPOOL, COPYPOOL1, COPYPOOL2, and ACTIVEDATAPOOL.
- C and D are written to DISKPOOL, COPYPOOL1, and COPYPOOL2.
- File E is written to TAPEPOOL, COPYPOOL1 and COPYPOOL2.

See Figure 23 on page 294.

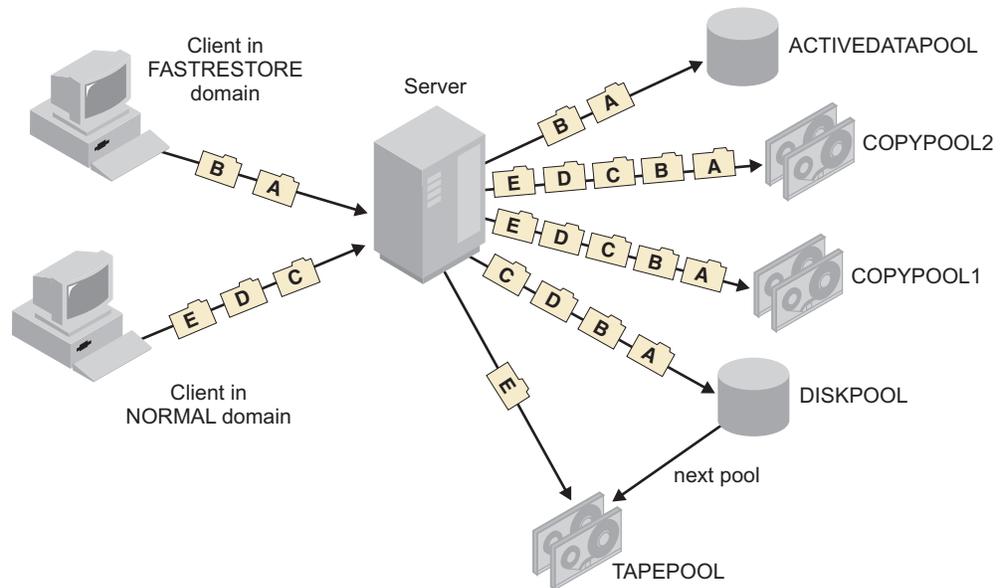


Figure 23. Inheriting a list of copy storage pools

As a precaution, issue the `BACKUP STGPOOL` and `COPY ACTIVATEDATA` commands after the backup operation has completed. Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied when storage pools are backed up or when active data is copied.

Inheritance of an empty copy storage-pool list during a simultaneous-write operation:

In this example, the next storage pool in a hierarchy inherits empty copy storage pool and active-data pool lists from the primary storage pool at the top of the storage hierarchy.

You do not specify a list of copy storage pools for `DISKPOOL`. However, you do specify copy storage pools for `TAPEPOOL` (`COPYPOOL1` and `COPYPOOL2`) and an active-data pool (`ACTIVATEDATAPOOL`). You also specify a value of `YES` for the `COPYCONTINUE` parameter. Issue the following commands to enable the simultaneous-write function:

```
define stgpool copypool1 mytapedevice pooltype=copy
define stgpool copypool2 mytapedevice pooltype=copy
define stgpool activedatapool mydiskdevice pooltype=activedata
update stgpool tapepool copystgpools=copypool1,copypool2
        copycontinue=yes activedatapools=activedatapool
```

where `MYTAPEDEVICE` is the device-class name associated with the copy storage pools and `MYDISKDEVICE` is the device-class name associated with the active-data pool. Figure 24 on page 295 displays this configuration.

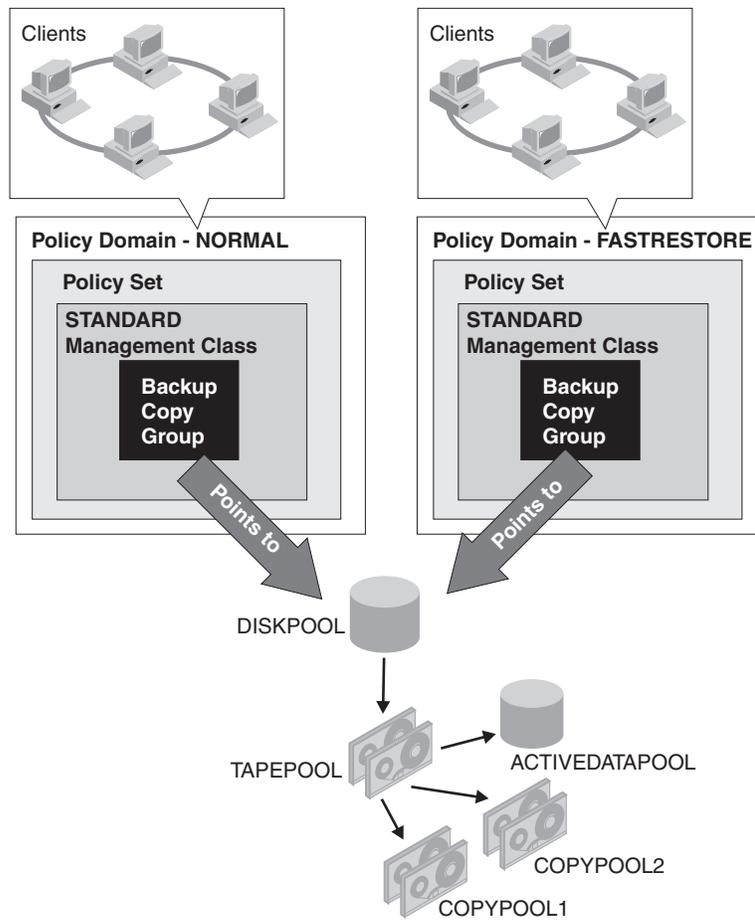


Figure 24. Example of storage pool hierarchy with copy storage pools defined for TAPEPOOL

When files A, B, C, D, and E are backed up, the following events occur:

- A, B, C, and D are written to DISKPOOL.
- File E is written to TAPEPOOL.

See Figure 25 on page 296.

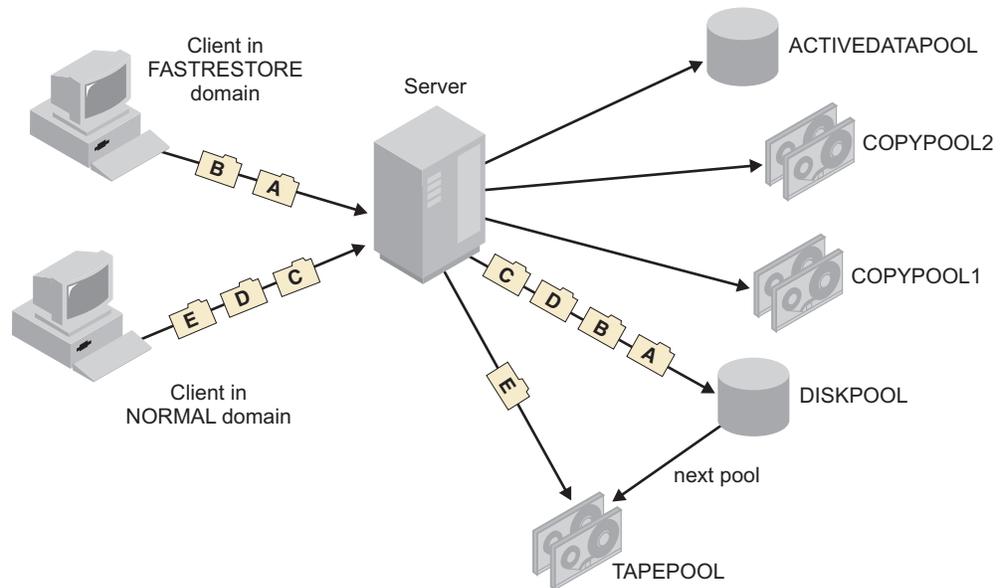


Figure 25. Inheriting an empty copy storage pool list

Although TAPEPOOL has copy storage pools and an active-data pool defined, file E is not copied because TAPEPOOL inherits empty copy storage pool and active-data pool lists from DISKPOOL.

As a precaution, issue the `BACKUP STGPOOL` and `COPY ACTIVEDATA` commands after the backup operation has completed. Data that is simultaneously that is written to copy storage pools or active-data pools during migration is not copied when primary storage pools are backed up or when active data is copied.

A simultaneous-write error during a client store operation:

In this example, data is not written to one copy storage pool in a simultaneous-write operation.

You specify `COPYPOOL1` and `COPYPOOL2` as copy storage pools for `DISKPOOL` and you set the value of the `COPYCONTINUE` parameter to `YES`. You also specify `ACTIVEDATAPOOL` as the active-data pool for `DISKPOOL`. This configuration is identical to the configuration in the first example.

When files A, B, C, D, and E are backed up, the following events occur:

- An error occurs while writing to `COPYPOOL1`, and it is removed from the copy storage pool list that is held in memory by the server. The transaction fails.
- Because the value of the `COPYCONTINUE` parameter is `YES`, the client tries the backup operation again. The in-memory copy storage pool list, which is retained by the server for the duration of the client session, no longer contains `COPYPOOL1`.
- Files A and B are simultaneously written to `DISKPOOL`, `ACTIVEDATAPOOL`, and `COPYPOOL2`.
- Files C and D are simultaneously written to `DISKPOOL` and `COPYPOOL2`.
- File E is simultaneously written to `TAPEPOOL` and `COPYPOOL2`.

See Figure 26 on page 297.

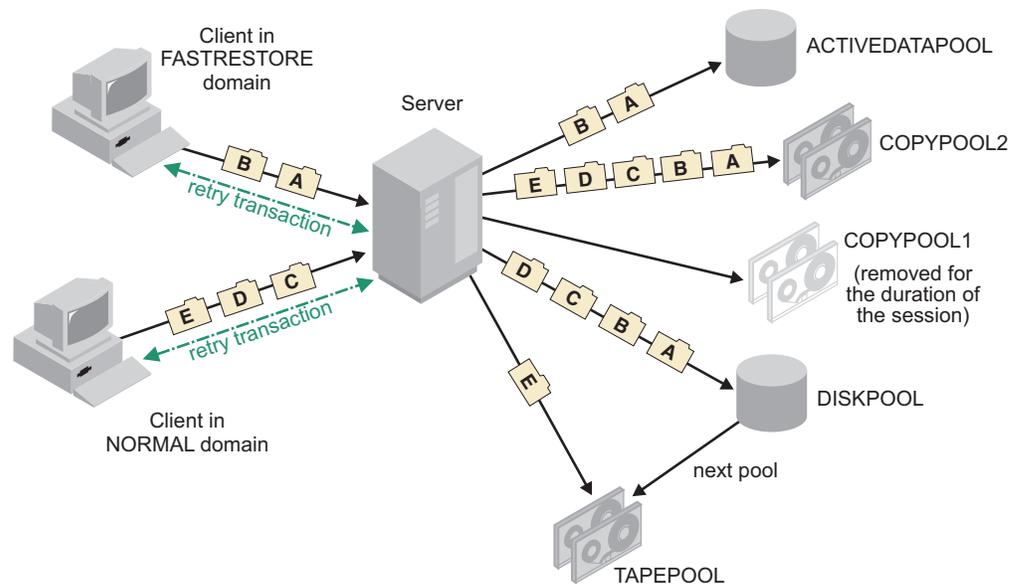


Figure 26. Inheriting a list of copy storage pools

In this scenario, if the primary storage pools and COPYPOOL2 become damaged or lost, you might not be able to recover your data. For this reason, issue the following BACKUP STGPOOL command for the copy storage pool that failed:

```
backup stgpool diskpool copystgpool1
backup stgpool tapepool copystgpool1
```

Suppose, in this scenario, that an error occurred while writing to ACTIVEDATAPOOL, rather than COPYPOOL1. In this situation, ACTIVEDATAPOOL is removed from the active-data pool list held in memory by the server, and the transaction fails. The client tries the backup operation again. The in-memory active-data pool list does not contain ACTIVEDATAPOOL. Files A, B, C, and D are written simultaneously to DISKPOOL, COPYPOOL1, and COPYPOOL2. File E is written to TAPEPOOL, COPYPOOL1, and COPYPOOL2. However, files A and B are not written to the active-data pool.

You can still recover the primary storage pools from COPYPOOL1 and, if necessary, COPYPOOL2. However, if you want active backup data available in the active-data pool for fast client restores, you must issue the following command:

```
copy activedata diskpool activedatapool
```

As a precaution, issue the BACKUP STGPOOL and COPY ACTIVEDATA commands after the backup operation has completed. Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied when primary storage pools are backed up or when active data is copied.

Examples of simultaneous-write operations during server data-migration processes

Examples show how the simultaneous-write function works during server migration of data within a storage pool hierarchy.

For these examples, assume the following conditions:

- Primary storage pools FILEPOOL, which is associated with a sequential-access device (device type FILE), and TAPEPOOL are linked to form a storage hierarchy. FILEPOOL is at the top of the storage hierarchy. TAPEPOOL is the next pool in the storage hierarchy.
- The files in FILEPOOL are eligible to be migrated.
- One or more copy storage pools are defined to FILEPOOL and TAPEPOOL. There are no active-data pools.
- Copies of one or more of the files in FILEPOOL exist in a copy storage pool.

Simultaneous-write operation in a simple migration scenario:

In this example, the storage pool hierarchy contains two primary storage pools. The next storage pool has two copy storage pools defined. A copy of one of the files to be migrated to the next storage pool exists in one of the copy storage pools.

FILEPOOL and TAPEPOOL are defined in your storage pool hierarchy. Two copy storage pools, COPYPOOL1 and COPYPOOL2, are defined to TAPEPOOL. Files A, B, and C are in FILEPOOL and eligible to be migrated. A copy of file C exists in COPYPOOL2.

The storage pool hierarchy and the copy storage pools that are associated with TAPEPOOL are displayed in Figure 27.

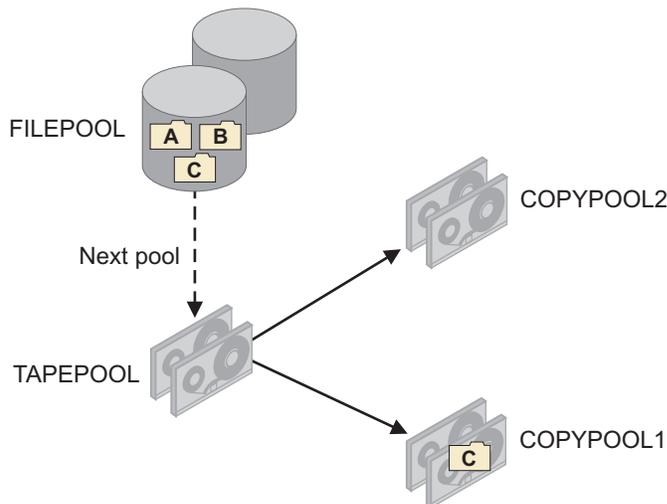


Figure 27. A storage pool hierarchy with files eligible to be migrated

To specify simultaneous-write operations only during migration, issue the following command for TAPEPOOL:

```
update stgpool tapepool autocopy=migration
```

Tip: In this example, the setting of the **AUTOCOPY** parameter for FILEPOOL is not relevant. TAPEPOOL is the target of the data migration.

When files A, B and C are migrated, the following events occur:

- Files A and B are simultaneously written to TAPEPOOL, COPYPOOL1 and COPYPOOL2.
- File C is simultaneously written to TAPEPOOL and COPYPOOL2. File C is not written to COPYPOOL1 because COPYPOOL1 has a copy of the file.

See Figure 28.

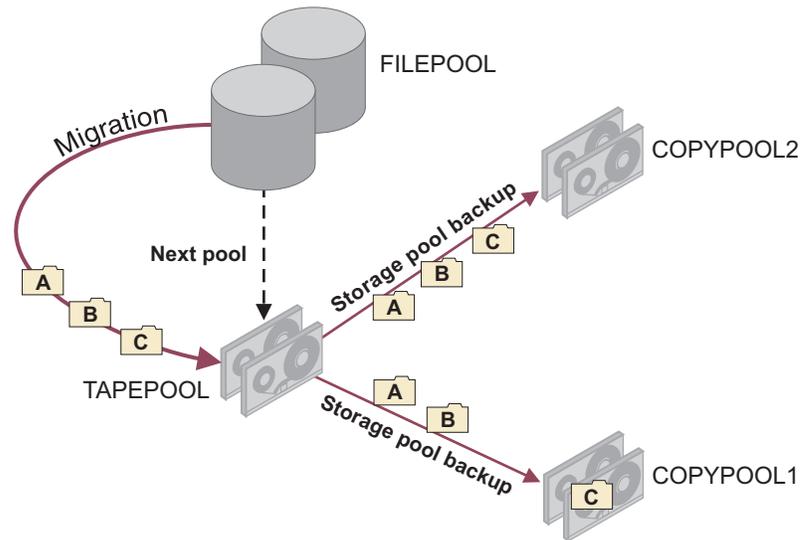


Figure 28. Simultaneous-write operation during migration to two copy storage pools

As a precaution, issue the BACKUP STGPOOL and COPY ACTIVATEDATA commands after the migration operation has completed. Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied when primary storage pools are backed up or when active data is copied.

A simultaneous-write error during server data migration:

In this example, the storage pool hierarchy contains two primary storage pools. The next storage pool has two copy storage pools defined. A copy of one of the files to be migrated to the next storage pool exists in a copy storage pool. A write error to the pool occurs.

FILEPOOL and TAPEPOOL are defined in the storage pool hierarchy. Two copy storage pools, COPYPOOL1 and COPYPOOL2, are defined to TAPEPOOL. Files A, B, and C are in FILEPOOL and are eligible to be migrated. A copy of file C exists in COPYPOOL1.

The storage pool hierarchy and the copy storage pools that are associated with TAPEPOOL are displayed in Figure 29 on page 300.

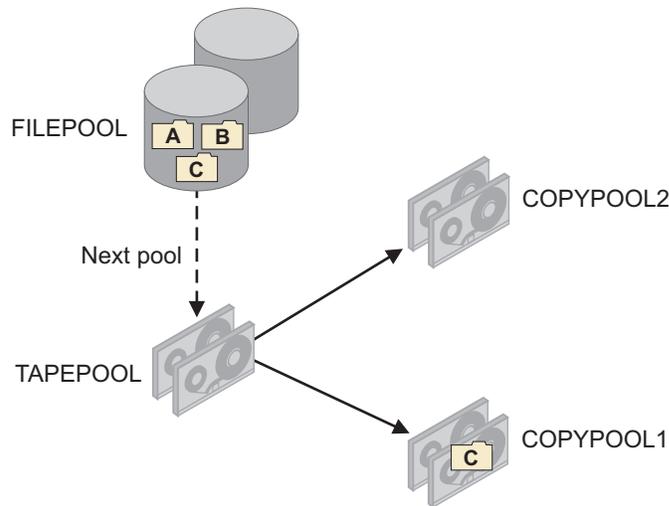


Figure 29. A storage pool hierarchy with files eligible to be migrated

To specify simultaneous-write operations only during migration, issue the following command for TAPEPOOL:

```
update stgpool tapepool autocopy=migration
```

Tip: In this example, the setting of the **AUTOCOPY** parameter for FILEPOOL is not relevant. TAPEPOOL is the target of the data migration.

When files A, B and C are migrated, the following events occur:

- An error occurs writing to COPYPOOL1.
- COPYPOOL1 is removed from the in-memory list. The in-memory list is kept for the duration of the migration process.
- The transaction fails and the server tries the operation again.
- Files A, B, and C are simultaneously written to TAPEPOOL and COPYPOOL2.

See Figure 30.

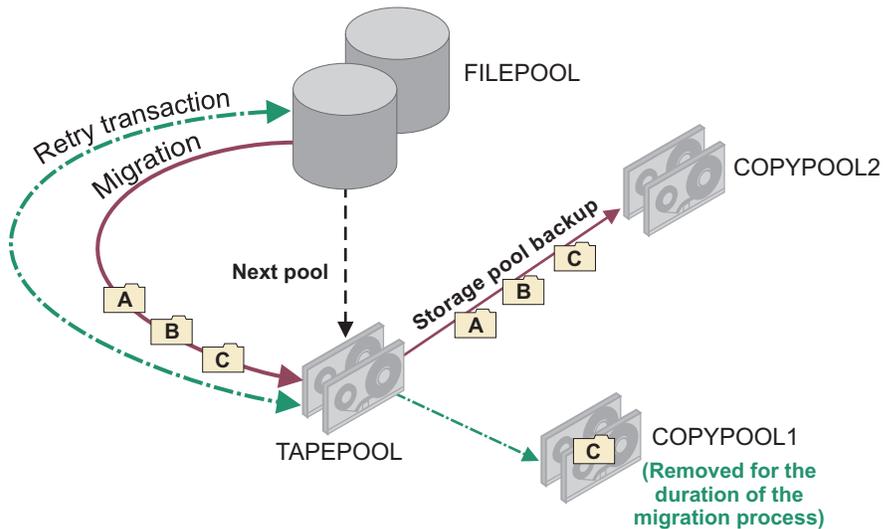


Figure 30. An error occurs during simultaneous-write operation during migration

As a precaution, issue the `BACKUP STGPOOL` and `COPY ACTIVATEDATA` commands after the migration operation has completed. Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied when primary storage pools are backed up or when active data is copied.

Inheriting a list of copy storage pools during a simultaneous-write operation:

In this example, three primary storage pools are linked to form a storage pool hierarchy. The next storage pool in the hierarchy has a storage pool list. The last pool in the hierarchy inherits the list during a simultaneous-write operation.

`FILEPOOL1`, `FILEPOOL2`, and `TAPEPOOL` are defined in your storage pool hierarchy. One copy storage pool, `COPYPOOL`, is defined to `FILEPOOL2`.

- Files A, B, and C on `FILEPOOL1` are eligible to be migrated.
- `FILEPOOL2` has sufficient space only for files B and C, but not A. `TAPEPOOL` has enough space for file A.
- A copy of file C exists in `COPYPOOL`.

The storage pool hierarchy and the copy storage pool are displayed in Figure 31.

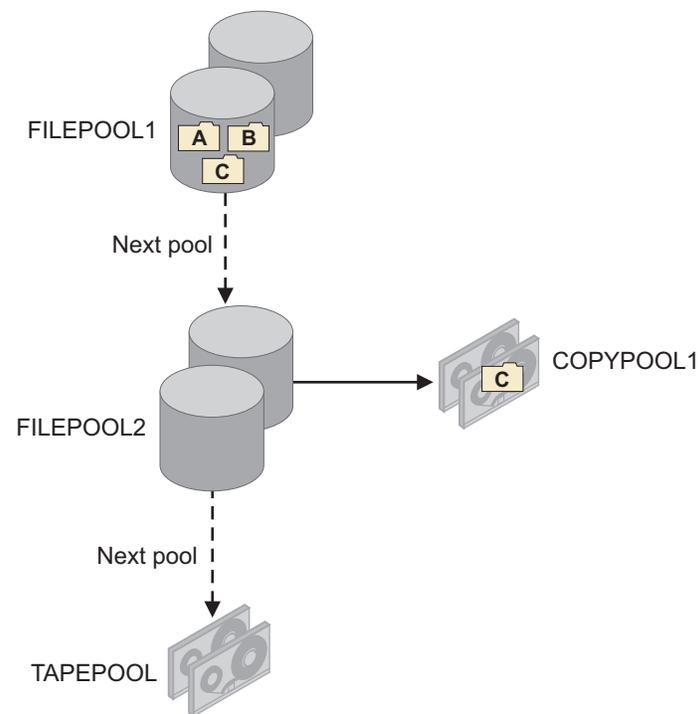


Figure 31. Three-tiered storage pool hierarchy with one copy storage pool

Issue the following commands for `FILEPOOL2` and `TAPEPOOL` to enable the simultaneous-write function only during migration:

```
update stgpool filepool2 autocopy=migration
update stgpool tapepool autocopy=migration
```

Tip: In this example, the setting of the `AUTOCOPY` parameter for `FILEPOOL1` is not relevant. `FILEPOOL2` and `TAPEPOOL` are the targets of the data migration.

When files A, B, and C are migrated, the follow events occur:

- File B is migrated to FILEPOOL2 and simultaneously written to COPYPOOL.
- File C is migrated to FILEPOOL2. It is not written to COPYPOOL because a copy of that file exists in COPYPOOL.
- File A is migrated to TAPEPOOL. TAPEPOOL inherits the copy storage pool list from FILEPOOL2 and simultaneously writes File A to COPYPOOL.

See Figure 32.

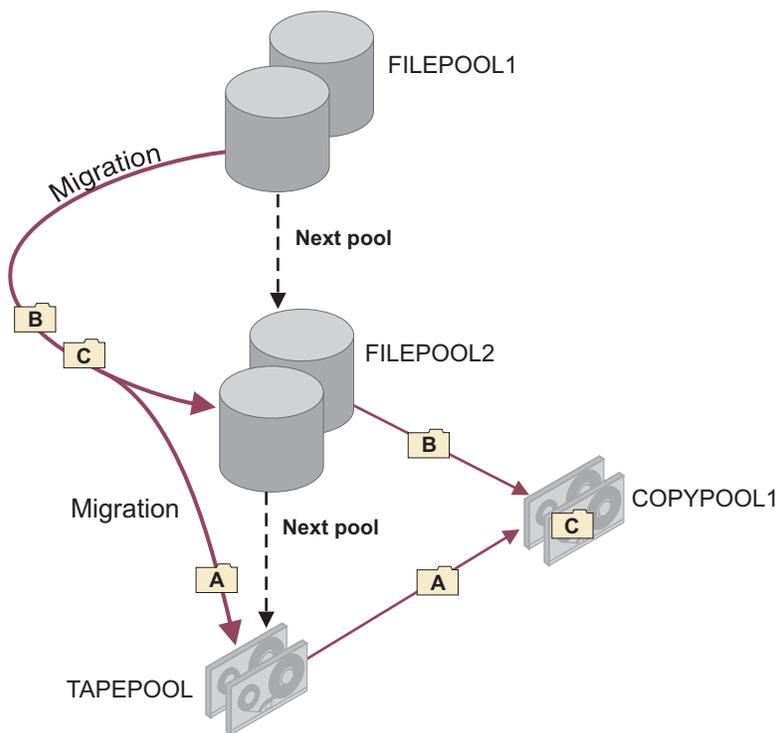


Figure 32. Inheriting a list of copy storage pools

As a precaution, issue the `BACKUP STGPOOL` and `COPY ACTIVEDATA` commands after the migration operation has completed. Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied when primary storage pools are backed up or when active data is copied.

Example of migration without a simultaneous-write operation

In this example, the simultaneous-write function to two copy storage pools is enabled for client store sessions. Files in the primary storage pool at the top of the storage hierarchy are migrated. Simultaneous-write operations during migration are not enabled.

Primary storage pools FILEPOOL and TAPEPOOL are linked to form a storage hierarchy. FILEPOOL is at the top of the storage hierarchy. TAPEPOOL is the next pool in the storage hierarchy. Two copy storage pools, COPYPOOL1 and COPYPOOL2, are defined to FILEPOOL. The value of the `AUTOCOPY` parameter for FILEPOOL is `CLIENT`. The value of the `AUTOCOPY` parameter for TAPEPOOL is `NONE`.

- Files A, B, and C were written to FILEPOOL during client backup operations.
- File C was simultaneously written to COPYPOOL1.
- The files in FILEPOOL are eligible to be migrated.

The storage pool hierarchy and the copy storage pools associated with FILEPOOL are displayed in Figure 33.

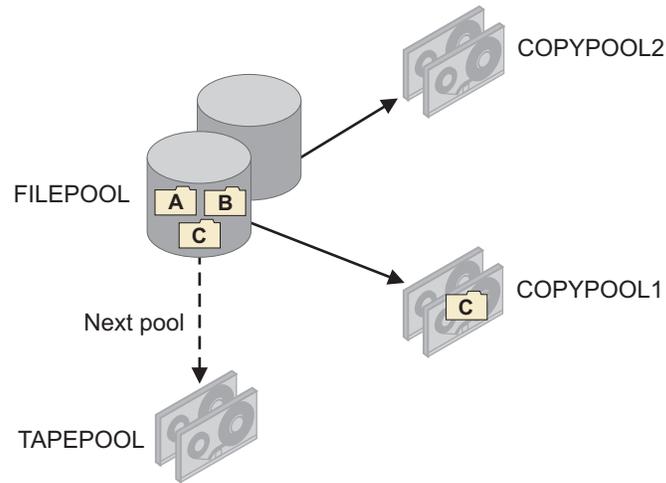


Figure 33. Storage pool hierarchy with files eligible to be migrated

When files A, B and C are migrated, they are written to TAPEPOOL. See Figure 34.

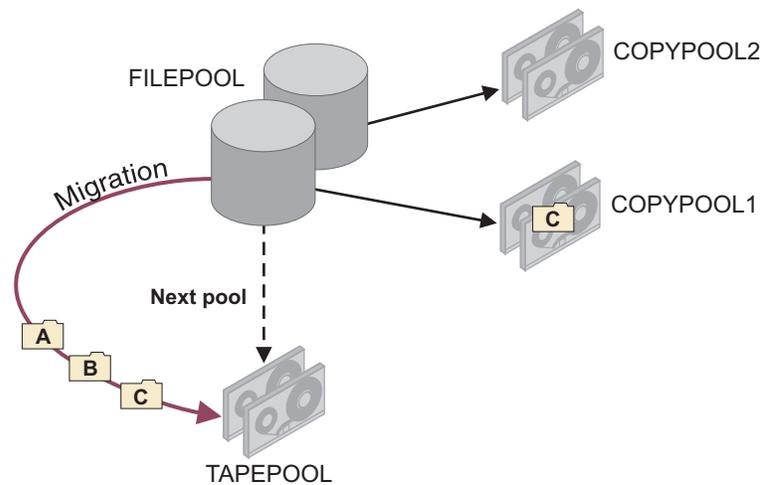


Figure 34. Migration after a simultaneous-write operation

As a precaution, issue the `BACKUP STGPOOL` and `COPY ACTIVATEDATA` commands after the migration operation has completed. Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied when primary storage pools are backed up or when active data is copied.

Example of a simultaneous-write operation during both migration and a client store operation

In this example, the storage pool hierarchy consists of a two primary storage pools. A single copy storage pool is defined to the pools. The simultaneous-write function during client store operations was enabled. However, one of the backed-up files was not written to the copy storage pool. The simultaneous-write function during migration is enabled.

Primary storage pools FILEPOOL and TAPEPOOL are linked to form a storage hierarchy. FILEPOOL is at the top of the storage hierarchy. TAPEPOOL is the next pool in the storage hierarchy. One copy storage pool, COPYPOOL, is defined to both FILEPOOL and TAPEPOOL:

- The simultaneous-write function during client store operations was enabled. (The setting of the **AUTOCOPY** parameter for FILEPOOL is CLIENT.)
- During client store operations, files A, B, and C were written to COPYPOOL. A failure occurred while writing file D to COPYPOOL.
- The simultaneous-write function during migration is enabled for TAPEPOOL. (The setting of the **AUTOCOPY** parameter for TAPEPOOL is MIGRATION.)

The storage pool hierarchy and the copy storage pool that are associated with FILEPOOL and TAPEPOOL are displayed in Figure 35.

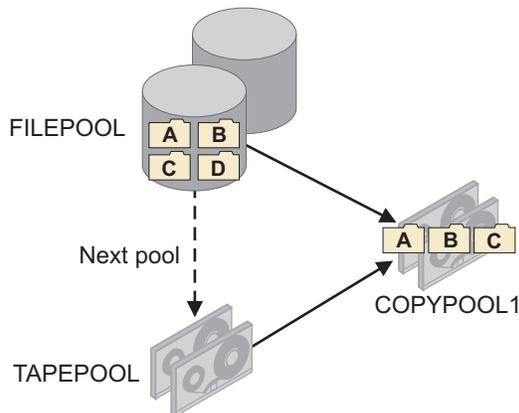


Figure 35. Storage pool hierarchy with files eligible to be migrated

When files A, B, C, and D are migrated, the following events occur:

- File D is migrated to TAPEPOOL and simultaneously written to COPYPOOL.
- Files A, B, and C are migrated to TAPEPOOL. They are not written to COPYPOOL because copies of those files exist in COPYPOOL.

See Figure 36 on page 305.

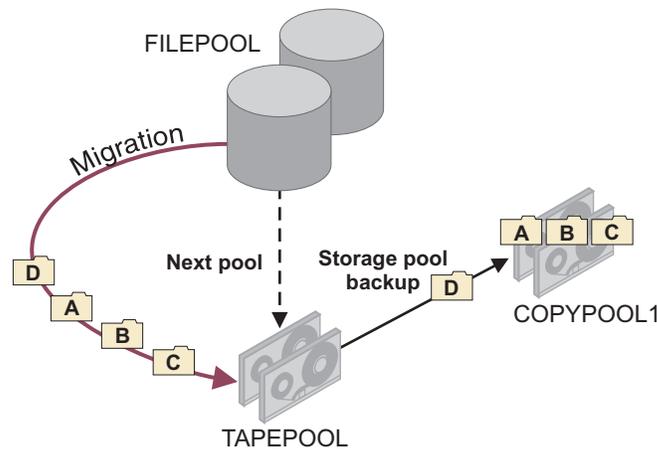


Figure 36. A simultaneous-write operation during both migration and client backup operations

As a precaution, issue the `BACKUP STGPOOL` and `COPY ACTIVATEDATA` commands after the migration operation has completed. Data that is simultaneously written to copy storage pools or active-data pools during migration is not copied when primary storage pools are backed up or when active data is copied.

Planning simultaneous-write operations

Before enabling the simultaneous-write function, you must consider available resources and configuration settings. As a best practice, you can separate data into discrete storage hierarchies.

Controlling the number of client mount points for simultaneous-write operations

During simultaneous-write operations, a client session requires a mount point for each sequential-access storage pool to which data is written. Transactions fail if the number of mount points that are required for a client session is insufficient.

Give careful consideration to the number of mount points that are available for a simultaneous-write operation. A client session requires a mount point to store data to a sequential-access storage pool. For example, if a storage pool hierarchy includes a *sequential* primary storage pool, the client node requires one mount point for that pool plus one mount point for each copy storage pool and active-data pool.

Suppose, for example, you create a storage pool hierarchy like the hierarchy shown in Figure 22 on page 293. `DISKPOOL` is a random-access storage pool, and `TAPEPOOL`, `COPYPOOL1`, `COPYPOOL2`, and `ACTIVEDATAPOOL` are sequential-access storage pools. For each client backup session, the client might have to acquire four mount points if it has to write data to `TAPEPOOL`. To run two backup sessions concurrently, the client requires a total of eight mount points.

To indicate the number of mount points a client can have, specify a value for the `MAXNUMMP` parameter on the `REGISTER NODE` or `UPDATE NODE` commands. Verify the value of the `MAXNUMMP` parameter and, if necessary, update it if you want to enable the simultaneous-write function. A value of 3 for the `MAXNUMMP` parameter might be sufficient if, during a client session, all the data is stored in `DISKPOOL`, `COPYPOOL1`, `COPYPOOL2`, and `ACTIVEDATAPOOL`.

If the number of mount points that are required for a client session exceeds the value of the client **MAXNUMMP** parameter, the transaction fails. If the transaction involves an active-data pool, all the active-data pools are removed from the active-data pool list for the duration of the client session, and the client tries the operation again. If the transaction involves a copy storage pool, the setting of the **COPYCONTINUE** parameter in the storage pool definition determines whether the transaction is tried again:

- If the value of the **COPYCONTINUE** parameter is **NO**, the client does not try the operation again.
- If the value of the **COPYCONTINUE** parameter is **YES**, **all** the copy storage pools are removed from the copy storage pool list for the duration of the client session. The client tries the operation again.

Restrictions:

- The setting of the **COPYCONTINUE** parameter does not affect active-data pools. If a write failure occurs for any of 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 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.

Controlling the number of mount points for a device class during simultaneous-write operations

If the number of sequential-access volumes that must be mounted for a simultaneous-write operation exceeds the maximum number of mount points specified for a device class, the server is not able to acquire the mount points and the operation fails.

To specify the maximum number of sequential-access volumes that can be simultaneously mounted, use the **MOUNTLIMIT** parameter in the device class definition.

If the simultaneous-write operation involves an active-data pool, the IBM Tivoli Storage Manager server tries to remove the active-data pools that use this device class until enough mount points can be acquired. The transaction fails, and the client tries the operation again. If sufficient mount points can be acquired when the operation is tried again, the data is written into the primary storage pool, any remaining active-data pools, and any copy storage pools, if they exist.

If the operation involves a copy storage pool, the value of the **COPYCONTINUE** parameter in the storage pool definition determines whether the client tries the operation again:

- If the value of the **COPYCONTINUE** parameter is **NO**, the client does not try the operation again.

- If the value of the **COPYCONTINUE** parameter is **YES**, the server tries to remove the copy storage pools that use this device class until enough mount points can be acquired. The transaction fails, and the client tries the operation again. If sufficient mount points can be acquired when the operation is tried again, the data is written into the primary storage pool, any remaining copy storage pools, and any active-data pools, if they exist.

Restrictions:

- The setting of the **COPYCONTINUE** parameter does not affect active-data pools. If a write failure occurs for any of 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 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.

Storing data without using the simultaneous-write function

Writing data simultaneously to copy storage pools and active-data pools might not be an efficient solution for every primary storage pool. When simultaneous-write operations are not practical, use the **BACKUP STGPOOL** and **COPY ACTIVATEDATA** commands to store data in copy storage pools and active-data pools.

Suppose you use a **DISK** primary storage pool that is accessed by many clients at the same time during client data-storage operations. If this storage pool is associated with copy storage pools, active-data pools, or both, the clients might have to wait until enough tape drives are available to perform the store operation. In this scenario, simultaneous-write operations could extend the amount of time required for client store operations. It might be more efficient to store the data in the primary storage pool and use the **BACKUP STGPOOL** command to back up the **DISK** storage pool to the copy storage pools and the **COPY ACTIVATEDATA** command to copy active backup data from the **DISK** storage pool to the active-data pools.

Reducing the potential for switching storage pools during simultaneous-write operations

Switching primary storage pools can delay the completion of a simultaneous-write operation. To reduce the potential for switching, ensure that enough space is available in the primary storage pools and that the pools can accommodate files of any size.

Resources such as disk space, tape drives, and tapes are allocated at the beginning of a simultaneous-write operation, and typically remain allocated during the entire operation. If, for any reason, the destination primary pool cannot contain the data being stored, the IBM Tivoli Storage Manager server attempts to store the data into a next storage pool in the storage hierarchy. This next storage pool typically uses a

sequential-access device class. If new resources must be acquired for the next storage pool, or the allocated resources must be released because the server has to wait to acquire the new resources, the client session must wait until the resources are available.

To reduce the potential for switching storage pools, follow these guidelines:

- Ensure that enough space is available in the primary storage pools that are targets for the simultaneous-write operation. For example, to make space available, run the server migration operation before backing up or archiving client data and before migration operations by Hierarchical Storage Management (HSM) clients.
- The **MAXSIZE** parameter on the **DEFINE STGPOOL** and **UPDATE STGPOOL** commands limits the size of the files that the Tivoli Storage Manager server can store in the primary storage pools during client operations. Honoring the **MAXSIZE** parameter for a storage pool during a store operation causes the server to switch pools. To prevent switching pools, avoid using this parameter if possible.

Separate storage hierarchies for simultaneous-write operations

When using the simultaneous-write function as part of a backup strategy, separate data into different storage pool hierarchies.

For example, you can configure production servers to store mission critical data in one storage pool hierarchy and use the simultaneous-write function to back up the data to copy storage pools and an active-data pool. See Figure 37. In addition, you can configure the servers to store noncritical, workstation data in another storage pool hierarchy and back up that data using the **BACKUP STGPOOL** command.

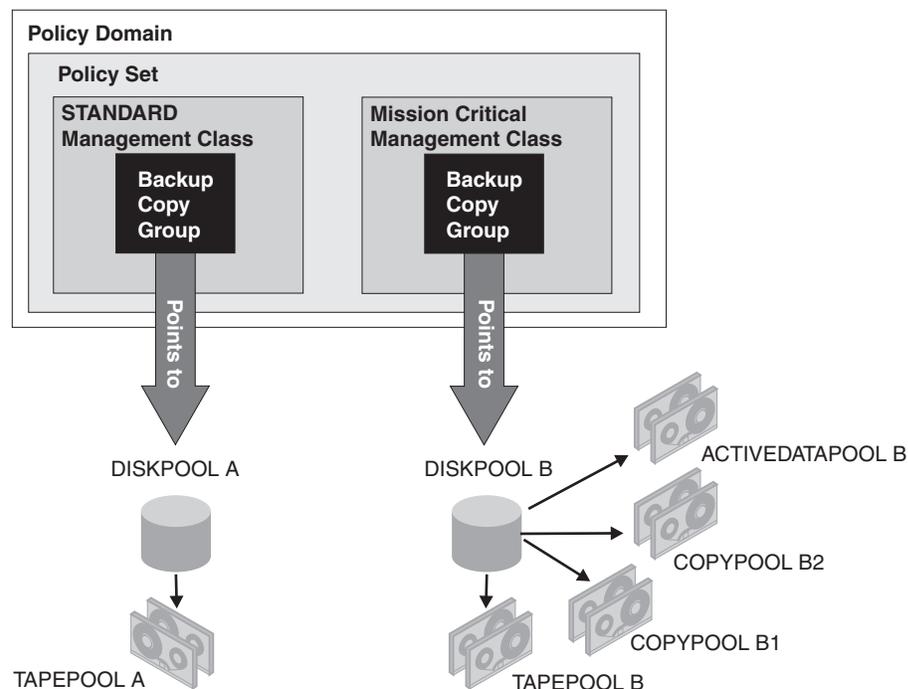


Figure 37. Separate storage pool hierarchies for different types of data

Simultaneous-write function as part of a backup strategy: Example

The simultaneous-write function is used to create on-site backups of a storage pool for easy availability. The BACKUP STGPOOL command is used to create storage pool backups and database backups that are moved off-site to provide data protection in case a disaster occurs.

This example also shows how to use the COPY ACTIVATEDATA command to copy active data from primary storage pools to an on-site sequential-access disk (FILE) active-data pool. When designing a backup strategy, carefully consider your own system, data storage, and disaster-recovery requirements.

1. Define the following storage pools:
 - Two copy storage pools, ONSITECOPYPOOL and DRCOPYPOOL
 - One active-data pool, ACTIVEDATAPOOL
 - Two primary storage pools, DISKPOOL and TAPEPOOL

As part of the storage pool definition for DISKPOOL, specify TAPEPOOL as the next storage pool, ONSITECOPYPOOL as the copy storage pool, and ACTIVEDATAPOOL as the active-data pool. Set the copy continue parameter for copy storage pools to YES. If an error occurs writing to a copy storage pool, the operation will continue storing data into the primary pool, the remaining copy storage pool, and the active-data pool.

```
define stgpool tapepool mytapedevice
define stgpool onsitepool mytapedevice
define stgpool drcopypool mytapedevice
define stgpool activedatapool mydiskdevice
define stgpool diskpool mydiskdevice nextstgpool=tapepool
        copystgpool=onsitecopypool copycontinue=yes activedatapools=
        activedatapool
```

This basic configuration is like the configuration that is shown in Figure 22 on page 293.

2. Schedule or issue the following commands to ensure that all the files are backed up:

```
backup stgpool diskpool onsitecopypool
backup stgpool tapepool onsitecopypool
copy activedata diskpool activedatapool
copy activedata tapepool activedatapool
```
3. To create the storage-pool backup volumes that to be moved off-site, schedule the following two commands to run every night:

```
backup stgpool diskpool drcopypool
backup stgpool tapepool drcopypool
```
4. Every night, after the storage pool backups have completed, back up the database.
5. To process the database and storage pool backups for off-site storage, issue the following command every night:

```
move drmedia copystgpool=drcopypool wherestate=mountable tostate=vault wait=yes
```
6. Start migration of the files in the DISKPOOL to ensure that sufficient space is available in DISKPOOL in preparation for the next storage operations:

```
migrate stgpool diskpool
```

Keeping client files together using collocation

With collocation enabled, the server attempts to keep files belonging to a group of client nodes, a single client node, or client file space on a minimal number of sequential-access storage volumes. Collocation reduces the number of volume mounts required when users restore, retrieve, or recall a large number of files from the storage pool. Collocation thus reduces the amount of time required for these operations.

You can set collocation for each sequential-access storage pool when you define or update the pool.

Figure 38 shows an example of collocation by client node with three clients, each having a separate volume containing that client's data.

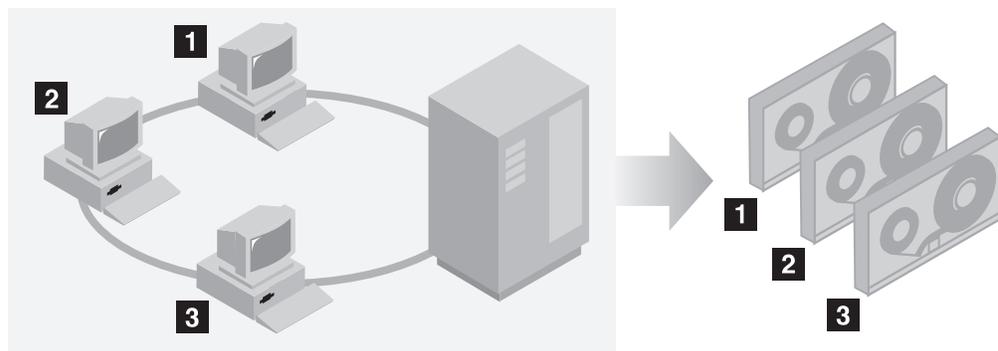


Figure 38. Example of collocation enabled

Figure 39 shows an example of collocation by group of client nodes. Three groups have been defined, and the data for each group is stored on separate volumes.

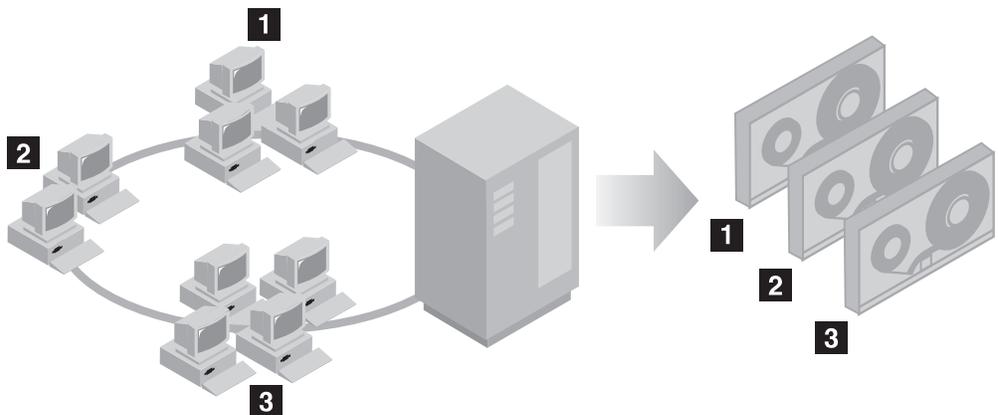


Figure 39. Example of collocation enabled

When collocation is disabled, the server attempts to use all available space on each volume before selecting a new volume. While this process provides better utilization of individual volumes, user files can become scattered across many volumes. Figure 40 on page 311 shows an example of collocation disabled, with three clients sharing space on single volume.

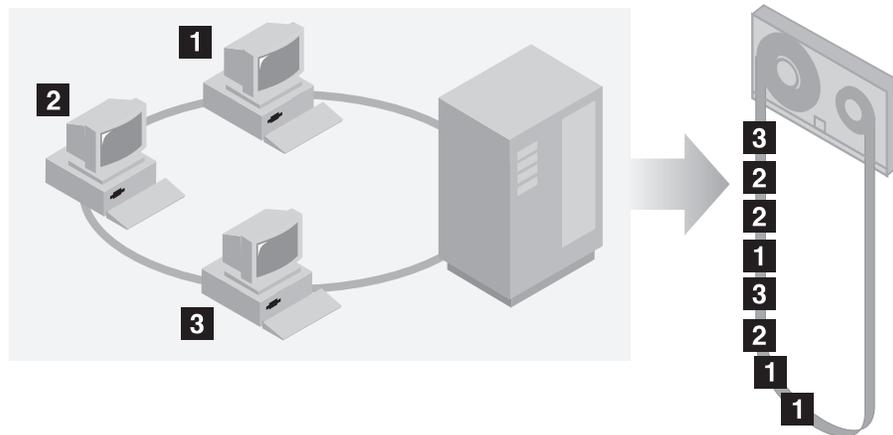


Figure 40. Example of collocation disabled

With collocation disabled, more media mount operations might be required to mount volumes when users restore, retrieve, or recall a large number of files.

Collocation by group is the Tivoli Storage Manager system default for primary sequential-access storage pools. The default for copy storage pools and active-data pools is *no collocation*.

The effects of collocation on operations

The effect of collocation on resources and system performance depends on the type of operation that is being performed.

Table 35 summarizes the effects of collocation on operations.

Table 35. Effect of collocation on operations

Operation	Collocation Enabled	Collocation Disabled
Backing up, archiving, or migrating client files	More media mounts to collocate files.	Usually fewer media mounts are required.
Restoring, retrieving or recalling client files	Large numbers of files can be restored, retrieved, or recalled more quickly because files are located on fewer volumes.	Multiple mounts of media may be required for a single user because files may be spread across multiple volumes. More than one user's files can be stored on the same sequential-access storage volume. For example, if two users attempt to recover a file that resides on the same volume, the second user will be forced to wait until the first user's files are recovered.
Storing data on tape	The server attempts to use all available tape volumes to separate user files before it uses all available space on every tape volume.	The server attempts to use all available space on each tape volume before using another tape volume.

Table 35. Effect of collocation on operations (continued)

Operation	Collocation Enabled	Collocation Disabled
Media mount operations	<p>More mount operations when user files are backed up, archived, or migrated from client nodes directly to sequential-access volumes.</p> <p>More mount operations during reclamation and storage pool migration.</p> <p>More volumes to manage because volumes are not fully used.</p>	More mount operations required during restore, retrieve, and recall of client files.
Generating backup sets	Less time spent searching database entries and fewer mount operations.	More time spent searching database entries and fewer mount operations.

During the following server operations, all the data belonging to a collocation group, a single client node, or a single client file space is moved or copied by one process: For example, if data is collocated by group, all data for all nodes belonging to the same collocation group is migrated by the same process.

1. Moving data from random-access and sequential-access volumes
2. Moving node data from sequential-access volumes
3. Backing up a random-access or sequential-access storage pool
4. Restoring a sequential-access storage pool
5. Reclamation of a sequential-access storage pool or off-site volumes
6. Migration from a random-access storage pool.

When collocating node data, the Tivoli Storage Manager server attempts to keep files together on a minimal number of sequential-access storage volumes. However, when the server is backing up data to volumes in a sequential-access storage pool, the backup process has priority over collocation settings. As a result, the server completes the backup, but might not be able to collocate the data. For example, suppose you are collocating by node, and you specify that a node can use two mount points on the server. Suppose also that the data being backed up from the node could easily fit on one tape volume. During backup, the server might mount two tape volumes, and the node's data might be distributed across two tapes, rather than one.

If collocation is by node or file space, nodes or file spaces are selected for migration based on the amount of data to be migrated. The node or file space with the most data is migrated first. If collocation is by group, all nodes in the storage pool are first evaluated to determine which node has the most data. The node with the most data is migrated first along with all the data for all the nodes belonging to that collocation group regardless of the amount of data in the nodes' file spaces or whether the low migration threshold has been reached.

One reason to collocate by group is that individual client nodes often do not have sufficient data to fill high-capacity tape volumes. Collocating data by groups of nodes can reduce unused tape capacity by putting more collocated data on individual tapes. In addition, because all data belonging to all nodes in the same collocation group are migrated by the same process, collocation by group can reduce the number of times a volume containing data to be migrated needs to be mounted. Collocation by group can also minimize database scanning and reduce tape passes during data transfer from one sequential-access storage pool to

another.

How the server selects volumes with collocation enabled

Volume selection depends on whether collocation is by group, by node, or by file space.

Table 36 shows how the Tivoli Storage Manager server selects the first volume when collocation is enabled for a storage pool at the client-node, collocation-group, and file-space level.

Table 36. How the server selects volumes when collocation is enabled

Volume Selection Order	When collocation is by group	When collocation is by node	When collocation is by file space
1	A volume that already contains files from the collocation group to which the client belongs	A volume that already contains files from the same client node	A volume that already contains files from the same file space of that client node
2	An empty predefined volume	An empty predefined volume	An empty predefined volume
3	An empty scratch volume	An empty scratch volume	An empty scratch volume
4	A volume with the most available free space among volumes that already contain data	A volume with the most available free space among volumes that already contain data	A volume containing data from the same client node
5	Not applicable	Not applicable	A volume with the most available free space among volumes that already contain data

When the server needs to continue to store data on a second volume, it uses the following selection order to acquire additional space:

1. An empty predefined volume
2. An empty scratch volume
3. A volume with the most available free space among volumes that already contain data
4. Any available volume in the storage pool

When collocation is by client node or file space, the server attempts to provide the best use of individual volumes while minimizing the mixing of files from different clients or file spaces on volumes. This is depicted in Figure 41 on page 314, which shows that volume selection is *horizontal*, where all available volumes are used before all available space on each volume is used. A, B, C, and D represent files from four different client nodes.

Remember:

1. If collocation is by node and the node has multiple file spaces, the server does not attempt to collocate those file spaces.
2. If collocation is by file space and a node has multiple file spaces, the server attempts to put data for different file spaces on different volumes.

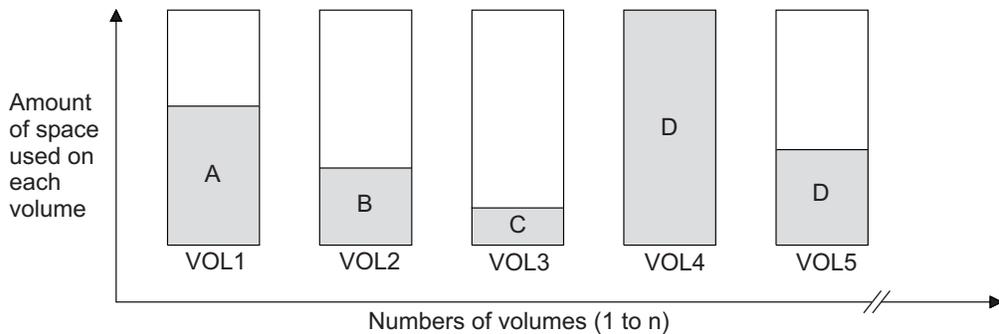


Figure 41. Using all available sequential access storage volumes with collocation enabled at the group or file space level

When collocation is by group, the server attempts to collocate data from nodes belonging to the same collocation group. As shown in the Figure 42, data for the following groups of nodes has been collocated:

- Group 1 consists of nodes A, B, and C
- Group 2 consists of nodes D and E
- Group 3 consists of nodes F, G, H, and I

Whenever possible, the Tivoli Storage Manager server collocates data belonging to a group of nodes on a single tape, as represented by Group 2 in the figure. Data for a single node can also be spread across several tapes associated with a group (Group 1 and 2). If the nodes in the collocation group have multiple file spaces, the server does not attempt to collocate those file spaces.

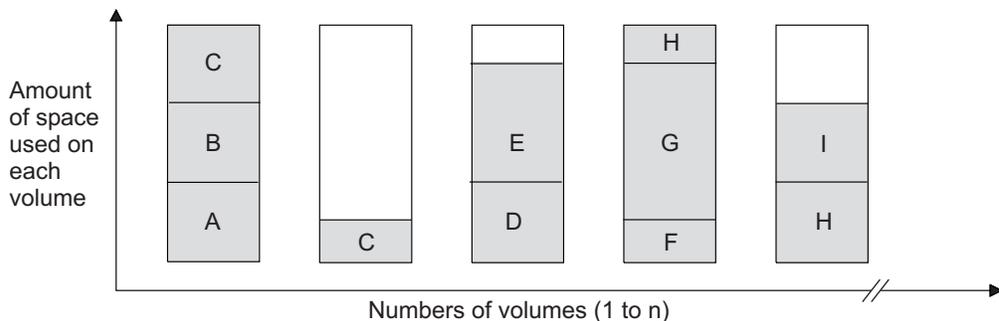


Figure 42. Using all available sequential access storage volumes with collocation enabled at the group level

Remember: Normally, the Tivoli Storage Manager server always writes data to the current filling volume for the operation being performed. Occasionally, however, you might notice more than one filling volume in a collocated storage pool. This can occur if different server processes or client sessions attempt to store data into the collocated pool at the same time. In this situation, Tivoli Storage Manager will allocate a volume for each process or session needing a volume so that both operations complete as quickly as possible.

How the server selects volumes with collocation disabled

When collocation is disabled, the server attempts to use all available space in a storage volume before it accesses another volume.

When storing client files in a sequential-access storage pool where collocation is disabled, the server selects a volume using the following selection order:

1. A previously used sequential volume with available space (a volume with the most amount of data is selected first)
2. An empty volume

When the server needs to continue to store data on a second volume, it attempts to select an empty volume. If none exists, the server attempts to select any remaining available volume in the storage pool.

Figure 43 shows that volume utilization is *vertical* when collocation is disabled. In this example, fewer volumes are used because the server attempts to use all available space by mixing client files on individual volumes. A, B, C, and D represent files from four different client nodes.

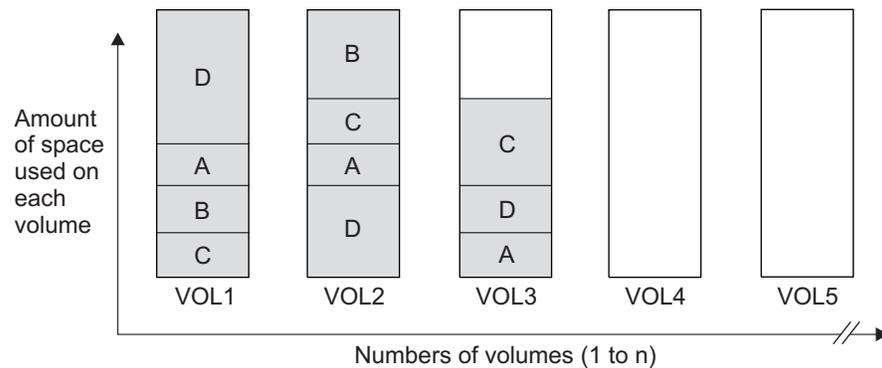


Figure 43. Using all available space on sequential volumes with collocation disabled

Collocation on or off settings

After you define a storage pool, you can change the collocation setting by updating the storage pool. The change in collocation for the pool does not affect files that are already stored in the pool.

For example, if collocation is off for a storage pool and you turn it on, *from then on* client files stored in the pool are collocated. Files that had previously been stored in the pool are *not* moved to collocate them. As volumes are reclaimed, however, the data in the pool tends to become more collocated. You can also use the MOVE DATA or MOVE NODEDATA commands to move data to new volumes to increase collocation. However, this causes an increase in the processing time and the volume mount activity.

Remember: A mount wait can occur or increase when collocation by file space is enabled and a node has a volume containing multiple file spaces. If a volume is eligible to receive data, Tivoli Storage Manager will wait for that volume.

Collocation of copy storage pools and active-data pools

Using collocation on copy storage pools and active-data pools requires special consideration. Collocation of copy storage pools and active-data pools, especially by node or file space, results in more partially filled volumes and potentially unnecessary off-site reclamation activity.

Using collocation on copy storage pools and active-data pools requires special consideration.

Primary storage pools perform a different recovery role than those performed by copy storage pools and active-data pools. Normally you use primary storage pools (or active-data pools) to recover data to clients directly. In a disaster, when *both* clients *and* the server are lost, you might use off-site active-data pool volumes to recover data directly to clients and the copy storage pool volumes to recover the primary storage pools. The types of recovery scenarios that concern you the most will help you to determine whether to use collocation on your copy storage pools and active-data pools.

Collocation typically results in partially filled volumes when you collocate by node or by file space. (Partially filled volumes are less prevalent, however, when you collocate by group.) Partially filled volumes might be acceptable for primary storage pools because the volumes remain available and can be filled during the next migration process. However, this may be unacceptable for copy storage pools and active-data pools whose storage pool volumes are taken off-site immediately. If you use collocation for copy storage pools or active-data pools, you must decide among the following:

- Taking more partially filled volumes off-site, thereby increasing the reclamation activity when the reclamation threshold is lowered or reached. Remember that rate of reclamation for volumes in an active-data pool is typically faster than the rate for volumes in other types of storage pools.
- Leaving these partially filled volumes on-site until they fill and risk not having an off-site copy of the data on these volumes.
- Whether to collocate by group in order to use as much tape capacity as possible

With collocation disabled for a copy storage pool or an active-data pool, typically there will be only a few partially filled volumes after data is backed up to the copy storage pool or copied to the active-data pool.

Consider carefully before using collocation for copy storage pools and active-data pools. Even if you use collocation for your primary storage pools, you may want to disable collocation for copy storage pools and active-data pools. Collocation on copy storage pools or active-data pools might be desirable if you have few clients, but each of them has large amounts of incremental backup data each day.

Planning for and enabling collocation

Understanding the effects of collocation can help reduce the number of media mounts, make better use of space on sequential volumes, and improve the efficiency of server operations.

Table 37 on page 317 lists the four collocation options that you can specify on the DEFINE STGPOOL and UPDATE STGPOOL commands. The table also describes the effects of collocation on data belonging to nodes that are members of collocation groups and nodes that are not members of any collocation group.

Table 37. Collocation options and effects on node data

Collocation option	If a node is not defined as a member of a collocation group...	If a node is defined as a member of a collocation group...
No	The node's data is not collocated.	The node's data is not collocated.
Group	The server stores the node's data on as few volumes in the storage pool as possible.	The server stores the data for the node and for other nodes that belong to the same collocation group on as few volumes as possible.
Node	The server stores the node's data on as few volumes as possible.	The server stores the node's data on as few volumes as possible.
Filespace	The server stores the data for the node's file space on as few volumes as possible. If a node has multiple file spaces, the server stores the data for different file spaces on different volumes in the storage pool.	The server stores the data for the node's file space on as few volumes as possible. If a node has multiple file spaces, the server stores the data for different file spaces on different volumes in the storage pool.

When deciding whether and how to collocate data, do the following:

1. Familiarize yourself with the potential advantages and disadvantages of collocation, in general. For a summary of effects of collocation on operations, see Table 35 on page 311.
2. If the decision is to collocate, determine how data should be organized, whether by client node, group of client nodes, or file space. If the decision is to collocate by group, you need to decide how to group nodes:
 - If the goal is to save space, you may wish to group small nodes together to better use tapes.
 - If the goal is potentially faster client restores, group nodes together so that they fill as many tapes as possible. Doing so increases the probability that individual node data will be distributed across two or more tapes and that more tapes can be mounted simultaneously during a multi-session No Query Restore operation.
 - If the goal is to departmentalize data, then you can group nodes by department.
3. If collocation by group is the desired result:
 - a. Define collocation groups using the `DEFINE COLLOCGROUP` command.
 - b. Add client nodes to the collocation groups using the `DEFINE COLLOCGROUPMEMBER` command.

The following query commands are available to help in collocating groups:

QUERY COLLOCGROUP

Displays the collocation groups defined on the server.

QUERY NODE

Displays the collocation group, if any, to which a node belongs.

QUERY NODEDATA

Displays information about the data for one or more nodes in a sequential-access storage pool.

QUERY STGPOOL

Displays information about the location of client data in a sequential-access storage pool and the amount of space a node occupies in a volume.

For more information about these commands, refer to the *Administrator's Reference*.

You can also use Tivoli Storage Manager server scripts or PERL scripts to display information that can be useful in defining collocation groups.

4. Specify how data is to be collocated in a storage pool using the COLLOCATE parameter on the DEFINE STGPOOL or UPDATE STGPOOL command.
5. If you decide later that you want to delete members of a collocation group, you can use the DELETE COLLOCMEMBER command. You can also update the description of a collocation group using the UPDATE COLLOGROUP command and delete entire collocation groups using the DELETE COLLOGROUP command.

Tip: If you use collocation, but want to reduce the number of media mounts and use space on sequential volumes more efficiently, you can do the following:

- Define a storage pool hierarchy and policy to require that backed-up, archived, or space-managed files are stored initially in disk storage pools.

When files are migrated from a disk storage pool, the server attempts to migrate all files belonging to the client node or collocation group that is using the most disk space in the storage pool. This process works well with the collocation option because the server tries to place all of the files from a given client on the same sequential-access storage volume.

- Use scratch volumes for sequential-access storage pools to allow the server to select new volumes for collocation.
- Specify the client option COLLOCATEBYFILESPEC to limit the number of tapes to which objects associated with one file specification are written. This collocation option makes collocation by the server more efficient; it does not override collocation by file space or collocation by node. For general information about client options, see "Managing client option files" on page 411. For details about the COLLOCATEBYFILESPEC option, refer to the *Backup-Archive Clients Installation and User's Guide*.

When creating collocation groups, keep in mind that the ultimate destination of the data belonging to nodes in a collocation group depends on the policy domain to which nodes belong. For example, suppose you create a collocation group consisting of nodes that belong to Policy Domain A. Policy Domain A specifies an active-data pool as the destination of active data only and has a backup copy group that specifies a primary storage pool, Primary1, as the destination for active and inactive data. Other nodes in the same collocation group belong to a domain, Policy Domain B, that does not specify an active-data pool, but that has a backup copy group that specifies Primary1 as the destination for active and inactive data. Primary1 has a designated copy storage pool. The collocation setting on PRIMARY1, the copy storage pool, and the active-data pool is GROUP.

When the node data is backed up and a simultaneous-write operation occurs, active and inactive data is stored in Primary1 and the copy storage pool. Note, however, that although all the nodes belong to a single collocation group, only the active data belonging to nodes in Domain A are stored in the active-data pool. The data in Primary1 and the copy storage pool is collocated by group. The data in the active-data pool is also collocated by group, but the "group" consists only of nodes that are members of Policy Domain A.

Reclaiming space in sequential-access storage pools

Space on a sequential-access storage volume becomes reclaimable as files expire or are deleted from the volume. Reclamation processing involves consolidating the remaining data from many sequential-access volumes onto fewer new sequential-access volumes.

Files become obsolete because of aging or limits on the number of versions of a file. Space in volumes in active-data pools also becomes reclaimable as updated files are added to the pools and as older file versions are deactivated. In reclamation processing, the server rewrites files on the volume being reclaimed to other volumes in the storage pool, making the reclaimed volume available for reuse.

The server reclaims the space in storage pools based on a *reclamation threshold* that you can set for each sequential-access storage pool. When the percentage of space that can be reclaimed on a volume rises above the reclamation threshold, the server reclaims the volume.

Restrictions:

- Storage pools defined with the NETAPPDUMP, the CELERRADUMP or the NDMPDUMP data format cannot be reclaimed. However, you can use the MOVE DATA command to move data out of a volume so that the volume can be reused. The volumes in the target storage pool must have the same data format as the volumes in the source storage pool.

How Tivoli Storage Manager reclamation works

You can set a reclamation threshold for a sequential-access storage pool when you define or update the pool. When the percentage of reclaimable space on a volume exceeds the reclamation threshold set for the storage pool, the volume is eligible for reclamation.

The server checks whether reclamation is needed at least once per hour and begins space reclamation for eligible volumes. During space reclamation, the server copies files that remain on eligible volumes to other volumes. For example, Figure 44 on page 320 shows that the server consolidates the files from tapes 1, 2, and 3 on tape 4. During reclamation, the server copies the files to volumes in the same storage pool unless you have specified a reclamation storage pool. Use a reclamation storage pool to allow automatic reclamation for a storage pool with only one drive.

Remember: To prevent contention for the same tapes, the server does not allow a reclamation process to start if a DELETE FILESPACE process is active. The server checks every hour for whether the DELETE FILESPACE process has completed so that the reclamation process can start. After the DELETE FILESPACE process has completed, reclamation begins within one hour.

The server also reclaims space within an aggregate. An aggregate is a physical file that contains multiple logical files that are backed up or archived from a client in a single transaction. Space within the aggregate becomes reclaimable space as logical files in the aggregate expire, as files are deleted by the client, or as files become deactivated in active-data pools. The server removes unused space as the server copies the aggregate to another volume during reclamation processing. However, reclamation does not aggregate files that were originally stored in non-aggregated form. Reclamation also does not combine aggregates to make new aggregates. You can also reclaim space in an aggregate by issuing the MOVE DATA command. See

“Reclaiming space in aggregates by moving data” on page 353 for details.

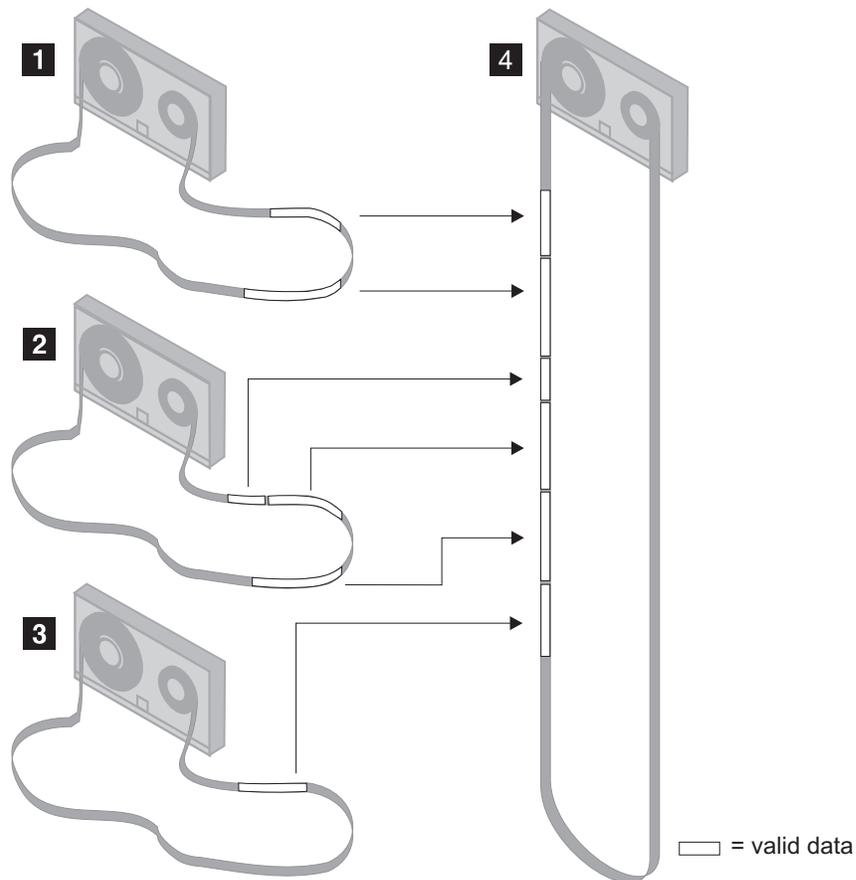


Figure 44. Tape reclamation

After the server moves all readable files to other volumes, one of the following occurs for the reclaimed volume:

- If you have explicitly defined the volume to the storage pool, the volume becomes available for reuse by that storage pool
- If the server acquired the volume as a scratch volume, the server deletes the volume from the Tivoli Storage Manager database

Volumes that have a device type of SERVER are reclaimed in the same way as other sequential-access volumes. However, because the volumes are actually data stored in the storage of another Tivoli Storage Manager server, the reclamation process can consume network resources. See “Controlling reclamation of virtual volumes” on page 324 for details about how the server reclaims these types of volumes.

Volumes in a copy storage pool and active-data pools are reclaimed in the same manner as a primary storage pool except for the following:

- Off-site volumes are handled differently.
- The server copies active files from the candidate volume only to other volumes in the *same* storage pool.

For details, see “Reclaiming copy storage pools and active-data pools” on page 325.

Reclamation thresholds

Space is reclaimable because it is occupied by files that have been expired or deleted from the Tivoli Storage Manager database, or because the space has never been used. The reclamation threshold indicates how much reclaimable space a volume must have before the server reclaims the volume.

The server checks whether reclamation is needed at least once per hour. The lower the reclamation threshold, the more frequently the server tries to reclaim space. Frequent reclamation optimizes the use of a sequential-access storage pool's space, but can interfere with other processes, such as backups from clients.

If the reclamation threshold is high, reclamation occurs less frequently. A high reclamation threshold is useful if mounting a volume is a manual operation and the operations staff is at a minimum. Setting the reclamation threshold to 100% prevents automatic reclamation from occurring. You might want to do this to control when reclamation occurs, to prevent interfering with other server processes. When it is convenient for you and your users, you can use the RECLAIM STGPOOL command to invoke reclamation, or you can lower the reclamation threshold to cause reclamation to begin.

If you set the reclamation threshold to 50% or greater, the server can combine the usable files from two or more volumes onto a single new volume.

Reclamation of volumes in an active-data pool usually returns volumes to scratch status more frequently than reclamation of volumes in non-active-data pools. This is because the percentage of reclaimable space for sequential volumes in active-data pools reflects not only the space of deleted files, but also the space of inactive files. Frequent reclamation requires more resources such as tape drives and libraries to mount and dismount volumes.

If reclamation is occurring too frequently in your active-data pools, you can increase the reclamation thresholds until the rate of reclamation is acceptable. Accelerated reclamation of volumes has more of an effect on active-data pools that use removable media and, in particular, on removable media that is taken off-site.

Reclaiming volumes with the most reclaimable space

If you have been running with a high reclamation threshold and decide you need to reclaim volumes, you can lower the threshold in several steps. Lowering the threshold in steps ensures that volumes with the most reclaimable space are reclaimed first.

For example, if you set the reclamation threshold to 100%, first lower the threshold to 98%. Volumes that have reclaimable space of 98% or greater are reclaimed by the server. Lower the threshold again to reclaim more volumes.

If you lower the reclamation threshold while a reclamation process is active, the reclamation process does not immediately stop. If an on-site volume is being reclaimed, the server uses the new threshold setting when the process begins to reclaim the next volume. If off-site volumes are being reclaimed, the server does not use the new threshold setting during the process that is running (because all eligible off-site volumes are reclaimed at the same time).

Use the CANCEL PROCESS command to stop a reclamation process.

Starting reclamation manually or in a schedule

To gain more control over how and when the reclamation process occurs, you can use the RECLAIM STGPOOL command. You can also specify the maximum amount of time a reclamation process will take before it is automatically canceled.

To perform reclamation when it is least intrusive to normal production needs, include the RECLAIM STGPOOL command in a schedule. For example, to start reclamation in a storage pool named ALTPPOOL, and to have reclamation end as soon as possible after one hour, you would issue the following command:

```
reclaim stgpool altpool duration=60
```

For copy storage pools and active-data pools, you can also use the RECLAIM STGPOOL command to specify the maximum number of off-site storage pool volumes the server should attempt to reclaim:

```
reclaim stgpool altpool duration=60 offsitereclaimlimit=230
```

Do not use this command if you are going to use automatic reclamation for the storage pool. To prevent automatic reclamation from running, set the RECLAIM parameter of the storage pool definition to 100.

For details about the RECLAIM STGPOOL command, refer to the *Administrator's Reference*.

Optimizing drive usage using multiple concurrent reclamation processes

Multiple reclamation processes run concurrently, allowing you to make better use of your available tape drives or FILE volumes.

You can specify one or more reclamation processes for each primary sequential-access storage pool, copy storage pool, or active-data pool using the RECLAIMPROCESS parameter on the DEFINE STGPOOL and UPDATE STGPOOL commands.

Each reclamation process requires at least two simultaneous volume mounts (at least two mount points) and, if the device type is not FILE, at least two drives. One of the drives is for the input volume in the storage pool being reclaimed. The other drive is for the output volume in the storage pool to which files are being moved.

When calculating the number of concurrent processes to run, you must carefully consider the resources you have available, including the number of storage pools that will be involved with the reclamation, the number of mount points, the number of drives that can be dedicated to the operation, and (if appropriate) the number of mount operators available to manage reclamation requests. 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 storage pools that are involved in the reclamation. For more information about mount limit, see:

“Controlling the number of simultaneously mounted volumes” on page 202

For example, suppose that you want to reclaim the volumes from two sequential storage pools simultaneously and that all storage pools involved have the same device class. Each process requires two mount points and, if the device type is not FILE, two drives. To run four reclamation processes simultaneously (two for each

storage pool), you need a total of at least eight mount points and eight drives. The device class for each storage pool must have a mount limit of at least eight.

If the device class for the storage pools being reclaimed does not have enough mount points or drives, you can use the RECLAIMSTGPOOL parameter to direct the reclamation to a storage pool with a different device class that has the additional mount points or drives.

If the number of reclamation processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait indefinitely or until the other reclamation processes complete and mount points or drives become available.

The Tivoli Storage Manager server will start the specified number of reclamation processes regardless of the number of volumes that are eligible for reclamation. For example, if you specify ten reclamation processes and only six volumes are eligible for reclamation, the server will start ten processes and four of them will complete without processing a volume.

Multiple concurrent reclamation processing does not affect collocation. For additional information, see “How collocation affects reclamation” on page 329.

Reclaiming volumes in a storage pool with one drive

When a storage pool has only one mount point (that is, just one drive) available to it through the device class, data cannot be reclaimed from one volume to another within that same storage pool. To reclaim volumes in a storage pool that has only one drive, you can define a *reclamation storage pool* and use it for temporary storage of reclaimed data.

When the server reclaims volumes, the server moves the data from volumes in the original storage pool to volumes in the reclamation storage pool. The server always uses the reclamation storage pool when one is defined, even when the mount limit is greater than one.

If the reclamation storage pool does not have enough space to hold all of the data being reclaimed, the server moves as much of the data as possible into the reclamation storage pool. Any data that could not be moved to volumes in the reclamation storage pool still remains on volumes in the original storage pool.

The pool identified as the reclamation storage pool must be a primary sequential storage pool. The primary purpose of the reclamation storage pool is for temporary storage of reclaimed data. To ensure that data moved to the reclamation storage pool eventually moves back into the original storage pool, specify the original storage pool as the next pool in the storage hierarchy for the reclamation storage pool. For example, if you have a tape library with one drive, you can define a storage pool to be used for reclamation using a device class with a device type of FILE:

```
define stgpool reclaimpool fileclass maxscratch=100
```

Define the storage pool for the tape drive as follows:

```
define stgpool tapepool1 tapeclass maxscratch=100  
reclaimstgpool=reclaimpool
```

Finally, update the reclamation storage pool so that data migrates back to the tape storage pool:

```
update stgpool reclaimpool nextstgpool=tapepool1
```

Tip:

- You can specify multiple concurrent reclamation processes for a primary storage pool with one drive by using the RECLAIMSTGPOOL parameter. If multiple concurrent processing is not desired, specify a value of 1 for the RECLAIMPROCESS parameter on the DEFINE STGPOOL or UPDATE STGPOOL commands.
- In a mixed-media library, reclaiming volumes in a storage pool defined with a device class with a single mount point (that is, a single drive) requires one of the following:
 - At least one other drive with a compatible read/write format
 - Enough disk space to create a storage pool with a device type of FILE

Reducing the time to reclaim tape volumes with high capacity

When a storage pool uses tape volumes with high capacity, reclamation processes might run for a long time if the drives are relatively slow at positioning tapes. There are steps that you can take to reduce overall process time.

To help reduce overall process time:

1. Set up the storage pool hierarchy so that the tape storage pool is the next storage pool for a storage pool that uses either a DISK device type or a FILE device type.
2. When you need to reclaim volumes, move data from the tape storage pool to the DISK or FILE storage pool.
3. Allow the data to migrate from the DISK or FILE storage pool back to the tape storage pool by adjusting the migration thresholds.

Controlling reclamation of virtual volumes

When virtual volumes (volumes with the device type of SERVER) in a primary storage pool are reclaimed, the client data stored on those volumes is sent across the network between the source server and the target server. As a result, the reclamation process can tie up your network resources.

To control when reclamation starts for these volumes, consider setting the reclamation threshold to 100% for any primary storage pool that uses virtual volumes. Lower the reclamation threshold at a time when your network is less busy, so that the server can reclaim volumes.

For virtual volumes in a copy storage pool or an active-data pool, the server reclaims a volume as follows:

1. The source server determines which files on the volume are still valid.
2. The source server obtains these valid files from volumes in a primary storage pool, or if necessary, from removable-media volumes in an on-site copy storage pool or in an on-site active-data pool. The server can also obtain files from virtual volumes in a copy storage pool or an active-data pool.
3. The source server writes the files to one or more new virtual volumes in the copy storage pool or active-data pool and updates its database.
4. The server issues a message indicating that the volume was reclaimed.

Tip: You can specify multiple concurrent reclamation processes for a primary storage pool with a device type of SERVER. However, running multiple concurrent

processes for this type of storage pool can tie up network resources because the data is sent across the network between the source server and target server. Therefore, if you want to run multiple concurrent processes, do so when the network is less busy. If multiple concurrent processing is not desired, specify a value of 1 for the RECLAIMPROCESS parameter on the DEFINE STGPOOL or UPDATE STGPOOL commands.

For information about using the SERVER device type, see “Using virtual volumes to store data on another server” on page 712.

Reclaiming copy storage pools and active-data pools

On-site and off-site volumes in copy storage pools and active-data pools are reclaimed when the amount of unused space exceeds the reclamation threshold. When reclamation occurs and how reclamation processing is done depends on whether the volumes are marked as off-site.

Reclamation of volumes in copy storage pools and active-data pools is similar to reclamation in primary storage pools. For volumes that are on-site, reclamation usually occurs after the volume is full and then begins to empty because of file deletion, expiration, or, in the case of active-data pools, deactivation. When the percentage of reclaimable space on a volume rises above the reclamation threshold, the server reclaims the volume. Active files on the volume are rewritten to other volumes in the storage pool, making the original volume available for new files.

For off-site volumes, reclamation can occur when the percentage of unused space on the volume is greater than the reclaim parameter value. The unused space in copy storage pool volumes includes both space that has never been used on the volume and space that has become empty because of file deletion or expiration. For volumes in active-data pools, reclaimable space also includes inactive versions of files. Most volumes in copy storage pools and active-data pools might be set to an access mode of off-site, making them ineligible to be mounted. During reclamation, the server copies valid files on off-site volumes from the original files in the primary storage pools. In this way, the server copies valid files on off-site volumes without having to mount these volumes. For more information, see “Reclamation of off-site volumes” on page 326.

Reclamation of copy storage pool volumes and active-data pool volumes should be done periodically to allow the reuse of partially filled volumes that are off-site. Reclamation can be done automatically by setting the reclamation threshold for the copy storage pool or the active-data pool to less than 100%. However, you need to consider controlling when reclamation occurs because of how off-site volumes are treated. For more information, see “Controlling when reclamation occurs for off-site volumes” on page 326.

Virtual Volumes: Virtual volumes (volumes that are stored on another Tivoli Storage Manager server through the use of a device type of SERVER) cannot be set to the off-site access mode.

Using the RECLAIMPROCESS parameter on the DEFINE STGPOOL or UPDATE STGPOOL command, you can specify multiple concurrent reclamation processes for a single copy storage pool or active-data pool. Doing so will let you make better use of your available tape drives or FILE volumes. The principles underlying multiple concurrent reclamation processes for copy storage pools and active-data pools are the same principles as those for primary sequential-access storage pools. In particular, you need to carefully consider available resources (for example, the

number of mount points) when calculating how many processes you can run concurrently. For details, see “Optimizing drive usage using multiple concurrent reclamation processes” on page 322.

Reclamation of primary storage pool volumes does not affect copy storage pool files or files in active-data pools.

Reclamation of off-site volumes

Volumes with the access value of off-site are eligible for reclamation if the amount of empty space on a volume exceeds the reclamation threshold for the copy storage pool or active-data pool. The default reclamation threshold for copy storage pools and active-data pools is 100%, which means that reclamation is not performed.

When an off-site volume is reclaimed, the files on the volume are rewritten to a *read/write* volume. Effectively, these files are moved back to the on-site location. The files may be obtained from the off-site volume after a disaster, if the volume has not been reused and the database backup that you use for recovery references the files on the off-site volume.

The server reclaims an off-site volume as follows:

1. The server determines which files on the volume are still valid.
2. The server obtains these valid files from a primary storage pool or, if necessary, from an on-site volume of a copy storage pool or active-data pool.
3. The server writes the files to one or more volumes in the copy storage pool or active-data pool and then updates the database. If a file is an aggregate with unused space, the unused space is removed during this process.
4. A message is issued indicating that the off-site volume was reclaimed.
For a single storage pool, the server reclaims all off-site volumes that are eligible for reclamation at the same time. Reclaiming all the eligible volumes at the same time minimizes the tape mounts for primary storage pool volumes.

If you are using the disaster recovery manager, see:

“Moving copy storage pool and active-data pool volumes on-site” on page 817

Controlling when reclamation occurs for off-site volumes

If you send copy storage pool volumes off-site, you can control reclamation by adjusting the reclamation threshold.

Suppose you plan to make daily storage pool backups to a copy storage pool, then mark all new volumes in the copy storage pool as *offsite* and send them to the off-site storage location. This strategy works well with one consideration if you are using automatic reclamation (the reclamation threshold is less than 100%).

Each day's storage pool backups will create a number of new copy-storage pool volumes, the last one being only partially filled. If the percentage of empty space on this partially filled volume is higher than the reclaim percentage, this volume becomes eligible for reclamation as soon as you mark it off-site. The reclamation process would cause a new volume to be created with the same files on it. The volume you take off-site would then be empty according to the Tivoli Storage Manager database. If you do not recognize what is happening, you could perpetuate this process by marking the new partially filled volume off-site.

One way to resolve this situation is to keep partially filled volumes on-site until they fill up. However, this would mean a small amount of your data would be without an off-site copy for another day.

If you send copy storage pool volumes off-site, it is recommended you control pool reclamation by using the default value of 100. This turns reclamation off for the copy storage pool. You can start reclamation processing at desired times by changing the reclamation threshold for the storage pool. To monitor off-site volume utilization and help you decide what reclamation threshold to use, enter the following command:

```
query volume * access=offsite format=detailed
```

Depending on your data expiration patterns, you may not need to do reclamation of off-site volumes each day. You may choose to perform off-site reclamation on a less frequent basis. For example, suppose you ship copy-storage pool volumes to and from your off-site storage location once a week. You can run reclamation for the copy-storage pool weekly, so that as off-site volumes become empty they are sent back for reuse.

When you do perform reclamation for off-site volumes, the following sequence is recommended:

1. Back up your primary-storage pools to copy-storage pools or copy the active data in primary-storage pools to active-data pools.
2. Turn on reclamation for copy-storage pools and active-data pools by lowering the reclamation threshold for copy-storage pools below 100%. The default for active-data pools is 60.
3. When reclamation processing completes, turn off reclamation by raising the reclamation thresholds to 100%.
4. Mark any newly created copy-storage pool volumes and active-data pool volumes as off-site, and then move them to the off-site location.

This sequence ensures that the files on the new copy-storage pool volumes and active-data pool volumes are sent off-site, and are not inadvertently kept on-site because of reclamation.

Preventing off-site marking of partially-filled copy storage pool and active-data pool volumes:

To prevent marking partially-filled copy storage pool or active-data pool volumes as off-site, you can use storage on another Tivoli Storage Manager server (device type of SERVER) for storage-pool backups.

If the other server is at a different site, the copy-storage pool volumes or active-data pool volumes are already off-site, with no moving of physical volumes between the sites. See "Using virtual volumes to store data on another server" on page 712 for more information.

Limiting the number of off-site volumes to be reclaimed

To ensure that reclamation completes within the desired amount of time, you can use **OFFSITERECLAIMLIMIT** parameter on the **DEFINE STGPOOL** or **UPDATE STGPOOL** command to limit the number of off-site volumes to be reclaimed.

When determining the value for the **OFFSITERECLAIMLIMIT** parameter, consider using the statistical information in the message issued at the end of the off-site volume reclamation operation.

Alternatively, you can use the following Tivoli Storage Manager SQL **SELECT** command to obtain records from the **SUMMARY** table for the off-site volume reclamation operation:

```
select * from summary where activity='OFFSITE RECLAMATION'
```

Two kinds of records are displayed for the off-site reclamation process. One volume record is displayed for each reclaimed off-site volume. However, the volume record does not display the following items:

- The number of examined files.
- The number of affected files.
- The total bytes involved in the operation.

This information is summarized in the statistical summary record for the offsite reclamation. The statistical summary record displays the following items:

- The number of examined files.
- The number of affected files.
- The total bytes involved in the operation.
- The number of off-site 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 off-site 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.

Delayed reuse of reclaimed volumes

Delaying reuse may help you to recover data under certain conditions during recovery from a disaster.

As a best practice, delay the reuse of any reclaimed volumes in copy storage pools and active-data pools for as long as you keep your oldest database backup. For more information about delaying volume reuse, see “Delaying reuse of volumes for recovery purposes” on page 766.

Reclamation of volumes in active-data pools

Inactive files in volumes in an active-data pool are deleted by reclamation processing. The rate at which reclaimable space accumulates in active-data pool volumes is typically faster than the rate for volumes in non-active-data pools.

If reclamation of volumes in an active-data pool is occurring too frequently, requiring extra resources such as tape drives and libraries to mount and dismount volumes, you can adjust the reclamation threshold until the rate of reclamation is acceptable. The default reclamation threshold for active-data pools is 60 percent, which means that reclamation begins when the storage pool reaches 60 percent of capacity. Accelerated reclamation of volumes has more of an effect on active-data pools that use removable media and, in particular, on removable media that is taken off-site.

How collocation affects reclamation

If collocation is enabled and reclamation occurs, the server tries to move the files for each client node, group of client nodes or client file space onto a minimal number of volumes.

If the volumes are manually mounted, the mount operators must:

- Be aware that a tape volume may be rewound more than once if the server completes a separate pass to move the data for each client node or client file space.
- Mount and dismount multiple volumes to allow the server to select the most appropriate volume on which to move data for each client node or client file space. The server tries to select a volume in the following order:
 1. A volume that already contains files belonging to the client file space or client node
 2. An empty volume
 3. The volume with the most available space
 4. Any available volume

If collocation is disabled and reclamation occurs, the server tries to move usable data to new volumes by using the following volume selection criteria, in the order shown:

1. The volume that contains the most data
2. Any partially full volume
3. An empty predefined volume
4. An empty scratch volume

If you specify collocation and multiple concurrent processes, the server attempts to move the files for each collocation group, client node, or client file space onto as few volumes as possible. However, if files belonging to a single collocation group (or node or file space) are on different volumes to begin with and are being moved

at the same time by different processes, the files could be moved to separate output volumes. For details about multiple concurrent reclamation processing, see “Optimizing drive usage using multiple concurrent reclamation processes” on page 322.

See also “Reducing the time to reclaim tape volumes with high capacity” on page 324.

Estimating space needs for storage pools

Three default random-access disk storage pools are provided at installation. You can add space to these storage pools by adding volumes, or you can define additional storage pools.

The following default random-access disk storage pools are available at installation:

- BACKUPPOOL for backed-up files
- ARCHIVEPOOL for archived files
- SPACEMGPOOL for files migrated from client nodes (space-managed files)

As your storage environment grows, you may want to consider how policy and storage pool definitions affect where workstation files are stored. Then you can define and maintain multiple storage pools in a hierarchy that allows you to control storage costs by using sequential-access storage pools in addition to disk storage pools, and still provide appropriate levels of service to users.

To help you determine how to adjust your policies and storage pools, get information about how much storage is being used (by client node) and for what purposes in your existing storage pools. For more information on how to do this, see “Obtaining information about the use of storage space” on page 345.

Estimating space requirements in random-access storage pools

The amount of storage space required for each random-access disk storage pool is based on your storage needs for backup, archive, and space-management operations.

To estimate the amount of storage space required for each random-access disk storage pool:

- Determine the amount of disk space needed for different purposes:
 - For backup storage pools, provide enough disk space to support efficient daily incremental backups.
 - For archive storage pools, provide sufficient space for a user to archive a moderate size file system without causing migration from the disk storage pool to occur.
 - For storage pools for space-managed files, provide enough disk space to support the daily space-management load from HSM clients, without causing migration from the disk storage pool to occur.
- Decide what percentage of this data you want to keep on disk storage space. Establish migration thresholds to have the server automatically migrate the remainder of the data to less expensive storage media in sequential-access storage pools.

See “Migration thresholds” on page 256 for recommendations on setting migration thresholds.

Estimating space for backed-up files in random-access storage pools

Space requirements for backed-up files stored in a single random-access storage pool are based on the total number of workstations, the average data capacity of a workstation, the fraction of each workstation disk space used, and the number backup versions you will keep.

To estimate the total amount of space needed for all backed-up files stored in a single random-access (disk) storage pool, use the following formula:

$\text{Backup space} = \text{WkstSize} * \text{Utilization} * \text{VersionExpansion} * \text{NumWkst}$

where:

Backup Space

The total amount of storage pool disk space needed.

WkstSize

The average data storage capacity of a workstation. For example, if the typical workstation at your installation has a 4 GB hard drive, then the average workstation storage capacity is 4 GB.

Utilization

An estimate of the fraction of each workstation disk space used, in the range 0 to 1. For example, if you expect that disks on workstations are 75% full, then use 0.75.

VersionExpansion

An expansion factor (greater than 1) that takes into account the additional backup versions, as defined in the copy group. A rough estimate allows 5% additional files for each backup copy. For example, for a version limit of 2, use 1.05, and for a version limit of 3, use 1.10.

NumWkst

The estimated total number of workstations that the server supports.

If clients use compression, the amount of space required may be less than the amount calculated, depending on whether the data is compressible.

Estimating space for archived files in random-access storage pools

The number of archived files generated by users is not necessarily related to the amount of data stored on their workstations. To estimate the total amount of space needed for all archived files in a single random-access (disk) storage pool, determine what percentage of user files are typically archived.

Work with policy administrators to calculate this percentage based on the number and type of archive copy groups defined. For example, if policy administrators have defined archive copy groups for only half of the policy domains in your enterprise, then estimate that you need less than 50% of the amount of space you have defined for backed-up files.

Because additional storage space can be added at any time, you can start with a modest amount of storage space and increase the space by adding storage volumes to the archive storage pool, as required.

Estimating space needs in sequential-access storage pools

Estimating the space needs in sequential-access storage pools is a relatively complex calculation based upon multiple considerations.

To estimate the amount of space required for sequential-access storage pools, consider:

- The amount of data being migrated from disk storage pools
- The length of time backed-up files are retained, as defined in backup copy groups
- The length of time archived files are retained, as defined in archive copy groups
- How frequently you reclaim unused space on sequential volumes

See “Reclaiming space in sequential-access storage pools” on page 319 for information about setting a reclamation threshold.

- Whether or not you use collocation to reduce the number of volume mounts required when restoring or retrieving large numbers of files from sequential volumes

If you use collocation, you may need additional tape drives and volumes.

See “Keeping client files together using collocation” on page 310 for information about using collocation for your storage pools.

- The type of storage devices and sequential volumes supported at your installation

Monitoring storage-pool and volume usage

Monitor your storage pools and volumes to determine space requirements, the status of data migration from one to storage pool to the next storage pool in the storage hierarchy, and the use of disk space by cached copies of files that have been migrated to the next storage pool.

Monitoring space available in a storage pool

Monitoring the space available in storage pools is important to ensure that client operations such as backup can complete successfully. To make more space available, you might need to define more volumes for disk storage pools, or add more volumes for sequential-access storage pools such as tape.

For more information about maintaining a supply of volumes in libraries, see:

“Managing the volume inventory” on page 143

Obtaining capacity estimates and utilization percentages of storage pools

Standard reports about storage pools list basic information, such as the estimated capacity and utilization percentage of all storage pools defined to the system.

To obtain a standard report, issue the following command:

```
query stgpool
```

Figure 45 on page 333 shows a standard report with all storage pools defined to the system. To monitor the use of storage pool space, review the *Estimated Capacity* and *Pct Util* columns.

Storage Pool Name	Device Class Name	Estimated Capacity	Pct Util	Pct Migr	High Mig Pct	Low Mig Pct	Next Storage Pool
ARCHIVEPOOL	DISK	0.0 M	0.0	0.0	90	70	
BACKTAPE	TAPE	180.0 M	85.0	100.0	90	70	
BACKUPPOOL	DISK	80.0 M	51.6	51.6	50	30	BACKTAPE
COPYPOOL	TAPE	300.0 M	42.0				
ENGBACK1	DISK	0.0 M	0.0	0.0	85	40	BACKTAPE

Figure 45. Information about storage pools

Estimated Capacity

Specifies the space available in the storage pool in megabytes (M) or gigabytes (G).

For a disk storage pool, this value reflects the total amount of available space in the storage pool, including any volumes that are varied offline.

For a sequential-access storage pool, this value is an estimate of the total amount of available space on all volumes in the storage pool. The total includes volumes with any access mode (read-write, unavailable, read-only, off-site, or destroyed). The total includes scratch volumes that the storage pool can acquire only when the storage pool is using at least one scratch volume for data.

Volumes in a sequential-access storage pool, unlike those in a disk storage pool, do not contain a precisely known amount of space. Data is written to a volume as necessary until the end of the volume is reached. For this reason, the estimated capacity is truly an *estimate* of the amount of available space in a sequential-access storage pool.

Pct Util

Specifies, as a percentage, the space used in each storage pool.

For disk storage pools, this value reflects the total number of disk blocks currently allocated by Tivoli Storage Manager. Space is allocated for backed-up, archived, or space-managed files that are eligible for server migration, cached files that are copies of server-migrated files, and files that reside on any volumes that are varied offline.

Note: The value for Pct Util can be higher than the value for Pct Migr if you query for storage pool information while a client transaction (such as a backup) is in progress. The value for Pct Util is determined by the amount of space actually allocated (while the transaction is in progress). The value for Pct Migr represents only the space occupied by *committed* files. At the end of the transaction, Pct Util and Pct Migr become synchronized.

For sequential-access storage pools, this value is the percentage of the total bytes of storage available that are currently being used to store active data (data that is not expired). Because the server can only estimate the available capacity of a sequential-access storage pool, this percentage also reflects an estimate of the actual utilization of the storage pool.

Figure 45 shows that the estimated capacity for a disk storage pool named BACKUPPOOL is 80 MB, which is the amount of available space on disk storage. More than half (51.6%) of the available space is occupied by either backup files or cached copies of backup files.

The estimated capacity for the tape storage pool named BACKTAPE is 180 MB, which is the total estimated space available on all tape volumes in the storage pool. This report shows that 85% of the estimated space is currently being used to store workstation files.

Note: This report also shows that volumes have not yet been defined to the ARCHIVEPOOL and ENGBACK1 storage pools, because the storage pools show an estimated capacity of 0.0 MB.

Obtaining statistics about space-trigger and scratch-volume utilization in storage pools

Detailed reports about a storage pools list not only estimated capacity and utilization percentage, but also space-trigger and scratch-volume utilization.

To obtain a detailed report, issue the following command:

```
query stgpool format=detailed
```

Space Trigger Utilization

Specifies the utilization of a storage pool, as calculated by the storage pool space trigger, if any, for the storage pool. You can define space triggers only for storage pools associated with DISK or FILE device types.

For sequential-access devices, space trigger utilization is expressed as a percentage of the number of used bytes on each sequential-access volume relative to the size of the volume, and the estimated capacity of all existing volumes in the storage pool. It does not include potential scratch volumes. Unlike the calculation for percent utilization (Pct Util), the calculation for space trigger utilization favors creation of new private file volumes by the space trigger over usage of additional scratch volumes.

For disk devices, space trigger utilization is expressed as a percentage of the estimated capacity, including cached data and deleted data that is waiting to be shredded. However, it excludes data that resides on any volumes that are varied offline. If you issue QUERY STGPOOL while a file creation is in progress, the value for space trigger utilization can be higher than the value for percent migration (Pct Migr). The value for space trigger utilization is determined by the amount of space actually allocated while the transaction is in progress. The value for percent migration represents only the space occupied by committed files. At the end of the transaction, these values are synchronized.

The value for space trigger utilization includes cached data on disk volumes. Therefore, when cache is enabled and migration occurs, the value remains the same because the migrated data remains on the volume as cached data. The value decreases only when the cached data expires or when the space that cached files occupy needs to be used for no-cached files.

Number of Scratch Volumes Used

Specifies the number of scratch volumes used in a sequential-access storage pool. You can use this value, along with the value of the field Maximum Scratch Volumes Allowed to determine the remaining number of scratch volumes that the server can request for a storage pool.

Monitoring the use of storage pool volumes

Monitoring how storage pool volumes are used lets you make the most efficient use available storage.

Task	Required Privilege Class
Display information about volumes	Any administrator

You can query the server for information about storage pool volumes:

- General information about a volume, for example:
 - Current access mode and status of the volume
 - Amount of available space on the volume
 - Location
- Contents of a storage pool volume (user files on the volume)
- The volumes that are used by a client node

Obtaining information about storage pool volumes

Standard reports provide a quick overview of basic information about storage pool volumes. More information is available in detailed reports.

To request general information about all volumes defined to the server, enter:

```
query volume
```

Figure 46 shows an example of the output of this standard query. The example illustrates that data is being stored on the 8 mm tape volume named WREN01, as well as on several other volumes in various storage pools.

Volume Name	Storage Pool Name	Device Class Name	Estimated Capacity	Pct Util	Volume Status
/dev/raixvo11	AIXPOOL1	DISK	240.0 M	26.3	On-Line
/dev/raixvo12	AIXPOOL2	DISK	240.0 M	36.9	On-Line
/dev/rdosvo11	DOSPOOL1	DISK	240.0 M	72.2	On-Line
/dev/rdosvo12	DOSPOOL2	DISK	240.0 M	74.1	On-Line
/dev/ros2vo11	OS2POOL1	DISK	240.0 M	55.7	On-Line
/dev/ros2vo12	OS2POOL2	DISK	240.0 M	51.0	On-Line
WREN00	TAPEPOOL	TAPE8MM	2.4 G	0.0	Filling
WREN01	TAPEPOOL	TAPE8MM	2.4 G	2.2	Filling

Figure 46. Information about storage pool volumes

To query the server for a detailed report on volume WREN01 in the storage pool named TAPEPOOL, enter:

```
query volume wren01 format=detailed
```

Figure 47 on page 336 shows the output of this detailed query. Table 38 on page 336 gives some suggestions on how you can use the information.

```

Volume Name: WREN01
Storage Pool Name: TAPEPOOL
Device Class Name: TAPE8MM
Estimated Capacity: 2.4 G
Pct Util: 26.3
Volume Status: Filling
Access: Read/Write
Pct. Reclaimable Space: 5.3
Scratch Volume?: No
In Error State?: No
Number of Writable Sides: 1
Number of Times Mounted: 4
Write Pass Number: 2
Approx. Date Last Written: 09/04/2002 11:33:26
Approx. Date Last Read: 09/03/2002 16:42:55
Date Became Pending:
Number of Write Errors: 0
Number of Read Errors: 0
Volume Location:
Last Update by (administrator): TANAGER
Last Update Date/Time: 09/04/2002 11:33:26

```

Figure 47. Detailed information for a storage pool volume

Table 38. Using the detailed report for a volume

Task	Fields and Description
Ensure the volume is available.	<p><i>Volume Status</i> <i>Access</i></p> <hr/> <p>Check the <i>Volume Status</i> to see if a disk volume has been varied offline, or if a sequential-access volume is currently being filled with data.</p> <p>Check the <i>Access</i> to determine whether files can be read from or written to this volume.</p>
Monitor the use of storage space.	<p><i>Estimated Capacity</i> <i>Pct Util</i></p> <hr/> <p>The <i>Estimated Capacity</i> is determined by the device class associated with the storage pool to which this volume belongs. Based on the estimated capacity, the system tracks the percentage of space occupied by client files (<i>Pct Util</i>).</p> <p>In this example, 26.3% of the estimated capacity is currently in use.</p>
Monitor the error status of the volume.	<p><i>Number of Write Errors</i> <i>Number of Read Errors</i></p> <hr/> <p>The server reports when the volume is in an error state and automatically updates the access mode of the volume to read-only. The <i>Number of Write Errors</i> and <i>Number of Read Errors</i> indicate the type and severity of the problem. Audit a volume when it is placed in error state. See "Auditing storage pool volumes" on page 782 for information about auditing a volume.</p>

Table 38. Using the detailed report for a volume (continued)

Task	Fields and Description
Monitor the life of sequential-access volumes that you have defined to the storage pool.	<p><i>Scratch Volume?</i> <i>Write Pass Number</i> <i>Number of Times Mounted</i> <i>Approx. Date Last Written</i> <i>Approx. Date Last Read</i></p> <hr/> <p>The server maintains usage statistics on volumes that are defined to storage pools. Statistics on a volume explicitly defined by an administrator remain for as long as the volume is defined to the storage pool. The server continues to maintain the statistics on defined volumes even as the volume is reclaimed and reused. However, the server deletes the statistics on the usage of a scratch volume when the volume returns to scratch status (after reclamation or after all files are deleted from the volume).</p> <p>In this example, WREN01 is a volume defined to the server by an administrator, not a scratch volume (<i>Scratch Volume?</i> is No).</p> <p>The <i>Write Pass Number</i> indicates the number of times the volume has been written to, starting from the beginning of the volume. A value of one indicates that a volume is being used for the first time.</p> <p>In this example, WREN01 has a write pass number of two, which indicates space on this volume may have been reclaimed or deleted once before.</p> <p>Compare this value to the specifications provided with the media that you are using. The manufacturer may recommend a maximum number of write passes for some types of tape media. You may need to retire your tape volumes after reaching the maximum passes to better ensure the integrity of your data. To retire a volume, move the data off the volume by using the MOVE DATA command. See “Moving data from one volume to another volume” on page 349.</p> <p>Use the <i>Number of Times Mounted</i>, the <i>Approx. Date Last Written</i>, and the <i>Approx. Date Last Read</i> to help you estimate the life of the volume. For example, if more than six months have passed since the last time this volume has been written to or read from, audit the volume to ensure that files can still be accessed. See “Auditing storage pool volumes” on page 782 for information about auditing a volume.</p> <p>The number given in the field, <i>Number of Times Mounted</i>, is a count of the number of times that the server has opened the volume for use. The number of times that the server has opened the volume is not always the same as the number of times that the volume has been physically mounted in a drive. After a volume is physically mounted, the server can open the same volume multiple times for different operations, for example for different client backup sessions.</p>
Determine the location of a volume in a sequential-access storage pool.	<p><i>Location</i></p> <hr/> <p>When you define or update a sequential-access volume, you can give location information for the volume. The detailed query displays this location name. The location information can be useful to help you track volumes (for example, off-site volumes in copy storage pools or active-data pools).</p>
Determine if a volume in a sequential-access storage pool is waiting for the reuse delay period to expire.	<p><i>Date Became Pending</i></p> <hr/> <p>A sequential-access volume is placed in the pending state after the last file is deleted or moved from the volume. All the files that the pending volume had contained were expired or deleted, or were moved from the volume. Volumes remain in the pending state for as long as specified with the REUSEDELAY parameter for the storage pool to which the volume belongs.</p>

Whether or not a volume is full, at times the Pct Util (percent of the volume utilized) plus the Pct Reclaimable Space (percent of the volume that can be reclaimed) may add up to more than 100 percent. This can happen when a volume contains aggregates that have empty space because of files in the aggregates that have expired or been deleted. The Pct Util field shows all space occupied by both non-aggregated files and aggregates, including empty space within aggregates. The Pct Reclaimable Space field includes any space that is reclaimable on the volume, also including empty space within aggregates. Because both fields include the empty space within aggregates, these values may add up to more than 100 percent. For more information about aggregates, see “How the server groups files before storing” on page 243 and “Obtaining information about the use of storage space” on page 345.

Obtaining information about the contents of a storage pool volume

Any administrator can request information about the contents of a storage pool volume. Viewing the contents of a storage volume is useful when a volume is damaged or before you request the server to correct inconsistencies in the volume, move files from one volume to another, or delete a volume from a storage pool.

Because the server tracks the contents of a storage volume through its database, the server does not need to access the requested volume to determine its contents.

To produce a report that shows the contents of a volume, issue the `QUERY CONTENT` command.

This report can be extremely large and may take a long time to produce. To reduce the size of this report, narrow your search by selecting one or all of the following search criteria:

Node name

Name of the node whose files you want to include in the query.

File space name

Names of file spaces to include in the query. File space names are case-sensitive and must be entered exactly as they are known to the server. Use the `QUERY FILESPACE` command to find the correct capitalization.

Number of files to be displayed

Enter a positive integer, such as 10, to list the first ten files stored on the volume. Enter a negative integer, such as -15, to list the last fifteen files stored on the volume.

Filetype

Specifies which types of files, that is, backup versions, archive copies, or space-managed files, or a combination of these. If the volume being queried is assigned to an active-data pool, the only valid values are `ANY` and `Backup`.

Format of how the information is displayed

Standard or detailed information for the specified volume.

Damaged

Specifies whether to restrict the query output either to files that are known to be damaged, or to files that are not known to be damaged.

Copied

Specifies whether to restrict the query output to either files that are backed

up to a copy storage pool, or to files that are not backed up to a copy storage pool. Whether files are stored in an active-data pool does not affect the output.

Note: There are several reasons why a file might have no usable copy in a copy storage pool:

The file was recently added to the volume and has not yet been backed up to a copy storage pool

The file should be copied the next time the storage pool is backed up.

The file is damaged

To determine whether the file is damaged, issue the QUERY CONTENT command, specifying the DAMAGED=YES parameter.

The volume that contains the files is damaged

To determine which volumes contain damaged files, issue the following command:

```
select * from contents where damaged=yes
```

The file is segmented across multiple volumes, and one or more of the other volumes is damaged

To determine whether the file is segmented, issue the QUERY CONTENT command, specifying the FORMAT=DETAILED parameter. If the file is segmented, issue the following command to determine whether any of the volumes containing the additional file segments are damaged:

```
select volume_name from contents where damaged=yes and  
file_name like '%filename%'
```

For more information about using the SELECT command, see the *Administrator's Reference*.

Example: Generating a standard report about the contents of a volume:

A standard report about the contents of a volume displays basic information such as the names of files.

To view the first seven backup files on volume WREN01 from file space /usr on client node TOMC, for example, enter:

```
query content wren01 node=tomc filespace=/usr count=7 type=backup
```

Figure 48 displays a standard report which shows the first seven files from file space /usr on TOMC stored in WREN01.

Node Name	Type	Filespace Name	Client's Name for File
TOMC	Bkup	/usr	/bin/ acctcom
TOMC	Bkup	/usr	/bin/ acledit
TOMC	Bkup	/usr	/bin/ aclput
TOMC	Bkup	/usr	/bin/ admin
TOMC	Bkup	/usr	/bin/ ar
TOMC	Bkup	/usr	/bin/ arcv
TOMC	Bkup	/usr	/bin/ banner

Figure 48. A standard report on the contents of a volume

The report lists logical files on the volume. If a file on the volume is an aggregate of logical files (backed-up or archived client files), all logical files that are part of the aggregate are included in the report. An aggregate can be stored on more than one volume, and therefore not all of the logical files in the report may actually be stored on the volume being queried.

Example: Generating a detailed report about the contents of a volume:

A detailed report about volume contents provides basic information as well as information about whether the file is stored across multiple volumes, whether the file is part of an aggregate, and whether the file is a cached copy of a file that has been migrated to the next storage pool in the hierarchy.

To display detailed information about the files stored on volume VOL1, enter:

```
query content vol1 format=detailed
```

Figure 49 on page 341 displays a detailed report that shows the files stored on VOL1. The report lists logical files and shows whether each file is part of an aggregate. If a logical file is stored as part of an aggregate, the information in the **Segment Number**, **Stored Size**, and **Cached Copy?** fields apply to the aggregate, not to the individual logical file.

If a logical file is part of an aggregate, the **Aggregated?** field shows the sequence number of the logical file within the aggregate. For example, the **Aggregated?** field contains the value 2/4 for the file AB0CTGLO.IDE, meaning that this file is the second of four files in the aggregate. All logical files that are part of an aggregate are included in the report. An aggregate can be stored on more than one volume, and therefore not all of the logical files in the report may actually be stored on the volume being queried.

For disk volumes, the **Cached Copy?** field identifies whether the file is a cached copy of a file that has been migrated to the next storage pool in the hierarchy.

```

Node Name: DWE
Type: Bkup
Filespace Name: OS2
Client's Name for File: \ README
Aggregated?: No
Stored Size: 27,089
Segment Number: 1/1
Cached Copy?: No

Node Name: DWE
Type: Bkup
Filespace Name: DRIVE_L_K:
Client's Name for File: \COMMON\DSMCOMM\ AB0CTCOM.ENT
Aggregated?: 1/4
Stored Size: 202,927
Segment Number: 1/1
Cached Copy?: No

Node Name: DWE
Type: Bkup
Filespace Name: DRIVE_L_K:
Client's Name for File: \COMMON\DSMCOMM\ AB0CTGLO.IDE
Aggregated?: 2/4
Stored Size: 202,927
Segment Number: 1/1
Cached Copy?: No

Node Name: DWE
Type: Bkup
Filespace Name: DRIVE_L_K:
Client's Name for File: \COMMON\DSMCOMM\ AB0CTTRD.IDE
Aggregated?: 3/4
Stored Size: 202,927
Segment Number: 1/1
Cached Copy?: No

Node Name: DWE
Type: Bkup
Filespace Name: DRIVE_L_K:
Client's Name for File: \COMMON\DSMCOMM\ AB0CTSYM.ENT
Aggregated?: 4/4
Stored Size: 202,927
Segment Number: 1/1
Cached Copy?: No

```

Figure 49. Viewing a detailed report of the contents of a volume

Identifying the volumes used by a client node

To identify the sequential volumes used by a client node, you can use the server's SELECT command.

The SELECT command queries the VOLUMEUSAGE table in the Tivoli Storage Manager database. For example, to get a list of volumes used by the EXCH1 client node in the TAPEPOOL storage pool, enter the following command:

```
select volume_name from volumeusage where node_name='EXCH1' and
stgpool_name='TAPEPOOL'
```

The results are something like the following:

```
VOLUME_NAME
-----
TAPE01
TAPE08
TAPE13
TAPE21
```

For more information about using the SELECT command, see the *Administrator's Reference*.

Monitoring migration processes

To obtain information about migration processing, you can request a standard storage-pool report.

Four fields on the standard storage-pool report provide you with information about the migration process. They include:

Pct Migr

Specifies the percentage of data in each storage pool that can be migrated. This value is used to determine when to start or stop migration.

For random-access and sequential-access disk storage pools, this value represents the amount of disk space occupied by backed-up, archived, or space-managed files that can be migrated to another storage pool. The calculation for random-access disk storage pools excludes cached data, but includes files on volumes that are varied offline.

For sequential-access tape and optical storage pools, this value is the percentage of the total volumes in the storage pool that actually contain data at the moment. For example, assume a storage pool has four explicitly defined volumes, and a maximum scratch value of six volumes. If only two volumes actually contain data at the moment, then Pct Migr is 20%.

This field is blank for copy storage pools and active-data pools.

High Mig Pct

Specifies when the server can begin migrating data from this storage pool. Migration can begin when the percentage of data that can be migrated reaches this threshold. (This field is blank for copy storage pools and active-data pools.)

Low Mig Pct

Specifies when the server can stop migrating data from this storage pool. Migration can end when the percentage of data that can be migrated falls below this threshold. (This field is blank for copy storage pools and active-data pools.)

Next Storage Pool

Specifies the primary storage pool destination to which data is migrated. (This field is blank for copy storage pools and active-data pools.)

Example: Monitoring data migration between storage pools

A storage pool is queried to determine high and low migration thresholds. The server is queried to monitor the migration process.

Figure 45 on page 333 shows that the migration thresholds for BACKUPPOOL storage pool are set to 50% for the *high migration threshold* and 30% for the *low migration threshold*.

When the amount of migratable data stored in the BACKUPPOOL storage pool reaches 50%, the server can begin to migrate files to BACKTAPE.

To monitor the migration of files from BACKUPPOOL to BACKTAPE, enter:

```
query stgpool back*
```

See Figure 50 on page 343 for an example of the results of this command.

If caching is on for a disk storage pool and files are migrated, the Pct Util value does not change because the cached files still occupy space in the disk storage

pool. However, the Pct Migr value decreases because the space occupied by cached files is no longer migratable.

Storage Pool Name	Device Class Name	Estimated Capacity	Pct Util	Pct Migr	High Mig Pct	Low Mig Pct	Next Storage Pool
BACKTAPE	TAPE	180.0 M	95.2	100.0	90	70	
BACKUPPOOL	DISK	80.0 M	51.6	28.8	50	30	BACKTAPE

Figure 50. Information on backup storage pools

You can query the server to monitor the migration process by entering:
query process

A message similar to Figure 51 is displayed:

Process Number	Process Description	Status
2	Migration	Disk Storage Pool BACKUPPOOL, Moved Files: 1086, Moved Bytes: 25555579, Unreadable Files: 0, Unreadable Bytes: 0

Figure 51. Information on the migration process

When migration is finished, the server displays the following message:

ANR1101I Migration ended for storage pool BACKUPPOOL.

Managing problems during migration processes

Migration processes can be suspended if a problem occurs. If migration is suspended, you can retry the process, cancel the process, end the migration process by changing the attributes of the storage pool from which data is being migrated, or provide additional space.

Canceling migration processes

To stop server migration when a problem occurs or when you need the resources the process is using, you can cancel the process.

First determine the identification number of the migration process by entering:
query process

A message similar to Figure 52 is displayed:

Process Number	Process Description	Status
1	Migration	ANR1113W Migration suspended for storage pool BACKUPPOOL - insufficient space in subordinate storage pool.

Figure 52. Getting the identification number of the migration process

Then you can cancel the migration process by entering:
cancel process 1

Stopping repeated attempts by the server to restart migration

Some errors cause the server to continue attempting to restart the migration process after 60 seconds. (If the problem still exists after several minutes, the migration process ends.) To stop the repeated attempts at restart, you can change some characteristics of the storage pool from which data is being migrated.

Depending on your environment, you can:

- Set higher migration thresholds for the storage pool from which data is being migrated. The higher threshold means the storage pool must have more migratable data before migration starts. This change delays migration.

In the example in “Example: Monitoring data migration between storage pools” on page 342, you could update the disk storage pool BACKUPPOOL.

- Add volumes to the pool from which data is being migrated. Adding volumes decreases the percentage of data that is migratable (Pct Migr).

In the example in “Example: Monitoring data migration between storage pools” on page 342, you could add volumes to the disk storage pool BACKUPPOOL to increase its storage capacity.

Tip: Do this only if you received an out-of-space message for the storage pool to which data is being migrated.

Providing additional space for the migration process

A migration process can be suspended because of insufficient space in the storage pool to which data is being migrated. To allow the migration process to complete, you can provide additional storage volumes for that storage pool.

In the example in “Example: Monitoring data migration between storage pools” on page 342, you can add volumes to the BACKTAPE storage pool or increase the maximum number of scratch tapes allowed for it. Either way, you increase the storage capacity of BACKTAPE.

Monitoring the use of cache space on disk storage

To determine whether cache is being used on disk storage and to monitor how much space is being used by cached copies, query the server for a detailed storage pool report.

The Pct Util value includes cached data on a volume (when cache is enabled) and the Pct Migr value excludes cached data. Therefore, when cache is enabled and migration occurs, the Pct Migr value decreases while the Pct Util value remains the same. The Pct Util value remains the same because the migrated data remains on the volume as cached data. In this case, the Pct Util value only decreases when the cached data expires.

If you update a storage pool from CACHE=YES to CACHE=NO, the cached files will not disappear immediately. The Pct Util value will be unchanged. The cache space will be reclaimed over time as the server needs the space, and no additional cached files will be created.

For example, to request a detailed report for BACKUPPOOL, enter:
`query stgpool backuppool format=detailed`

Figure 53 on page 345 displays a detailed report for the storage pool.

```

Storage Pool Name: BACKUPPOOL
Storage Pool Type: PRIMARY
Device Class Name: DISK
Estimated Capacity: 80.0 M
Space Trigger Util: 0.0
    Pct Util: 42.0
    Pct Migr: 29.6
    Pct Logical: 82.1
    High Mig Pct: 50
    Low Mig Pct: 30
    Migration Delay: 0
Migration Continue: Yes
Migration Processes: 1
Reclamation Processes:
    Next Storage Pool: BACKTAPE
Reclaim Storage Pool:
Maximum Size Threshold: No Limit
    Access: Read/Write
Description:
Overflow Location:
Cache Migrated Files?: Yes
    Collocate?:
Reclamation Threshold:
Offsite Reclamation Limit:
Maximum Scratch Volumes Allowed:
Number of Scratch Volumes Used:
Delay Period for Volume Reuse: 0 Day(s)
    Migration in Progress?: Yes
    Amount Migrated (MB): 0.10
Elapsed Migration Time (seconds): 5
Reclamation in Progress?:
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 09/04/2002 16:47:49
Storage Pool Data Format: Native
Copy Storage Pool(s):
Active-data Pool(s):
Continue Copy on Error?:
    CRC Data: No
Reclamation Type:
Overwrite Data when Deleted: 2 Time(s)

```

Figure 53. Detailed storage pool report

When **Cache Migrated Files?** is set to **Yes**, the value for Pct Util should not change because of migration, because cached copies of files migrated to the next storage pool remain in disk storage.

This example shows that utilization remains at 42%, even after files have been migrated to the BACKTAPE storage pool, and the current amount of data eligible for migration is 29.6%.

When **Cache Migrated Files?** is set to **No**, the value for Pct Util more closely matches the value for Pct Migr because cached copies are not retained in disk storage.

Obtaining information about the use of storage space

You can generate reports to determine the amount of space used by client nodes and file spaces, storage pools and device classes, or types of data (backup, archive, or space-managed). Generating occupancy reports on a regular basis can help you with capacity planning.

Task	Required Privilege Class
Query the server for information about server storage	Any administrator

To obtain reports with information broken out by node or file space, issue the QUERY OCCUPANCY command.

Each report gives two measures of the space in use by a storage pool:

- Logical space occupied

The amount of space used for logical files. A logical file is a client file. A logical file is stored either as a single physical file, or in an aggregate with other logical files. The logical space occupied in active-data pools includes the space occupied by inactive logical files. Inactive logical files in active-data pools are removed by reclamation.

- Physical space occupied

The amount of space used for physical files. A physical file is either a single logical file, or an aggregate composed of logical files.

An aggregate might contain empty space that was used by logical files that are now expired or deleted, or that were deactivated in active-data pools. Therefore, the amount of space used by physical files is equal to or greater than the space used by logical files. The difference gives you a measure of how much unused space any aggregates may have. The unused space can be reclaimed in sequential storage pools.

You can also use this report to evaluate the average size of workstation files stored in server storage.

Obtaining information about space used by client nodes

You can request information about how much data a client has backed up, archived, or migrated to server storage. You can also request information about the amount of storage space used by each client node and file space, as well as the number of files that are in server storage that were backed up to a copy storage pool or an active-data pool.

To determine the amount of server storage space used by the /home file space belonging to the client node MIKE, for example, enter:

```
query occupancy mike /home
```

File space names are case-sensitive and must be entered exactly as they are known to the server. To determine the correct capitalization, issue the QUERY FILESPACE command. For more information, see “Managing file spaces” on page 397.

Figure 54 shows the results of the query. The report shows the number of files backed up, archived, or migrated from the /home file space belonging to MIKE. The report also shows how much space is occupied in each storage pool.

If you back up the ENGBACK1 storage pool to a copy storage pool, the copy storage pool would also be listed in the report. To determine how many of the client node's files in the primary storage pool have been backed up to a copy storage pool, compare the number of files in each pool type for the client node.

Node Name	Type	Filespace Name	Storage Pool Name	Number of Files	Physical Space Occupied (MB)	Logical Space Occupied (MB)
MIKE	Bkup	/home	ENGBACK1	513	3.52	3.01

Figure 54. A report of the occupancy of storage pools by client node

You can also use the QUERY NODEDATA command to display information about the data for one or more nodes in a sequential-access storage pool. (The command is not supported for random-access storage pools.) The output of the QUERY NODEDATA command displays the name of the volume on which a node's data is written, the name of the storage pool in which the volume is located, and the amount of space occupied by the data on the volume. For example, to display information about the data for nodes whose names begin with the letter "e," you would enter the following command using a wildcard character:

```
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

For details about the QUERY NODEDATA command, refer to the *Administrator's Reference*.

Obtaining information about space utilization of storage pools

You can monitor the amount of space being used by an individual storage pool or a group of storage pools.

To query the server for the amount of data stored in backup tape storage pools belonging to the TAPECLASS device class, for example, enter:

```
query occupancy devclass=tapeclass
```

Figure 55 displays a report on the occupancy of tape storage pools assigned to the TAPECLASS device class.

Node Name	Type	Filespace Name	Storage Pool Name	Number of Files	Physical Space Occupied (MB)	Logical Space Occupied (MB)
CAROL	Arch	OS2C	ARCHTAPE	5	.92	.89
CAROL	Bkup	OS2C	BACKTAPE	21	1.02	1.02
PEASE	Arch	/home/pease/dir	ARCHTAPE	492	18.40	18.40
PEASE	Bkup	/home/pease/dir	BACKTAPE	33	7.60	7.38
PEASE	Bkup	/home/pease/dir1	BACKTAPE	2	.80	.80
TOMC	Arch	/home/tomc/driver5	ARCHTAPE	573	20.85	19.27
TOMC	Bkup	/home	BACKTAPE	13	2.02	1.88

Figure 55. A report on the occupancy of storage pools by device class

Tip: For archived data, you might see "(archive)" in the Filespace Name column instead of a file space name. This means that the data was archived before collocation by file space was supported by the server.

Requesting information about space used by backed-up, archived, and space-managed files

You can query the server for the amount of space used by backed-up, archived, and space-managed files. By determining the average size of workstation files stored in server storage, you can estimate how much storage capacity you might need when registering new client nodes to the server.

For example, to request a report about backup versions stored in the disk storage pool named BACKUPPOOL, enter:

```
query occupancy stgpool=backuppoo1 type=backup
```

Figure 56 displays a report on the amount of server storage used for backed-up files.

Node Name	Type	Filespace Name	Storage Pool Name	Number of Files	Physical Space Occupied (MB)	Logical Space Occupied (MB)
CAROL	Bkup	OS2C	BACKUPPOOL	513	23.52	23.52
CAROL	Bkup	OS2D	BACKUPPOOL	573	20.85	20.85
PEASE	Bkup	/marketing	BACKUPPOOL	132	12.90	9.01
PEASE	Bkup	/business	BACKUPPOOL	365	13.68	6.18
TOMC	Bkup	/	BACKUPPOOL	177	21.27	21.27

Figure 56. A report of the occupancy of backed-up files in storage pools

To determine the average size of backup versions stored in BACKUPPOOL, complete the following steps using the data provided in Figure 56:

1. Add the number of megabytes of space occupied by backup versions. In this example, backup versions occupy 92.22 MB of space in BACKUPPOOL.
2. Add the number of files stored in the storage pool. In this example, 1760 backup versions reside in BACKUPPOOL.
3. Divide the space occupied by the number of files to determine the average size of each file backed up to the BACKUPPOOL. In this example, the average size of each workstation file backed up to BACKUPPOOL is about 0.05 MB, or approximately 50 KB.

You can use this average to estimate the capacity required for additional storage pools that are defined to the server.

For information about planning storage space, see “Estimating space needs for storage pools” on page 330 and “Estimating space for archived files in random-access storage pools” on page 331.

Obtaining information about free disk space in FILE device classes

You can monitor the amount of free disk space in directories associated with FILE device classes. The Tivoli Storage Manager server uses directories as the location for files that represent storage-pool volumes.

To request information about the amount of free disk space in each directory for all device classes with a device type of FILE, issue QUERY DIRSPACE command.

Figure 57 on page 349 displays the output for this command.

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

Figure 57. A report of the free disk space for all device classes of device type FILE

To obtain the amount of free space associated with a particular device class, issue the following command:

```
query dirspace device_class_name
```

Moving data from one volume to another volume

You might need to move data in some situations, for example, when you need to salvage readable data from a damaged volume. To move data (files) from one volume to another volume in the same or a different storage pool, use the MOVE DATA command. The volumes can be on-site volumes or off-site volumes.

Task	Required Privilege Class
Move files from a volume in any storage pool to an available volume in any storage pool	System or unrestricted storage
Move files from one volume to an available volume in any storage pool to which you are authorized	Restricted storage

During the data movement process, the server:

- Moves any readable files to available volumes in the specified destination storage pool
- Deletes any cached copies from a disk volume
- Attempts to bypass any files that previously were marked as damaged

During the data movement process, users cannot access the volume to restore or retrieve files, and no new files can be written to the volume.

Remember:

- Files in a copy storage pool or an active-data pool do not move when primary files are moved.
- In addition to moving data from volumes in storage pools that have NATIVE or NONBLOCK data formats, you can also 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.

Data movement within the same storage pool

Moving files from one volume to other volumes in the same storage pool provides a number of benefits.

Moving files from one volume to other volumes in the same storage pool is useful:

- When you want to free up all space on a volume so that it can be deleted from the Tivoli Storage Manager server

See “Deleting storage pool volumes” on page 361 for information about deleting backed-up, archived, or space-managed data before you delete a volume from a storage pool.

- When you need to salvage readable files from a volume that has been damaged
- When you want to delete cached files from disk volumes

If you want to force the removal of cached files, you can delete them by moving data from one volume to another volume. During the move process, the server deletes cached files remaining on disk volumes.

If you move data between volumes within the same storage pool and you run out of space in the storage pool before all data is moved from the target volume, then you cannot move all the data from the target volume. In this case, consider moving data to available space in another storage pool as described in “Data movement to a different storage pool.”

Data movement to a different storage pool

You can move all data from a volume in one storage pool to volumes in another storage pool. When you specify a target storage pool that is different than the source storage pool, the server uses the storage hierarchy to move data if more space is required.

Remember: Data cannot be moved from a primary storage pool to a copy storage pool or to an active-data pool. Data in a copy storage pool or an active-data pool cannot be moved to another storage pool.

You can move data from random-access storage pools to sequential-access storage pools. For example, if you have a damaged disk volume and you have a limited amount of disk storage space, you could move all files from the disk volume to a tape storage pool. Moving files from a disk volume to a sequential storage pool may require many volume mount operations if the target storage pool is collocated. Ensure that you have sufficient personnel and media to move files from disk to sequential storage.

When a data move from a shred pool is complete, the original data is shredded. However, if the destination is not another shred pool, you must set the SHREDTONOSHRED parameter to YES to force the movement to occur. If this value is not specified, the server issues an error message and does not allow the data to be moved. See “Securing sensitive client data” on page 499 for more information about shredding.

Data movement from off-site volumes in copy storage pools or active-data pools

You can move data from off-site volumes without bringing the volumes on-site.

Processing of the MOVE DATA command for volumes in copy -storage pools and active-data pools is similar to that of primary-storage pools, with the following exceptions:

- Volumes in copy-storage pools and active-data pools might be set to an access mode of *offsite*, making them ineligible to be mounted. During processing of the MOVE DATA command, valid files on off-site volumes are copied from the original files in the primary-storage pools. In this way, valid files on off-site volumes are copied without having to mount these volumes. These new copies of the files are written to another volume in the copy-storage pool or active-data pool.
- With the MOVE DATA command, you can move data from any primary-storage pool volume to any primary-storage pool. However, you can move data from a copy-storage pool volume *only* to another volume within the same-copy storage pool. Similarly, you can move data from an active-data pool volume *only* to another volume within the same active-data pool.

When you move files from a volume marked as off-site, the server performs the following actions:

1. Determines which files are still active on the volume from which you are moving data
2. Obtains these active files from a primary-storage pool or from another copy-storage pool or active-data pool
3. Copies the files to one or more volumes in the destination copy-storage pool or active-data pool

Processing of the MOVE DATA command for primary-storage pool volumes does not affect copy-storage pool or active-data pool files.

Moving data

You can move data using the MOVE DATA command. Before moving data, however, take steps to ensure that the move operation succeeds.

Before beginning this procedure:

- If you want to ensure that no new files are written to a volume after you move data from it, change the volume's access mode to read-only. This prevents the server from filling the volume with data again as soon as data is moved. You might want to do this if you want to delete the volume.

See "Updating storage pool volumes" on page 238 for information about updating the access mode of a storage pool volume.

- Ensure sufficient space is available on volumes within the specified destination storage pool by:
 1. Querying the source storage volume to determine how much space is required on other volumes. See "Monitoring the use of storage pool volumes" on page 335 for information about requesting information about a storage volume.
 2. Querying the specified destination storage pool to ensure there is sufficient capacity to store the files being moved. See "Monitoring space available in a storage pool" on page 332 for information about querying a storage pool.

If you need more storage space, define volumes or increase the maximum number of scratch volumes in the specified destination storage pool. See “Defining storage pool volumes” on page 237 for preparing volumes to be used for server storage.

- If you are moving files from a volume in a sequential storage pool to another volume in the same storage pool, ensure that the mount limit of the device class associated with the storage pool is greater than one.

For information about the mount limit value for the device class, see:

“Obtaining information about device classes” on page 216

- If you are moving files from a tape volume to a tape storage pool, ensure that the two tape drives required are available.

To move data, issue the MOVE DATA command.

For example, to move the files stored in the /dev/vol3 volume to any available volume in the STGTMP1 storage pool, enter:

```
move data /dev/vol3 stgpool=stgtmp1
```

When you move data from a volume, the server starts a background process and sends informational messages, such as:

```
ANR1140I Move Data process started for volume /dev/vol3  
(process ID 32).
```

To run the MOVE DATA command in the foreground on an administrative client, issue the command with the WAIT=YES parameter.

Remember:

- A volume might not be totally empty after a move data operation completes. For example, the server may be unable to relocate one or more files to another volume because of input/output errors on the device or because errors were found in the file. You can delete the volume with DISCARDATA=YES to delete the volume and any remaining files. The server then deletes the remaining files that had I/O or other errors.
- In addition to moving data from volumes in storage pools that have NATIVE or NONBLOCK data formats, you can also 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.

Requesting information about the data-movement process

You can query the server for statistics about the data-movement process.

To request information, issue the QUERY PROCESS command.

Figure 58 on page 353 shows an example of the report that you receive about the data movement process.

Process Number	Process Description	Status
32	Move Data	Volume /dev/vol3, (storage pool BACKUPPOOL), Target Pool STGTMP1, Moved Files: 49, Moved Bytes: 9,121,792, Unreadable Files: 0, Unreadable Bytes: 0. Current File (bytes): 3,522,560 Current output volume: VOL1.

Figure 58. Information about the file movement process

Reclaiming space in aggregates by moving data

Empty space accumulates in a file aggregate as logical files in that aggregate are deleted. During reclamation processing, the aggregate is reconstructed by removing the empty space left by the deleted files. However, you cannot start reclamation processing for specific volumes.

To remove empty space in a specific volume and reconstruct an aggregate, issue the MOVE DATA command. By default, this command removes the empty space occupied by deleted files in an aggregate.

Remember:

1. Reclaiming empty space in NDMP-generated images is not an issue because NDMP-generated images are not aggregated.
2. Reconstruction removes inactive backup files in active-data pools. Specifying RECONSTRUCT=NO when moving data from volumes in an active-data pool prevents the inactive backup files from being removed.

Monitoring the movement of data between volumes

You can query the server for volume information to monitor the movement of data between volumes.

To request information, use the QUERY VOLUME command.

For example, to see how much data has moved from the source volume in the move operation example, enter:

```
query volume /dev/vol3 stgpool=backuppool
```

Near the beginning of the move process, querying the volume from which data is being moved gives the following results:

Volume Name	Storage Pool Name	Device Class Name	Estimated Capacity	Pct Util	Volume Status
/dev/vol3	BACKUPPOOL	DISK	15.0 M	59.9	On-Line

Querying the volume to which data is being moved (VOL1, according to the process query output) gives the following results:

Volume Name	Storage Pool Name	Device Class Name	Estimated Capacity	Pct Util	Volume Status
VOL1	STGTMP1	8500DEV	4.9 G	0.3	Filling

At the end of the move process, querying the volume from which data was moved gives the following results:

Volume Name	Storage Pool Name	Device Class Name	Estimated Capacity	Pct Util	Volume Status
/dev/vol3	BACKUPPOOL	DISK	15.0 M	0.0	On-Line

Moving data belonging to a client node

You can move data located in a sequential-access storage pool for one or more nodes, or for a single node with selected file spaces, using the MOVE NODEDATA command. The data can be located in either a primary storage pool, copy storage pool, or active-data pool.

When the source storage pool is a primary storage pool, you can move data to other volumes within the same pool or to another primary storage pool. When the source storage pool is a copy storage pool, data can only be moved to other volumes within that storage pool. When the source storage pool is an active-data pool, data can only be moved to other volumes within that same storage pool.

Tips:

- In addition to moving data from volumes in storage pools that have NATIVE or NONBLOCK data formats, you can also 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 files within the same storage pool, there must be volumes available that do not contain the data you are moving. That is, the server cannot use a destination volume containing data that will need to be moved.
- When moving data from volumes in an active-data pool, you have the option of reconstructing file aggregates during data movement. Reconstruction removes inactive backup files in the pool. Specifying no reconstruction prevents the inactive files from being removed.

Task	Required Privilege Class
Move data by node	System, unrestricted storage or restricted storage

Moving data in all file spaces belonging to one or more nodes

Moving data for all file spaces on one or more nodes provides a number of benefits.

Moving data is useful:

- When you want to optimize performance by reducing the number of volume mounts required during a restore operation by consolidating data for a specific node or nodes within a storage pool
- When you want to move data for specified nodes into a different storage pool
- When you want to increase performance of client restore processing by first moving data to a random-access storage pool

Best practice: Avoid movement of data into, out of, or within a storage pool while MOVE NODEDATA is concurrently processing data on the same storage pool.

To move all file spaces for a single node named ACCOUNTING where the data is in storage pool ACCTPOOL and the destination storage pool is BACKUPPOOL enter:

```
move nodedata accounting fromstgpool=acctpool tostgpool=backuppool
```

Moving data in selected file spaces belonging to a single node

Moving data for selected file spaces for one node provides a number of benefits.

Moving data is useful:

- When you want to optimize performance by reducing the number of volume mounts required during a restore operation by consolidating data for specific file spaces within a storage pool.
- When you want to consolidate data for critical file spaces allowing restore of these files to be given higher priority during recovery situations. This would be advantageous during data recovery when it is essential to first restore only business-critical data and then restore non-business-critical data.
- When you want to move specific file spaces into a different storage pool.
- When you want to increase performance of client restore processing by first moving data to a random-access storage pool.

For example, consider moving data for a single node and restricting the data movement to files in a specific non-Unicode file space (for this example, `\\eng\e$`) as well as a specific Unicode file space (for this example, `\\eng\d$`). The node name owning the data is ENGINEERING and it currently has data stored in the ENGPPOOL storage pool. After the move is complete, the data is located in the destination storage pool BACKUPPOOL. To move the data enter the following:

```
move nodedata engineering fromstgpool=engpool  
tostgpool=backuppool filespace=\\eng\e$ unifiespace=\\eng\d$
```

Another example is to move data for a single node named MARKETING from all primary sequential-access storage pools to a random-access storage pool named DISKPOOL. First obtain a list of storage pools that contain data for node MARKETING, issue either:

```
query occupancy marketing
```

or

```
SELECT * from OCCUPANCY where node_name='MARKETING';
```

For this example the list of resulting storage pool names all begin with the characters FALLPLAN. To move the data repeat the following command for every instance of FALLPLAN. The following example displays the command for FALLPLAN3:

```
move nodedata marketing fromstgpool=fallplan3  
tostgpool=diskpool
```

A final example shows moving both non-Unicode and Unicode file spaces for a node. For node NOAH move non-Unicode file space `\\servtuc\d$` and Unicode

file space \\tsmserv1\e\$ that has a file space ID of 2 from sequential-access storage pool TAPEPOOL to random-access storage pool DISKPOOL.

```
move nodedata noah fromstgpool=tapepool tostgpool=diskpool
  filespace=\\servtuc\d$ fsid=2
```

Obtaining information about data-movement processes

You can query the server for statistics about the data movement for a client node.

To request information on the data movement process, enter:

```
query process
```

Figure 59 shows an example of the report that you receive about the data movement process.

Process Number	Process Description	Status
3	Move Node Data	Storage Pool 3590FC, Target Pool 3590FC Files Moved: 0, Bytes Moved: 0, Unreadable Files: 0, Unreadable Bytes: 0. Current Physical File (bytes): 268,468,584 Current input volume: DST308. Current output volume: DST279.

Figure 59. Information on the data movement process

Troubleshooting incomplete data-movement operations

There are several reasons why an operation to move node data might not complete.

The most common reasons are:

- Files have been marked as damaged in the source storage pool. For more information about how to work with files that are marked as damaged, see “Fixing damaged files” on page 790.
- Files in the source storage pool reside on volumes whose access mode is off-site, destroyed or unavailable. To complete the move operation, bring the volumes on-site, restore destroyed volumes from a copy storage pool or an active-data pool, or make the volumes available.
- Files were moved, added or deleted during the move operation. To prevent this situation, avoid the following operations during move processing:
 - Migration of any type relating to the storage pool
 - Reclamation of volumes within the storage pool
 - Simultaneously running MOVE DATA processing for a volume in a storage pool that contains data to be moved during MOVE NODEDATA processing
 - Backup operations into a copy storage pool or an active-data pool while a MOVE NODEDATA is running for that pool
 - Storage of files from a client directly into the storage pool

Renaming storage pools

When distributing policy using enterprise configuration, you might need to rename a storage pool. Renaming a storage pool can require changing the destination storage pool in copy groups and management classes.

To rename a storage pool, issue the UPDATE STGPOOL command.

When you rename a storage pool, any administrators with restricted storage privilege for the storage pool automatically have restricted storage privilege to the storage pool under the new name. If the renamed storage pool is in a storage pool hierarchy, the hierarchy is preserved.

Copy groups and management classes might contain a storage pool name as a destination. If you rename a storage pool used as a destination, the destination in a copy group or management class is not changed to the new name of the storage pool. To continue to use the policy with the renamed storage pool as a destination, you must change the destination in the copy groups and management classes. You then activate the policy set with the changed destinations.

For information about setting up a managed server in an enterprise configuration, see “Setting up a managed server” on page 688.

Defining copy storage pools and active-data pools

Use a copy storage pool or an active-data pools to back up one or more primary storage pools. When defining copy storage pools or active-data pools, you can take advantage of various Tivoli Storage Manager functions by specifying certain properties.

To define a copy storage pool, issue the DEFINE STGPOOL command and specify POOLTYPE=COPY. To define an active-data pool, issue the DEFINE STGPOOL command and specify POOLTYPE=ACTIVEDATA. When you define a copy storage pool or an active-data pool, be prepared to provide some or all of the information in Table 39.

Remember:

1. To back up a primary storage pool to an active-data pool, the data format must be NATIVE or NONBLOCK. You can back up a primary storage pool to a copy storage pool using NATIVE, NONBLOCK, or any of the NDMP formats. The target storage pool must have the same data format as the source storage pool.

Table 39. Information for defining copy storage pools and active-data pools

Information	Explanation
Device class	Specifies the name of the device class assigned for the storage pool. This is a required parameter.
Pool type	Specifies that you want to define a copy storage pool or an active-data pool. This is a required parameter. You cannot change the pool type when updating a storage pool.

Table 39. Information for defining copy storage pools and active-data pools (continued)

Information	Explanation
Access mode	<p>Defines access to volumes in the storage pool for user operations (such as backup and restore) and system operations (such as reclamation). Possible values are:</p> <p>Read/Write User and system operations can read from or write to the volumes.</p> <p>Read-Only User operations can read from the volumes, but not write. However, system processes can move files within the volumes in the storage pool.</p> <p>Unavailable Specifies that users cannot access files stored on volumes in the copy storage pool or an active-data pool. Files can be moved from volume to volume with the same copy storage pool or from volume to volume within the same active-data pool, but no new writes are permitted to the volumes in the storage pool from volumes outside the storage pool.</p>
Maximum number of scratch volumes	<p>When you specify a value greater than zero, the server dynamically acquires scratch volumes when needed, up to this maximum number. This is a required parameter.</p> <p>For automated libraries, set this value equal to the physical capacity of the library. For details, see: “Maintaining a supply of scratch volumes in an automated library” on page 150</p>
Collocation	<p>When collocation is enabled, the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of sequential-access storage volumes. See “Collocation of copy storage pools and active-data pools” on page 316.</p>
Reclamation threshold	<p>Specifies when to initiate reclamation of volumes in the copy storage pool or active-data pool. Reclamation is a process that moves any remaining files from one volume to another volume, thus making the original volume available for reuse. A volume is eligible for reclamation when the percentage of unused space on the volume is greater than the reclaim parameter value.</p> <p>Reclamation processing works differently for off-site copy storage pool volumes, active-data pool volumes, and virtual volumes. When a copy storage pool volume or an active-data pool volume that is off-site becomes eligible for reclamation, the reclamation process tries to retrieve the files on the reclaimable volume from a primary or copy storage pool volume that is on-site. The process then writes these files to an available volume in the original copy storage pool or active-data pool. See “Reclaiming copy storage pools and active-data pools” on page 325 and “Controlling reclamation of virtual volumes” on page 324 for more details.</p>
Reuse delay period	<p>Specifies the number of days that must elapse after all of the files have been deleted from a volume before the volume can be rewritten or returned to the scratch pool. See “Delayed reuse of reclaimed volumes” on page 329.</p>
Off-site reclaim limit	<p>Specifies the number of off-site volumes to be reclaimed during reclamation for a storage pool. See “Reclamation of off-site volumes” on page 326.</p>

Table 39. Information for defining copy storage pools and active-data pools (continued)

Information	Explanation
Reclamation processes	Specifies the number of concurrent processes to use for reclaiming the volumes in a storage pool. See “Reclaiming copy storage pools and active-data pools” on page 325.

For more information, see “Backing up storage pools” on page 760.

Example: Defining a copy storage pool

Copies of the files are stored in the Tivoli Storage Manager default disk storage pools for disaster recovery purposes. You create a copy storage pool and decide to use only scratch tapes in the new pool.

Assume you need to maintain copies of the files stored in BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL (default disk storage pools) for disaster recovery purposes. You want to create a copy storage pool named DISASTER-RECOVERY. You decide to use only scratch tapes in the new pool, setting the maximum number of scratch volumes to an appropriate value. You enter the following command:

```
define stgpool disaster-recovery tapeclass pooltype=copy
maxscratch=100
```

To store data in the new storage pool, you must back up the primary storage pools (BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL) to the DISASTER-RECOVERY pool. See “Backing up storage pools” on page 760.

Properties of primary, copy, and active-data pools

Primary-storage pools, copy-storage pools, and active-data pools have different properties. Understanding these differences helps you make the most efficient use of storage space.

Table 40 compares the characteristics of primary, copy-storage, and active-data pools.

Table 40. Comparing primary-storage pools, copy-storage pools, and active-data pools

Characteristic	Primary-storage pool	Copy-storage pool	Active-data pool
Destination for backed-up or archived files (specified in backup or archive copy groups)	Yes	No	No
Destination for space-managed files (specified in the management class)	Yes	No	No
Off-site access mode for volumes	No	Yes, except for volumes with device type SERVER	Yes, except for volumes with device type SERVER
Destroyed access mode for volumes	Yes	No	No
Random-access storage volumes	Yes	No	No
Sequential-access storage volumes	Yes	Yes	Yes

Table 40. Comparing primary-storage pools, copy-storage pools, and active-data pools (continued)

Characteristic	Primary-storage pool	Copy-storage pool	Active-data pool
Contents	Client files (backup versions, archived files, space-managed files)	Copies of files that are stored in primary storage pools	Active-only versions of client backup files that are stored in primary storage pools. Archive data and space-managed files are not permitted.
Moving data allowed	Within the same primary-storage pool, or to any primary-storage pool	<p>Within the same storage pool only.</p> <ul style="list-style-type: none"> • If moving data by volume and volumes are off-site, data is copied from the original files in primary-storage pools. • If volumes are off-site, you cannot move data in those volumes by node. 	<p>Within the same storage pool only.</p> <ul style="list-style-type: none"> • If moving data by volume and volumes are off-site, data is copied from the original files in primary-storage pools. • If volumes are off-site, you cannot move data in those volumes by node.
Collocation	Yes (sequential-access storage pools only)	Yes	Yes
Reclamation	Yes (sequential-access storage pools only)	<p>Yes</p> <p>Virtual volumes (volumes with device type SERVER) and off-site volumes are handled differently. For details, see “Controlling reclamation of virtual volumes” on page 324 and “Reclamation of off-site volumes” on page 326.</p>	<p>Yes</p> <p>Virtual volumes (volumes with device type SERVER) and off-site volumes are handled differently. For details, see “Controlling reclamation of virtual volumes” on page 324 and “Reclamation of off-site volumes” on page 326.</p>
File deletion	<p>Files are deleted:</p> <ul style="list-style-type: none"> • During inventory expiration processing, if the files have expired • When a file space is deleted • When a volume is deleted with the option to discard the data • When a primary-storage pool volume is audited with the FIX=YES option, if the files on the volume are damaged and no other copies of the file exist 	<p>Files are deleted:</p> <ul style="list-style-type: none"> • Whenever the primary-copy of the file is deleted from the primary-storage pool (because of expiration, file space deletion, or volume deletion) • When a volume is deleted with the option to discard the data • When a copy-storage pool volume is audited with the FIX=YES option, if the files on the volume are damaged 	<p>Files are deleted:</p> <ul style="list-style-type: none"> • During reclamation when inactive backup files are removed • Whenever the primary copy of the file is deleted from the primary-storage pool (because of expiration, file space deletion, or volume deletion) • When a volume is deleted with the option to discard the data • When an active-data pool volume is audited with the FIX=YES option, if the files on the volume are damaged

Deleting storage pools

Before deleting a storage pool, you need to take certain precautions to prevent data loss and to ensure uninterrupted storage operations.

Task	Required Privilege Class
Delete storage pools	System

Before you delete a storage pool, ensure that:

- All volumes within the storage pool have been deleted
Ensure that you have saved any readable data that you want to preserve by issuing the MOVE DATA command. Moving all of the data that you want to preserve may require you to issue the MOVE DATA command several times.
Before you begin deleting all volumes that belong to the storage pool, change the access mode of the storage pool to unavailable so that no files can be written to or read from volumes in the storage pool.
See “Deleting storage pool volumes that contain data” on page 362 for information about deleting volumes.
- The storage pool is not identified as the next storage pool within the storage hierarchy
To determine whether this storage pool is referenced as the next storage pool within the storage hierarchy, query for storage pool information as described in “Monitoring space available in a storage pool” on page 332.
Update any storage pool definitions to remove this storage pool from the storage hierarchy by performing one of the following:
 - Naming another storage pool as the next storage pool in the storage hierarchy
 - Entering the value for the NEXTSTGPOOL parameter as "" (double quotation marks) to remove this storage pool from the storage hierarchy definitionSee “Defining storage pools” on page 227 for information about defining and updating storage pools.
- The storage pool to be deleted is not specified as the destination for any copy group in any management class within the active policy set of any domain. Also, a storage pool to be deleted cannot be the destination for space-managed files (specified in any management class within the active policy set of any domain).
If this pool is a destination and the pool is deleted, operations fail because there is no storage space to store the data.

To delete a storage pool, issue the DELETE STGPOOL command.

Deleting storage pool volumes

You can delete empty storage pool volumes or volumes that contain data from primary storage pools, copy storage pools, or active-data pools. You can also delete the client files that those volumes contain.

If files that are not cached are deleted from a primary storage pool volume, any copies of these files in copy storage pools and active-data pools will also be deleted.

Files in a copy storage pool or an active-data pool are never deleted unless:

- Data retention is off, or the files have met their retention criterion.

- The volume that contains the copy file is deleted by using the DISCARDDATA=YES option.
- A read error is detected by using AUDIT VOLUME with the FIX=YES option for a copy storage pool volume or an active-data pool volume.
- The primary file is deleted because of:
 - Policy-based file expiration
 - File space deletion
 - Deletion of the primary storage pool volume

Tip: If you are deleting many volumes, delete the volumes one at a time. Concurrently deleting many volumes can adversely affect server performance.

Task	Required Privilege Class
Delete volumes from any storage pool	System or unrestricted storage
Delete volumes from storage pools over which they have authority	Restricted storage

Deleting empty storage pool volumes

Use the DELETE VOLUME command to delete empty volumes. Before deleting a volume, the server issues a confirmation message.

You can delete empty storage pool volumes. For example, to delete an empty volume named WREN03, enter:

```
delete volume wren03
```

On an administrative client, you will receive the following confirmation messages, unless the client is running with the NOCONFIRM option:

```
ANR2200W This command will delete volume WREN03
from its storage pool after verifying that the volume
contains no data.
Do you wish to proceed? (Y/N)
```

Volumes in a shred pool (DISK pools only) are not deleted until shredding is completed. See “Securing sensitive client data” on page 499 for more information.

After you respond yes, the server generates a background process to delete the volume.

The command can be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

Deleting storage pool volumes that contain data

To prevent accidental deletion of backed-up, archived, or space-managed files, the server does not allow you to delete a volume that contains user data unless you specify DISCARDDATA=YES on the DELETE VOLUME command. After all files have been deleted from the volume, the server deletes the volume from the storage pool.

Tips:

1. The Tivoli Storage Manager server will not delete archive files that are on deletion hold.

2. If archive retention protection is enabled, the Tivoli Storage Manager server will delete only archive files whose retention period has expired.
3. Volumes in a shred pool (DISK pools only) are not deleted until the data on it is shredded. See “Securing sensitive client data” on page 499 for more information.

For example, to discard all data from volume WREN03 and delete the volume from its storage pool, enter:

```
delete volume wren03 discarddata=yes
```

The server generates a background process and deletes data in a series of batch database transactions. After all files have been deleted from the volume, the server deletes the volume from the storage pool. If the volume deletion process is canceled or if a system failure occurs, the volume might still contain data. Reissue the DELETE VOLUME command and explicitly request the server to discard the remaining files on the volume.

To delete a volume but not the files it contains, move the files to another volume. See “Moving data from one volume to another volume” on page 349 for information about moving data from one volume to another volume.

Residual data: Even after you move data, residual data may remain on the volume because of I/O errors or because of files that were previously marked as damaged. (Tivoli Storage Manager does not move files that are marked as damaged.) To delete any volume that contains residual data that cannot be moved, you must explicitly specify that files should be discarded from the volume.

Part 3. Managing client operations

Installations of Tivoli Storage Manager typically include backup-archive clients, Tivoli Storage Manager for Space Management (HSM clients), and application clients. You must register these clients with the server. Other priorities include managing their access to the server, managing client data, and scheduling operations such as backing up and archiving.

Chapter 11. Adding client nodes

When adding client nodes, the server views its registered clients as nodes that require services and resources from the server.

When the Tivoli Storage Manager server is installed, the Tivoli Storage Manager backup-archive client and the administrative client are installed on the same server by default. However, many installations of Tivoli Storage Manager include remote clients, and application clients on other servers, often running on different operating systems.

The term “nodes” indicate the following type of clients and servers that you can register as client nodes:

- Tivoli Storage Manager backup-archive clients
- Tivoli Storage Manager application clients, such as Tivoli Storage Manager for Mail clients
- Tivoli Storage Manager for Space Management (HSM client)
- Tivoli Storage Manager source server registered as a node on a target server
- Network-attached storage (NAS) file server using NDMP support

Each node must be registered with the server and requires an option file with a pointer to the server.

For details on many of the topics in this chapter, refer to the *Backup-Archive Clients Installation and User's Guide*.

Related concepts

“Accepting default closed registration or enabling open registration” on page 368

“Overview of clients and servers as nodes”

Related tasks

“Installing client node software” on page 368

“Registering nodes with the server” on page 368

Related reference

“Connecting nodes with the server” on page 372

“Comparing network-attached nodes to local nodes” on page 374

Overview of clients and servers as nodes

Each backup-archive client, Tivoli Storage Manager for Space Management (HSM client), application client, and source server is given a node name when it is registered as a node with the Tivoli Storage Manager server. The server considers each as a node that requires services and resources from the server.

Typically a node is equivalent to a server, as in the case of a backup-archive client that is installed on a user's computer for file system backups. However, multiple nodes can exist on a single server. For example, a Structured Query Language (SQL) server can contain both a Tivoli Storage Manager for SQL server application client for database and transaction log backups, and a Tivoli Storage Manager backup-archive client for file system backups.

Installing client node software

Administrators can install backup-archive clients, application clients, or Tivoli Storage Manager for Space Management clients using four different methods.

The following are the three methods for installing client node software:

- Install directly from the CD
- Transfer installable files from the CD to a target server
- Create client software images and install the images

For more information about installing:

- Client software, refer to *Backup-Archive Clients Installation and User's Guide*.
- System Storage Archive Manager application client software, refer to the application client documentation for your particular client.
- Tivoli Storage Manager application client software, refer to the application client documentation for your particular client.

Use the related procedures to configure a node after it is installed.

Registering nodes with the server

Administrators can register Tivoli Storage Manager clients, application clients, and Tivoli Storage Manager for Space Management (HSM clients) as client nodes.

When a node is registered, Tivoli Storage Manager automatically creates an administrative user ID with client owner authority over the node. You can use this administrative user ID to access the Web backup-archive client from remote locations through a Web browser. If an administrative user ID already exists with the same name, an administrative user ID is not automatically defined.

Tip: To connect to a Web backup-archive client directly from a supported Web browser or from a hyperlink in the Web administrative Enterprise Console, you must specify the node's URL and port number during the registration process or later update the node with this information.

Related concepts

"Overview of remote access to web backup-archive clients" on page 393

Accepting default closed registration or enabling open registration

Before a user can request Tivoli Storage Manager services, the node must be registered with the server.

Closed registration is the default. The administrator must register client nodes when registration is set to closed.

Open registration allows the client nodes to register their node names, passwords, and compression options. On UNIX and Linux systems, only the root user can register a client node with the server.

With either registration mode, by default, an administrative user ID with client owner authority is created over the node.

Important: Changes to the registration process do not affect existing registered client nodes.

Adding nodes with closed registration

To add a node with closed registration, an administrator can use the REGISTER NODE command to register the node and specify the initial password.

The administrator can also specify the following optional parameters:

- Contact information.
- The name of the policy domain to which the node is assigned.
- Whether the node compresses its files before sending them to the server for backup and archive.
- Whether the node can delete backups and archives from server storage.
- The name of a client option set to be used by the node.
- Whether to force a node to change or reset the password.
- The type of node being registered.
- The URL address used to administer the client node.
- The maximum number of mount points the node can use.
- Whether the client node keeps a mount point for an entire session.
- The transfer path used when the node sends data.
- The transfer path used when data is read for a client.
- Whether the server or client node initiates sessions.
- The IP address of the node.
- The low level address of the node.

Adding nodes with open registration

To add a node with open registration, the server prompts the user for a node name, password, and contact information the first time a user attempts to connect to the server.

With open registration, the server automatically assigns the node to the STANDARD policy domain. The server by default allows users to delete archive copies, but not backups stored in server storage.

1. Enable open registration by entering the following command from an administrative client command line:

```
set registration open
```

For examples and a list of open registration defaults, refer to the *Administrator's Reference*.

2. To change the defaults for a registered node, issue the UPDATE NODE command.

Node compression considerations

When you enable compression, it reduces network utilization and saves server storage, but causes additional central processing unit (CPU) overhead to the node. Data compression is recommended only when there is insufficient network capacity.

Attention: Use either client compression or drive compression, but not both.

To optimize performance or to ease memory constraints at the workstation, an administrator can restrict file compression. You can select one of three options:

- Compress files.
- Do not compress files.
- Use the value set in the COMPRESSION option.

Set the COMPRESSION option in the client system options file or in the API configuration file.

On a UNIX or a Linux system, a root user can define the COMPRESSION option in the dsm.opt client options file.

Related concepts

“Data compression” on page 218

Registering nodes with client options sets

Administrators can use client options sets in conjunction with the client options file to register nodes with the server.

Client option sets are considered advanced implementation.

Specify an option set for a node when you register or update the node. Issue the following example command:

```
register node mike pass2eng cloptset=engbackup
```

The client node MIKE is registered with the password pass2eng. When the client node MIKE performs a scheduling operation, the schedule log entries are kept for 5 days.

Related reference

“Managing client option files” on page 411

Registering a network-attached storage file server as a node

To include a network-attached storage (NAS) file server as a node that Tivoli Storage Manager can back up and restore with NDMP (network data management protocol) operations, you can register the file server as a NAS node. Data that is backed up from the NAS file server will be associated with the NAS node name.

The REGISTER NODE and UPDATE NODE commands have a default parameter of TYPE=CLIENT.

To register a NAS file server as a node, specify the TYPE=NAS parameter. Issue the following command, which is an example, to register a NAS file server with a node name of NASXYZ and a password of PW4PW:

```
register node nasxyz pw4pw type=nas
```

You must use this same node name when you later define the corresponding data mover name.

Related reference

Chapter 8, “Using NDMP for operations with NAS file servers,” on page 167

Registering a source server as a node on a target server

A virtual volume is a volume that appears to be a sequential media volume on a source server. The volume is actually stored as an archive file on a target server.

To use virtual volumes, register the source server as a client node on the target server.

The REGISTER NODE and UPDATE NODE commands have a default parameter of TYPE=CLIENT.

Register a source server as a node. Specify the **TYPE=SERVER** parameter.

Related tasks

“Using virtual volumes to store data on another server” on page 712

Registering an API to the server

Workstation users can request Tivoli Storage Manager services by using an application that uses the Tivoli Storage Manager API.

An administrator can issue the REGISTER NODE command to register the workstation as a node.

Setting the compression option

There are several ways to determine the compression for applications that use the Tivoli Storage Manager API.

You can determine the compression by using one of the following methods:

- An administrator during registration who can:
 - Require that files are compressed
 - Restrict the client from compressing files
 - Allow the application user or the client user to determine the compression status
- The client options file. If an administrator does not set compression on or off, Tivoli Storage Manager checks the compression status that is set in the client options file. The client options file is required, but the API user configuration file is optional.
- One of the object attributes. When an application sends an object to the server, some object attributes can be specified. One of the object attributes is a flag that indicates whether or not the data has already been compressed. If the application turns this flag on during either a backup or an archive operation, then Tivoli Storage Manager does not compress the data a second time. This process overrides what the administrator sets during registration.

For more information on setting options for the API and on controlling compression, see *IBM Tivoli Storage Manager Using the Application Program Interface*

Setting the file deletion option

An administrator can set the file deletion option for applications that use the Tivoli Storage Manager application programming interface (API).

The administrator who sets the file deletion option can use the following methods:

- An administrator during registration
 - If an administrator does not allow file deletion, then an administrator must delete objects or file spaces that are associated with the workstation from server storage.
 - If an administrator allows file deletion, then Tivoli Storage Manager checks the client options file.
- An application using the Tivoli Storage Manager API deletion program calls
 - If the application uses the `dsmDeleteObj` or `dsmDeleteFS` program call, then objects or files are marked for deletion when the application is executed.

Connecting nodes with the server

The client options file connects each node to the server. Administrators and users on all platforms can modify their client options file (`dsm.opt`) with a text editor. Client options files can be updated differently across platforms.

Important: If any changes are made to the `dsm.opt` file, the client must be restarted for changes in the options file to have any affect.

The client options file `dsm.opt` is located in the client, application client, or host server directory. If the file does not exist, copy the `dsm.smp` file. Users and administrators can edit the client options file to specify:

- The network address of the server
- The communication protocol
- Backup and archive options
- Space management options
- Scheduling options

Related concepts

“Creating or updating a client options file” on page 373

Required client options

Each node requires a client options file. Each client options file must contain the network address of the Tivoli Storage Manager server and other communication options that allow the node to communicate with the server.

Figure 60 on page 373 shows the contents of a client options file that is configured to connect to the server by using TCP/IP. The communication options specified in the client options file satisfy the minimum requirements for the node to connect with the server.

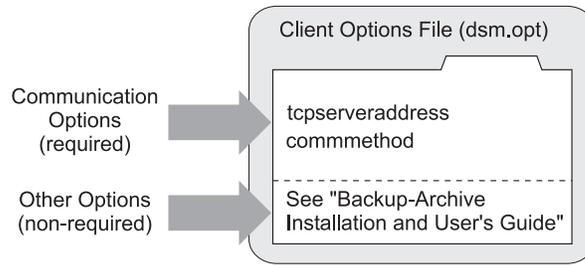


Figure 60. Client options file

Many non-required options are available that can be set at any time. These options control the behavior of Tivoli Storage Manager processing.

Refer to *Backup-Archive Clients Installation and User's Guide* for more information about non-required client options.

UNIX and Linux client options

For UNIX and Linux, client options are located in the client systems options file, client user options file, and the include-exclude options file.

Clients on other platforms use a single options file.

Creating or updating a client options file

Creating or updating client options files depends on the client platform. You might use a text editor, the client configuration wizard, or the client options file wizard.

Using a text editor to create or configure a client options file

All options files (dsm.opt) can be edited with a text editor. Anyone can edit the client options file if they have access to the directory where the node software is installed.

Editing individual options files is the most direct method, but may not be suitable for sites with many client nodes.

Using the client configuration wizard to create or update a client options file

When a local backup-archive client GUI starts initially and Tivoli Storage Manager does not find an options file, a setup wizard guides the user through the configuration process.

From the backup-archive client GUI, the client can also display the setup wizard by selecting **Utilities** → **Setup Wizard**. The user can follow the panels in the setup wizard to browse Tivoli Storage Manager server information in the Active Directory. The user can determine which server to connect to and what communication protocol to use.

Restriction: This wizard is not available for the Web client.

Comparing network-attached nodes to local nodes

A Tivoli Storage Manager environment can be either a server and client on the same server (stand-alone environment) or a server and network-attached clients (network environment).

The stand-alone environment of Tivoli Storage Manager consists of a backup-archive client and an administrative client on the same computer as the server. There is nothing more to do to connect the client. This is shown in Figure 61.

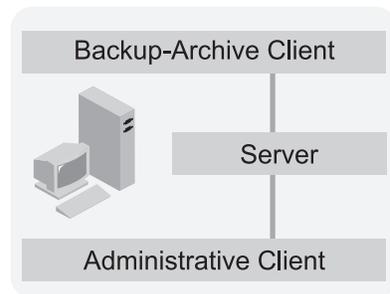


Figure 61. Stand-alone environment

Figure 62 shows that a network environment Tivoli Storage Manager consists of a backup-archive client and an administrative client on the same computer as the server. However, network-attached client nodes can also connect to the server.

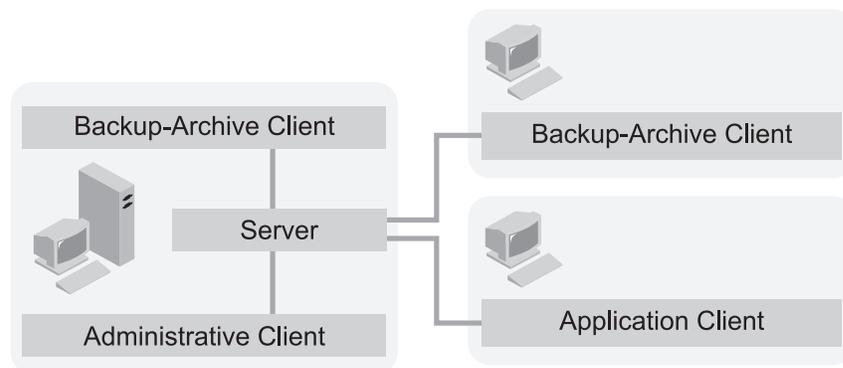


Figure 62. Network environment

Each client requires a client options file. A user can edit the client options file at the client node. The options file contains a default set of processing options that identify the server, communication method, backup and archive options, space management options, and scheduling options.

Adding clients through the administrative command line client

The administrator can register nodes by issuing the REGISTER NODE command.

For more information, refer to *Administrator's Reference*.

Enabling open registration

The default registration mode at installation is closed.

To change the default to open so users can register their own client nodes, issue the following command:

```
set registration open
```

Example: registering three client nodes using the administrative command line

For this example, the goal is to register three workstations from the engineering department and assign them to the ENGPOLDOM policy domain.

Before you can assign client nodes to a policy domain, the policy domain must exist.

You want to let users delete backed up or archived files from storage pools. From an administrative client, you can use the macro facility to register more than one client node at a time.

1. Create a macro file named REGENG.MAC, that contains the following REGISTER NODE commands:

```
register node ssteiner choir contact='department 21'  
domain=engpoldom archdelete=yes backdelete=yes  
register node carolh skiing contact='department 21, second shift'  
domain=engpoldom archdelete=yes backdelete=yes  
register node mab guitar contact='department 21, third shift'  
domain=engpoldom archdelete=yes backdelete=yes
```

2. Issue the MACRO command.

```
macro regeng.mac
```

For information on the MACRO command, see the *Administrator's Reference*.

Related concepts

Chapter 14, "Implementing policies for client data," on page 435

Chapter 12. Managing client nodes

If you have already installed and configured your client nodes, you must be able to manage and control their access to the server.

The Tivoli Storage Manager server views its registered clients, application clients, and source servers as nodes. The term “client node” refers to the following type of clients and servers:

- Tivoli Storage Manager backup-archive clients
- Tivoli Storage Manager application clients, such as Tivoli Storage Manager for Mail clients
- Tivoli Storage Manager source servers registered as nodes on a target server
- Network-attached storage (NAS) file servers using network data management protocol (NDMP) support

Related concepts

“Accepting default closed registration or enabling open registration” on page 368

“Overview of clients and servers as nodes” on page 367

Related tasks

“Installing client node software” on page 368

“Registering nodes with the server” on page 368

Related reference

“Connecting nodes with the server” on page 372

“Comparing network-attached nodes to local nodes” on page 374

Managing nodes

From the perspective of the server, each client and application client is a node requiring IBM Tivoli Storage Manager services.

Client nodes can be local or remote to the server.

Administrators can perform the following activities when managing client nodes.

Task	Required Privilege Class
Updating, renaming, locking, or unlocking any client nodes	System or unrestricted policy
Updating, renaming, locking, or unlocking client nodes assigned to specific policy domains	System, unrestricted policy, or restricted policy for those domains
Displaying information about client nodes or file spaces	Any administrator
Deleting any client nodes	System or unrestricted policy
Removing client nodes assigned to specific policy domains	System, unrestricted policy, or restricted policy for those domains
Managing client access authority levels	System

Related reference

“Defining client nodes and file spaces” on page 398

“Comparing network-attached nodes to local nodes” on page 374

Managing client nodes across a firewall

In most cases, the IBM Tivoli Storage Manager server and clients can work across a firewall or the server can securely manage client backup and restore operations and administrative functions across a firewall. Because every firewall is different, the firewall administrator might need to consult the instructions for the firewall software or hardware in use.

IBM Tivoli Storage Manager has two methods for enabling communication between the client and the server across a firewall: client-initiated communication and server-initiated communication. To allow either client-initiated or server-initiated communication across a firewall, client options must be set in concurrence with server parameters on the REGISTER NODE or UPDATE NODE commands. Enabling server-initiated communication overrides client-initiated communication, including client address information that the server may have previously gathered in server-prompted sessions.

Attention: A firewall should not be configured in a manner that causes terminations of sessions in use by either the server or the storage agent. When a firewall terminates a valid session, unpredictable problems can occur which make processes and sessions appear to stop due to communication input/output (I/O). Configuring IBM Tivoli Storage Manager components with known ports helps when you want to exclude IBM Tivoli Storage Manager sessions from timeout restrictions.

Client-initiated sessions

You can enable clients to communicate with a server across a firewall by opening the TCP/IP port for the server and modifying the `dsmserv.opt` file.

1. To enable clients to communicate with a server across a firewall, open the TCP/IP port for the server on the TCPPORT option in the `dsmserv.opt` file. The default TCP/IP port is 1500. When authentication is turned on, the information that is sent over the wire is encrypted.
2. To enable administrative clients to communicate with a server across a firewall, open the TCP/IP port for the server on the TCPADMINPORT option in the `dsmserv.opt` file. The default TCP/IP port is the TCPPORT value. When authentication is turned on, the information that is sent over the wire is encrypted. See the *Backup-Archive Clients Installation and User's Guide* for more information.

Note:

1. If the TCPADMINPORT option is specified, sessions from clients without administration authority can be started on the TCPPORT port only. If the server `dsmserv.opt` specifies TCPADMINPORT that is different from the TCPPORT and sets ADMINONCLIENTPORT to NO, then administrative client sessions can be started on the TCPADMINPORT port only.
2. You can specify either IPv4 or IPv4/IPv6 in the COMMMETHOD option when you start the server, storage agent, client, or API application. The same port numbers are used by the server, storage agent, client, or API application for both IPv4 and IPv6.

IPv6 address formats are acceptable for all functions that support IPv6. However, if you use IPv6 addresses for functions that do not support IPv6, communications fail. The following functions do not support IPv6:

- Network data management protocol (NDMP): backing up and restoring storage pools, copying and moving data
- Automated cartridge system library software (ACSL)
- Simple network management protocol (SNMP)
- Centera device support
- Shared memory protocol
- Windows Microsoft Management Console functions
- Tivoli Enterprise Console[®] support
- Administration Center

Remember: You can continue to use IPv4 address formats for the following functions:

- NDMP: backing up and restoring storage pools, copying and moving data
- ACSLS
- SNMP
- Centera device support
- Shared memory protocol
- Windows Microsoft Management Console functions
- Administration Center

If you select the CLIENTORSERVER option of the **SESSIONINITIATION** server parameter, the client may start sessions with the server. Or, server-prompted scheduling may be used to prompt the client to connect to the server.

Server-initiated sessions

To limit the start of backup-archive client sessions to the IBM Tivoli Storage Manager server, you must specify this on the server and also synchronize the information in the client option file.

In either the REGISTER NODE or UPDATE NODE command, select the SERVERONLY option of the **SESSIONINITIATION** parameter. Provide the HLADDRESS and LLADDRESS client node addresses. For example,

```
register node fran secretpw hladdress=9.11.521.125 lladdress=1501
  sessioninitiation=serveronly
```

The **HLADDRESS** specifies the IP address of the client node, and is used whenever the server contacts the client. The **LLADDRESS** specifies the low level address of the client node and is used whenever the server contacts the client. The client node listens for sessions from the server on the **LLADDRESS** port number.

If **SESSIONINITIATION=SERVERONLY** for a node defined on the IBM Tivoli Storage Manager server, the client must have **SESSIONINITIATION=SERVERONLY** in its option file. In addition, the TCP/IP address of the client must correspond to the information supplied with the **HLADDRESS** server parameter. Finally, **TCPCLIENTPORT** in the client option file must correspond to the information supplied with the **LLADDRESS** server parameter, or the server will not know how to contact the client.

Note:

1. If you switch from server-prompted to server-initiated sessions, the server will discard any addressing information it had and will use only the information from the **HLADDRESS** and **LLADDRESS** parameters of the REGISTER NODE and UPDATE NODE commands in contacting the client.
2. The server uses Domain Name System (DNS) to determine the name of client nodes. If your DNS is not correctly configured, there may be delays or failures in looking up names. The DNSLOOKUP option is available to restrict usage of DNS services. It can be changed dynamically through the SETOPT DNSLOOKUP command.

Table 41. Server-Initiated sessions

Setting or parameter on the IBM Tivoli Storage Manager server:	Location on the IBM Tivoli Storage Manager server	Must match this on the client:	Location on the client
SESSIONINITIATION=SERVERONLY	REGISTER NODE or UPDATE NODE command	SESSIONINITIATION=SERVERONLY	client option file
HLADDRESS	REGISTER NODE or UPDATE NODE command	TCP/IP address	TCP/IP address
LLADDRESS	REGISTER NODE or UPDATE NODE command	TCPCLIENTPORT	client option file

Updating client node information

You can issue the UPDATE NODE command to update information such as the client's assigned policy domain, the user's password or contact information, and the client option set used by the node.

Update client node TOMC to prevent it from deleting archived files from storage pools by entering the following example command:

```
update node tomc archdelete=no
```

Automatically updating backup-archive clients using the Administration Center

You can use the Tivoli Storage Manager Administration Center to download and install backup-archive client deployment packages onto Windows workstations. The feature allows you to schedule updates to specific backup-archive client nodes and to monitor the progress of the deployed packages.

The following steps give you an overview on how to automate the backup-archive client deployment feature:

1. Install the Tivoli Storage Manager server, V6.2 or later.
2. Install the Tivoli Storage Manager V6.2 Administration Center.
3. Issue the SET SERVERHLADDRESS command to identify the host or IP address of the server, enabling the reporting of deployment results.
4. Enable any applicable client events and assess their appropriate size and pruning duration. By enabling the events, the deployment manager can propagate the deployment messages to the server activity log. Determine the appropriate activity log size and pruning duration that can give you enough time to observe and react to the deployment results.
5. Configure the server for backup-archive client deployments. The **Configure Server for Client Deployment** wizard in the Administration Center can guide you through the process.

6. Use the Administration Center to check for new client deployment packages on the FTP site (`ftp://public.dhe.ibm.com/storage/tivoli-storage-management/`). Import the client deployment packages from the FTP site to the Tivoli Storage Manager server or to an internal FTP site. The **Import Client Deployment Packages** wizard can help you make the client deployment packages accessible to the Administration Center.
7. Review the list of available packages and select the Windows backup-archive clients to update. The **Client Deployment Wizard Scheduler** in the Administration Center can guide you through the process.
8. Verify that the deployment process was successful.

Backup-archive client automatic deployment feature: Overview

The Administration Center backup-archive client deployment packages are posted on the Service FTP site. Use the **Configure Server for Auto Deployment** wizard to set up the servers for deployments.

After configuring each server for deploying packages, the three steps in the process are downloading, moving, and importing the packages.

Downloading

The **Import Client Deployment Packages** wizard accesses the FTP site where the packages are stored and from where you can select the packages to import.

Moving

After you download the packages, they must be moved from the Administration Center workstation to the Tivoli Storage Manager server. The packages must be moved to a location that is referenced by the `IBM_CLIENT_DEPLOY_IMPORT` device class. This device class is created when you configure your server with the **Configure Server for Client Auto Deployment** wizard.

Importing

If you are configured for a local import, the Administration Center finds the packages that are stored locally and starts the process of deploying them. An Administration Center with Web access deploys the packages from the FTP site.

The advantage to using the Administration Center to configure your deployments is that you can define all aspects of the process in the wizard. Overlaps in the schedule are displayed, with high-priority events taking precedence over client deployments.

You can update a scheduled deployment in the **Manage Client Deployments** notebook. To open the notebook, click **Tivoli Storage Manager** → **Manage Servers**, select a server, then select **Manage Client Deployments**.

Backup-archive client automatic deployment requirements

The backup-archive client deployment feature can update multiple Windows backup-archive clients with V6.2 deployment packages, simultaneously. The deployment is configured and scheduled in the Administration Center using the **Configure Server for Client Deployment** wizard.

Your IBM Tivoli Storage Manager Windows backup-archive client must be at version 5.4.0 or later to use the deployment feature. The deployment feature does not install new backup-archive clients.

The Tivoli Storage Manager server must be at V6.2 and later to use the deployment feature. All package deployments upgrade the Windows backup-archive client to V6.2 or later. To use the feature, the following conditions on the backup-archive client must be true:

- The PASSWORDACCESS option must be set to *generate*.
- The client acceptor (CAD) or backup-archive client scheduler must be running at the time of the deployment. The backup-archive client is deployed from the server as a scheduled task.

Configuring the server to perform backup-archive client automatic deployments

Before you can use the backup-archive client deployment feature, you must configure every server from which you want to deploy packages. The Administration Center contains the **Configure Client Auto Deployment** wizard that can help guide you through the configuration process.

To access the **Configure Client Auto Deployment** wizard, click **Tivoli Storage Manager** → **Manage Servers**. Select a server from the table and then select **Configure Client Auto Deployment** from the table actions.

The wizard guides you in setting up the location where imported packages are to be stored, and how long they are stored.

Importing backup-archive client deployment packages with the Administration Center wizard

The automatic deployment feature imports backup-archive client packages from the Service FTP site and makes them available for deployment to Windows Tivoli Storage Manager backup-archive clients. You can perform this process using the import client deployment packages wizard.

Before you use the import client deployment packages wizard, ensure that you configure the server using the configure server for client auto deployment wizard.

The View Available Client Deployment Packages portlet displays a table of all available packages. You can either import the available deployment packages, check for new packages on the FTP site, or refresh the table from a local copy.

To use the import client deployment packages wizard, complete the following steps:

1. Open the Administration Center.
2. Click **Tivoli Storage Manager** → **Manage Servers**.
3. Access the wizard by selecting **View Client Deployment Packages** → **Import Client Deployment Packages**.

Deploying client software without direct Web access

When your Administration Center workstation does not have direct Web access to the Service FTP site, you must adjust the `catalog.properties` file.

The `properties` file holds critical information for the deployment feature and allows the Administration Center to find and import packages. The `catalog.properties` file is updated automatically when you configure the server to run deployments through the Administration Center.

In the following task descriptions, `<TIP_HOME>` is the root directory for your Tivoli Integrated Portal installation. For Windows, the `<TIP_HOME>` default location is `C:\Program Files\tivoli\tsm`. If the server does not have Web access, you must edit the `catalog.properties` file to point to the local `catalog.xml` file. The `catalog.properties` file is located in the following directories:

- For Windows: `<TIP_HOME>\products\tsm\clientDeployCatalog`
- For all other platforms: `<TIP_HOME>/products/tsm/clientDeployCatalog`

You can copy the packages to the server from media and then access the packages as if you are connected to the FTP site.

The `catalog.properties` file directs the Administration Center to the location of the deployment packages. Complete the following steps to schedule a client deployment without direct Web access:

1. Move the packages to a local FTP server that is configured for anonymous access.
2. Configure the servers for deployments. Access the configure server for client deployments wizard by clicking **Tivoli Storage Manager** → **Manage Servers**. Select a server and then select **Configure Automatic Client Deployment** from the action list.
3. Edit the `catalog.properties` file to point to the local `catalog.xml` file. See the following example of the `catalog.properties` file:

```
base.url=ftp://public.dhe.ibm.com/storage/tivoli-storage-management
```

where `public.dhe.ibm.com` can be changed to the address of your local FTP server.

4. Import the client deployment packages to the servers.
 - a. Click **Tivoli Storage Manager** → **Manage Servers**.
 - b. Select a server and then select **View all available packages** → **Import Client Deployment Packages**.
 - c. Use the import client deployment packages wizard to select **The client deployment packages are already downloaded. Automatically move the packages and run the import command**.
5. Schedule the deployment. The **Client Deployment Wizard Scheduler** can be accessed from several locations and is located in the table action menu.

Scheduling a backup-archive client automatic deployment

After you configure the server for backup-archive client deployments and import the available packages, you can schedule a deployment. If you want to process the client deployment schedule immediately, set the client scheduler to the SCHEDMODE PROMPTED setting.

The scheduling wizard is accessible from several Administration Center panels. You can access the client deployment wizard scheduler through one of the following actions:

- Click **Manage Servers** and select a server. Then select **Manage Client Deployments** from the action menu and open the **Schedules** tab or the **Packages** tab. Select the table action **Schedule an Auto Deployment**.
- Click **Client Nodes and Backupsets** and open the Client Nodes portlet. Select the table action from the node table **Schedule an Auto Deployment**.
- Click **Client Nodes and Backupsets** and open the Client Nodes portlet. Select the Search view, and in the table of nodes, select the table action **Schedule an Auto Deployment**.

You can schedule your deployments around your routine IBM Tivoli Storage Manager activities. When scheduling client deployments, place those schedules on a lower priority than regular storage management tasks (for example, back up, archive, restore, and retrieve).

The client deployment feature contains an overlap table that shows you when other operations overlap with a scheduled deployment. From the **Schedule** tab of the **Manage Client Auto Deployments** panel, select to create or modify a schedule. When you arrive at the Client Node panel, select **View Schedule Overlaps** from the table action menu.

The client deployment wizard scheduler gives you the option to allow a restart of the client operating system after the deployment completes. Restarting can impact any critical applications running on the client operating system.

Important: If you use AUTODEPLOY=NOREBOOT in your command, the client operating system does not restart after the deployment completes. Restarting can impact any critical applications running on the client operating system. Applications that are not Tivoli Storage Manager applications running on the client operating system might not be restarted automatically. In most cases, the installation can complete without restarting the client operating system. There are rare cases where the deployment manager cannot detect the restart. For example, if client processes are started from a script. In these cases, the new backup-archive client installation continues, but a manual restart of the client computer is required. If you do not check this box and the deployment requires a restart, most deployments are stopped and the original backup-archive client is left intact.

Command-line backup-archive client automatic deployment feature: overview

The backup-archive client automatic deployment feature helps you schedule updates to one or more client nodes with backup-archive client deployment packages. The deployment packages can be backup-archive client updates or full releases.

The following list outlines the deployment restrictions:

- IBM Tivoli Storage Manager backup-archive clients that are at levels earlier than 5.4 cannot be updated with the available deployment packages.
- Operating systems that are not supported, such as Windows 2000, cannot be upgraded through the deployment process.

The following list outlines the deployment requirements:

- You must upgrade the Tivoli Storage Manager server to V6.2 and the backup-archive client to V5.4 or later to use the deployment feature.
- All deployment packages contain Tivoli Storage Manager Windows backup-archive client V6.2 and later. The deployment feature does not use deployment packages that contain backup-archive clients that are at levels earlier than 6.2.
- You must use the SET SERVERHLADDRESS command for all automatic client deployments.

Related tasks

“Importing the target level to the server” on page 386

“Defining a schedule for an automatic deployment” on page 387

“Verifying the backup-archive client deployment results” on page 388

Related reference

“Using the command-line interface to configure the server for a backup-archive client deployment”

Using the command-line interface to configure the server for a backup-archive client deployment

You can access the backup-archive client automatic deployment feature from the command-line interface and schedule client deployments for packages that you acquire from the FTP site.

The following list outlines the deployment requirements:

- Before using the backup-archive client deployment feature, you must configure the server.
- You must be an administrator on the server to issue backup-archive deployment commands related to configuring the server and importing the deployment packages.
- To schedule a backup-archive client deployment for a node, you must have system privileges or domain privileges on the domain to which the node belongs.

The following example command can be used to configure the server to deploy backup-archive client packages with the command-line interface:

```

define devclass ibm_client_deploy_import devtype=file directory=import_directory
define stgpool stgpool_name storage_dc_name maxxscratch=20

define domain ibm_client_deploy

define policyset ibm_client_deploy ibm_client_deploy

define mgmtclass ibm_client_deploy ibm_client_deploy ibm_client_deploy

define copygroup ibm_client_deploy ibm_client_deploy ibm_client_deploy
standard type=archive destination=stgpool_name retver=retention_value

assign defmgmtclass ibm_client_deploy ibm_client_deploy ibm_client_deploy

activate policyset ibm_client_deploy ibm_client_deploy

set serverhladdress=server.serveraddress.com

```

where:

- *ibm_client_deploy_import* is the temporary location from where the deployment packages are imported. This parameter is defined by the deployment manager.
- *import_directory* is a previously defined directory that is accessible from the server.
- *stgpool_name* is the name of a storage pool of your choosing where the deployment packages are stored on the server. The storage pool name is based on a previously defined device class. That device class is different from the one which is used to perform IMPORT operations.
- *storage_dc_name* represents the device class where the deployment packages are stored on the server.
- *retention_value* (RETVER) of the DEFINE COPYGROUP command sets the retention time for the package. You can set it to NOLimit or to a number of days. If you specify a number, you must take into account the number of days that the package was on the FTP site. The days spent on the FTP site are included in the number.
The default for the Administration Center is five years.
- *server.serveraddress.com* is the server IP address or host name from which you scheduled the client automatic deployment

Importing the target level to the server

After you acquire the backup-archive client deployment packages, you must import them to the servers where they can be deployed.

Ensure that you configure the server for backup-archive client automatic deployments before importing the packages.

Complete the following steps to import deployment packages to the server:

1. Issue the IMPORT command. For example:

```

import node ibm_client_deploy_win devclass=upgradedev
filedata=archive mergefilespace=yes
volumenames=volname1.exp

```

where:

upgradedev is the file device class name.

volname1.exp is the deployment package name. You can also use a comma-separated list of package names.

If you want to view the progress, issue the QUERY PROCESS command.

2. Verify that the packages are in a location that the server can reach. Enter the following command:

```
select * from archives where node_name='ibm_client_deploy_win'
```

where archives is the type of file that is imported through the IMPORT NODE command.

Related reference

“Using the command-line interface to configure the server for a backup-archive client deployment” on page 385

Defining a schedule for an automatic deployment

You can schedule a backup-archive client deployment after you import the packages to the server.

Complete the following steps to create a schedule for an automatic client deployment:

1. Issue the DEFINE SCHEDULE command. For example:

```
define schedule domain=standard upgrade1 action=deploy
objects=
"\\ibm_anr_win\c$\TSM\MAINTENANCE\CLIENT\V6R2\WINDOWS\X64\V620\V6200\*
..\IBM_ANR_WIN\" options= "-fromnode=ibm_client_deploy_win
-postnschedulecmd=..\IBM_ANR_WIN\v6200\ deployclient.bat
schedule=upgrade1 autodeploy=noreboot nodeinfo=TBD"
-sub=yes -replace=all" startdate="11/20/2009"
```

where

standard is the domain name

upgrade1 is the name of the schedule that you are defining

\\ibm_anr_win\c\$\TSM\MAINTENANCE\CLIENT\V6R2\WINDOWS\X64\V620\V6200* is an example of how to specify a set of deployment packages

..\ibm_anr_win\ is an example of the destination for the deployment packages

Important: If you use AUTODEPLOY=NOREBOOT in your command, the client operating system does not restart after the deployment completes. Restarting can impact any critical applications running on the client operating system. Applications that are not Tivoli Storage Manager applications running on the client operating system might not be restarted automatically. In most cases, the installation can complete without restarting the client operating system. There are rare cases where the deployment manager cannot detect the restart. For example, if client processes are started from a script. In these cases, the new backup-archive client installation continues, but a manual restart of the client computer is required.

2. Issue the following command to update the node:

```
update node node_name targetlevel=v.r.m.f
```

where

node_name is the name of the node that you are updating.

v.r.m.f is the version, release, modification, and fix level of the package that you want to deploy. For example, 6.2.1.0.

Verifying the backup-archive client deployment results

You can issue several different commands to verify the results of a backup-archive client deployment.

Complete the following steps to confirm the deployment process:

1. Issue the QUERY EVENT command to determine if the scheduled deployment started successfully.

```
query event schedule_name format=detailed
```

If the status for the scheduled event is *Completed*, the deployment has started on the client operating system and a session with the server is activated. The session includes messages written to the server activity log. The activity log messages offer more information about the status of the deployment.

2. Issue the QUERY ACTLOG command to check the activity log on the server:

```
query actlog begindate=03/11/2010 begintime=00:00:01 node=testsrv
```

One result of the QUERY ACTLOG command is the publishing of the ANE4200I message reports. Message ANE4200I displays the status of the deployment and the *session number*. You can use the session number to search for more deployment information.

3. Issue the QUERY ACTLOG command with the session number as the target.

```
query actlog sessnum=778 begindate=03/11/2010 begintime=00:00:01 node=testsrv
```

4. Issue the QUERY NODE command:

```
query node testsrv format=detailed
```

Renaming client nodes

You can rename a client node by issuing the RENAME NODE command. You might need to rename a client node if the workstation network name or host name changes. For example, with UNIX and Linux clients, users define their node name based on the value returned by the HOSTNAME command.

When users access the server, their IBM Tivoli Storage Manager user IDs match the host name of their workstations. If the host name changes, you can update a client node user ID to match the new host name.

To rename CAROLH to ENGNODE, issue the following example command:

```
rename node carolh engnode
```

ENGNODE retains the contact information and access to backup and archive data that belonged to CAROLH. All files backed up or archived by CAROLH now belong to ENGNODE.

Locking and unlocking client nodes

You can prevent client nodes from accessing the server with the LOCK NODE command. This prevents client nodes from performing functions such as either backup and restore or archive and retrieve.

You can restore a locked node's access to the server with the UNLOCK NODE command.

1. To prevent client node MAB from accessing the server, issue the following example command:

```
lock node mab
```

2. To let client node MAB access the server again, issue the following example command:

```
unlock node mab
```

Related tasks

“Disabling or enabling access to the server” on page 417

Deleting client nodes

You can delete a client node from the server with the REMOVE NODE command. All file spaces that belong to the client node must first be deleted from server storage. After all of the client node's file spaces are deleted, you can delete the node.

Note: Before you can delete a network-attached storage (NAS) node, you must first delete any file spaces, then delete any defined paths for the data mover with the DELETE PATH command. Delete the corresponding data mover with the DELETE DATAMOVER command. Then you can issue the REMOVE NODE command to delete the NAS node.

For our example, we want to remove client node DEBBYG.

1. To delete the DEBBYG file space, issue the following command:

```
delete filespace debbyg * type=any
```

2. To delete the DEBBYG node, issue the following command:

```
remove node debbyg
```

Related tasks

“Deleting file spaces” on page 410

Consolidating multiple clients under a single client node name

Backups of multiple nodes which share storage can be consolidated to a common target node name on the Tivoli Storage Manager server. For example, several nodes in an IBM General Parallel File System (GPFS™) cluster, NODE_1, NODE_2, and NODE_3, can back up to the same node (NODE_OLIV) on the server.

This is useful when the server responsible for performing the backup may change over time, such as with a cluster. Consolidating shared data from multiple servers under a single name space on the Tivoli Storage Manager server means that the directories and files can be easily found when restore operations are required. Backup time can be reduced and clustered configurations can store data with proxy node support. Client nodes can also be configured with proxy node authority to support many of the systems which support clustering failover.

By granting client nodes proxy node authority to another node, you gain the ability to backup, archive, migrate, restore, recall, and retrieve shared data on multiple clients under a single node name on the Tivoli Storage Manager server. When authorized as agent nodes, Tivoli Storage Manager nodes and Tivoli Storage Manager for Space Management (HSM) clients can be directed to backup or restore data on behalf of another node (the target node).

Distributing workloads to reduce backup and restore time

Concurrent operations can reduce backup and restore times in environments such as IBM General Parallel File System (GPFS). Conventional cluster backups are unable to do this with very large file systems because if a password is changed by the Tivoli Storage Manager server, it takes some time to communicate that change to all the nodes.

Administrators must then create scripts that change the passwords manually before they expire. Using proxy node support, it is possible to break up a large GPFS into smaller units for backup purposes and not have password coordination issues.

The following example shows how scheduling would work where workload is distributed, for example in the DB2 Universal Database Enterprise Extended Edition (EEE) environment. In this example, NODE_A, NODE_B and NODE_C all work together to back up this distributed environment, all acting on behalf of NODE_Z. NODE_A directs the backup for all three physical servers. NODE_A either has ASNODENAME=NODE_Z in its local options file or the server (through the DEFINE SCHEDULE command) has indicated that NODE_A needs to request proxy authority to NODE_Z. See the *Backup-Archive Clients Installation and User's Guide* for more information on the ASNODENAME client option.

An administrator can define the schedule that does a DB2 UDB EEE backup on behalf of NODE_Z by issuing the following command:

```
DEFINE SCHEDULE STANDARD BACKUP-SCHED ACTION=INCREMENTAL  
OPTIONS='-ASNODENAME=NODE_Z'
```

Define the association of the schedule to the three nodes:

```
DEFINE ASSOCIATION STANDARD BACKUP-SCHED NODE_A
```

Defining agent and target nodes

Target client nodes own data and agent nodes act on behalf of the target nodes. When granted proxy node authority to a target 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.

Agent nodes are considered traditional nodes in that there is usually a one-to-one relationship between a traditional node and a physical server. A target node can be a logical entity, meaning no physical server corresponds to the node. Or, it can be a predefined node which corresponds to a physical server.

By using the GRANT PROXYNODE command, you can grant proxy node authority to all nodes sharing data in the cluster environment to access the target node on the Tivoli Storage Manager server. QUERY PROXYNODE displays the nodes to which a proxy node relationship was authorized. See the *Administrator's Reference* for more information about these commands.

Relationships between agent nodes and target nodes:

- A client node can be both an agent and a target at the same time. For example, NODE_A is an agent node for NODE_Z. NODE_A can also act as a target node for agent node NODE_D.
- Proxy node relationships can work conversely. For example, NODE_A can be an agent node for NODE_Z, and NODE_Z can be an agent node for NODE_A.

- Proxy node relationships cannot be inherited. For example, if NODE_A is an agent for NODE_X, and NODE_X is an agent for NODE_Z, NODE_A is not automatically an agent for NODE_Z.

Proxy node relationships will not be imported by default; however, the associations can be preserved by specifying the PROXYNODEASSOC option on the IMPORT NODE and IMPORT SERVER commands. Exporting to sequential media maintains proxy node relationships, but exporting to a server requires specifying the PROXYNODEASSOC option on EXPORT NODE and EXPORT SERVER.

Important:

- If a proxy node relationship is authorized for incompatible file spaces, there is a possibility of data loss or other corruption.
- Central command routing or importing of the GRANT PROXYNODE and REVOKE PROXYNODE commands can create access issues.
- The maximum number of mount points for agent nodes should be increased to allow parallel backup operations across the target nodes.

Configuring shared access example

Shared access must be configured according to the nodes on the server and the relationships between the nodes.

The following example shows how to set up proxy node authority for shared access. In the example, client agent nodes NODE_1, NODE_2, and NODE_3 all share the same General Parallel File System (GPFS). Because the file space is so large, it is neither practical nor cost effective to back up this file system from a single client node. By using Tivoli Storage Manager proxy node support, the very large file system can be backed up by the three agent nodes for the target NODE_GPFS. The backup effort is divided among the three nodes. The end result is that NODE_GPFS has a backup from a given point in time.

All settings used in the proxy node session are determined by the definitions of the target node, in this case NODE_GPFS. For example, any settings for DATAWRITEPATH or DATAREADPATH are determined by the target node, not the agent nodes (NODE_1, NODE_2, NODE_3).

Assume that NODE_1, NODE_2 and NODE_3 each need to execute an incremental backup and store all the information under NODE_GPFS on the server. Perform the following steps to set up a proxy node authority for shared access:

1. Define four nodes on the server: NODE_1, NODE_2, NODE_3, and NODE_GPFS. Issue the following commands:

```
register node node_1 mysecretpa5s
register node node_2 mysecret9pas
register node node_3 mypass1secret
register node node_gpfs myhidp3as
```

2. Define a proxy node relationship among the nodes by issuing the following commands:

```
grant proxynode target=node_gpfs agent=node_1,node_2,node_3
```

3. Define the node name and asnode name for each of the servers in the respective dsm.sys files. See the *Backup-Archive Clients Installation and User's Guide* for more information on the NODENAME and ASNODENAME client options. Issue the following commands:

```
nodename node_1
asnodename node_gpfs
```

4. Optionally, define a schedule:

```
define schedule standard gpfs-sched action=macro options="gpfs_script"
```
5. Assign a schedule to each client node by issuing the following commands:

```
define association standard gpfs-sched node_1
define association standard gpfs-sched node_2
define association standard gpfs-sched node_3
```
6. Execute the schedules by issuing the following command:

```
dsmc schedule
```

Displaying information about client nodes

You can display information about client nodes in different aspects.

For example, as a policy administrator, you might query the server about all client nodes assigned to the policy domains for which you have authority. Or you might query the server for detailed information about one client node.

Displaying information about client nodes assigned to specific policy domains

You can display information about client nodes assigned to specific policy domains.

For example, to view information about client nodes that are assigned to STANDARD and ENGPOLDOM policy domains, issue the following command:

```
query node * domain=standard,engpoldom
```

The data from that command may display similar to the following output:

Node Name	Platform	Policy Domain Name	Days Since Last Access	Days Since Password Set	Locked?
JOE	WinNT	STANDARD	6	6	No
ENGNODE	AIX	ENGPOLDOM	<1	1	No
HTANG	Mac	STANDARD	4	11	No
MAB	AIX	ENGPOLDOM	<1	1	No
PEASE	Linux86	STANDARD	3	12	No
SSTEINER	SUN	ENGPOLDOM	<1	1	No
	SOLARIS				

Displaying information about a specific client node

You can view information about specific client nodes.

For example, to review the registration parameters defined for client node JOE, issue the following command:

```
query node joe format=detailed
```

The resulting report may appear similar to the following output:

```

Node Name: JOE
Platform: WinNT
Client OS Level: 5.00
Client Version: Version 5, Release 1, Level 5.0
Policy Domain Name: STANDARD
Last Access Date/Time: 05/19/2002 18:55:46
Days Since Last Access: 6
Password Set Date/Time: 05/19/2002 18:26:43
Days Since Password Set: 6
Invalid Sign-on Count: 0
Locked?: No
Contact:
Compression: Client's Choice
Archive Delete Allowed?: Yes
Backup Delete Allowed?: No
Registration Date/Time: 03/19/2002 18:26:43
Registering Administrator: SERVER_CONSOLE
Last Communication Method Used: Tcp/Ip
Bytes Received Last Session: 108,731
Bytes Sent Last Session: 698
Duration of Last Session (sec): 0.00
Pct. Idle Wait Last Session: 0.00
Pct. Comm. Wait Last Session: 0.00
Pct. Media Wait Last Session: 0.00
Optionset:
URL: http://client.host.name:1581
Node Type: Client
Password Expiration Period: 60
Keep Mount Point?: No
Maximum Mount Points Allowed: 1
Auto Filespace Rename: No
Validate Protocol: No
TCP/IP Name: JOE
TCP/IP Address: 9.11.153.39
Globally Unique ID: 11.9c.54.e0.8a.b5.11.d6.b3.c3.00.06.29.45.c1.5b
Transaction Group Max: 0
Data Write Path: ANY
Data Read Path: ANY
Session Initiation: ClientOrServer
High-level Address: 9.11.521.125
Low-level Address: 1501
Collocation Group Name: minster
Proxynode Target: node_gpfs
Proxynode Agent: node_1
Node Groups:

```

Overview of remote access to web backup-archive clients

With the introduction of the Web backup-archive client, when a client node is registered with an IBM Tivoli Storage Manager server, an identical administrative user ID is created at the same time. This user ID has client owner authority over the node by default.

A Web backup-archive client can be accessed from the Administration Center interface. This allows an administrator with the proper authority to perform backup, archive, restore, and retrieve operations on any server that is running the Web backup-archive client. See the Administration Center for more information.

You can establish access to a Web backup-archive client for help desk personnel that do not have system or policy privileges by granting those users client access authority to the nodes they need to manage. Help desk personnel can then perform activities on behalf of the client node such as backup and restore operations.

A native backup-archive client can log on to IBM Tivoli Storage Manager using their node name and password, or administrative user ID and password. The

administrative user ID password is managed independently from the password that is generated with the `passwordaccess generate client` option. The client must have the option `passwordaccess generate` specified in their client option file to enable use of the Web backup-archive client.

To use the Web backup-archive client from your web browser, specify the URL and port number of the IBM Tivoli Storage Manager backup-archive client computer running the Web client. The browser you use to connect to a Web backup-archive client must be Microsoft Internet Explorer 5.0 or Netscape 4.7 or later. The browser must have the Java Runtime Environment (JRE) 1.3.1, which includes the Java Plug-in software. The JRE is available at <http://java.sun.com/j2se/1.3/download.html>.

A Tivoli Storage Manager Version 5.3 or later Web backup-archive client is required in order to hyperlink from the new Java-based Tivoli Storage Manager administrative client to a Tivoli Storage Manager client computer. If you attempt to hyperlink from the Java-based Tivoli Storage Manager administrative client to an earlier version of the Tivoli Storage Manager Web backup-archive client, you will have to re-enter your credentials.

During node registration, you have the option of granting client owner or client access authority to an existing administrative user ID. You can also prevent the server from creating an administrative user ID at registration. If an administrative user ID already exists with the same name as the node being registered, the server registers the node but does not automatically create an administrative user ID. This process also applies if your site uses open registration.

For more information about installing and configuring the Web backup-archive client, refer to *Backup-Archive Clients Installation and User's Guide*.

Defining node privilege class and client access authorities

Access to a Web backup-archive client requires either client owner authority or client access authority.

Administrators with system or policy privileges over the client node's domain, have client owner authority by default. The administrative user ID created automatically at registration has client owner authority by default. This administrative user ID is displayed when an administrator issues a `QUERY ADMIN` command.

The following definitions describe the difference between client owner and client access authority when defined for a user that has the node privilege class:

Client owner

You can access the client through the Web backup-archive client or native backup-archive client.

You own the data and have a right to physically gain access to the data remotely. You can backup and restore files on the same or different servers, you can delete file spaces or archive data.

The user ID with client owner authority can also access the data from another server using the `-NODENAME` or `-VIRTUALNODENAME` parameter.

The administrator can change the client node's password for which they have authority.

This is the default authority level for the client at registration. An administrator with system or policy privileges to a client's domain has client owner authority by default.

Client access

You can only access the client through the Web backup-archive client. You can restore data only to the original client.

A user ID with client access authority cannot access the client from another server using the `-NODENAME` or `-VIRTUALNODENAME \parameter`.

This privilege class authority is useful for help desk personnel so they can assist users in backing up or restoring data without having system or policy privileges. The client data can only be restored to none other than the original client. A user ID with client access privilege cannot directly access client's data from a native backup-archive client.

Managing client access authority levels

By default, an administrator with system or policy privilege over a client's domain can remotely access clients and perform backup and restore operations.

You can grant client access or client owner authority to other administrators by specifying `CLASS=NODE` and `AUTHORITY=ACCESS` or `AUTHORITY=OWNER` parameters on the `GRANT AUTHORITY` command. You must have one of the following privileges to grant or revoke client access or client owner authority:

- System privilege
- Policy privilege in the client's domain
- Client owner privilege over the node
- Client access privilege over the node

You can grant an administrator client access authority to individual clients or to all clients in a specified policy domain. For example, you may want to grant client access privileges to users that staff help desk environments.

Related tasks

“Example: setting up help desk access to client computers in a specific policy domain” on page 396

Granting client authority

You might have to grant client authority to a user at some time.

1. Issue the following command to grant client access authority to administrator FRED for the LABCLIENT node:

```
grant authority fred class=node node=labclient
```

The administrator FRED can now access the LABCLIENT client, and perform backup and restore. The administrator can only restore data to the LABCLIENT node.

2. Issue the following command to grant client owner authority to ADMIN1 for the STUDENT1 node:

```
grant authority admin1 class=node authority=owner node=student1
```

The user ID ADMIN1 can now perform backup and restore operations for the STUDENT1 client node. The user ID ADMIN1 can also restore files from the STUDENT1 client node to a different client node.

Automatically creating an administrative user ID with client owner authority

When you issue the REGISTER NODE command, by default, the server creates an administrative user ID in addition to the client node. The administrative user ID has client owner authority to the node when the node is defined to the server.

To register client node DESK2, issue the following example command:

```
register node desk2 pass2dsk
```

The following output is an example of this command:

```
ANR2060I Node DESK2 registered in policy domain STANDARD.  
ANR2099I Administrative userid DESK2 defined for OWNER access to node DESK2.
```

The DESK2 client node is registered, in addition to an administrative user ID with the same ID. The administrative user ID DESK2 has a password of pass2dsk with client owner authority to the DESK2 node. When the PASSWORDACCESS=GENERATE option is used by the client to change the password, the administrative DESK2 ID can still access the client from a remote location.

Preventing automatic creation of an administrative user ID with client owner authority

You can prevent automatic creation of an administrative user ID with client owner authority by specifying USERID=NONE on the REGISTER NODE command.

To register DESK2 without creating an administrative user ID with client owner authority by default, issue the following example command:

```
register node desk2 pass2dsk userid=none
```

Registering a node and granting an existing administrative ID client owner authority

You can grant client owner authority to an existing administrative user ID.

To give client owner authority to the HELPADMIN user ID when registering the NEWCLIENT node, issue the following command:

```
register node newclient pass2new userid=helpadmin
```

This command results in the NEWCLIENT node being registered with a password of pass2new, and also grants HELPADMIN client owner authority. This command would not create an administrator ID. The HELPADMIN client user ID is now able to access the NEWCLIENT node from a remote location.

Example: setting up help desk access to client computers in a specific policy domain

The example is for setting up help desk access for user HELP1 to the client nodes in the FINANCE domain.

You are also granting HELP1 client access authority to the FINANCE domain without having to grant system or policy privileges.

The client nodes have been previously set up as follows:

- Installed and configured. The URL and port numbers were specified during the REGISTER NODE process.

- Assigned to the FINANCE policy domain.
- Started the Client Acceptor service.
- Specified passwordaccess generate option in their client option files.

The help desk person, using HELP1 user ID, has a Web browser with Java Runtime Environment (JRE) 1.3.1.

1. Register an administrative user ID of HELP1.

```
register admin help1 05x23 contact="M. Smith, Help Desk x0001"
```

2. Grant the HELP1 administrative user ID client access authority to all clients in the FINANCE domain. With client access authority, HELP1 can perform backup and restore operations for clients in the FINANCE domain. Client nodes in the FINANCE domain are Dave, Sara, and Joe.

```
grant authority help1 class=node authority=access domains=finance
```

The following output is generated by this command:

```
ANR2126I GRANT AUTHORITY: Administrator HELP1 was granted ACCESS authority for client
DAVE.
ANR2126I GRANT AUTHORITY: Administrator HELP1 was granted ACCESS authority for client
JOE.
ANR2126I GRANT AUTHORITY: Administrator HELP1 was granted ACCESS authority for client
SARA.
```

3. The help desk person, HELP1, opens the Web browser and specifies the URL and port number for client computer Sara:

```
http://sara.computer.name:1581
```

A Java applet is started, and the client hub window is displayed in the main window of the Web browser. When HELP1 accesses the backup function from the client hub, the IBM Tivoli Storage Manager login screen is displayed in a separate Java applet window. HELP1 authenticates with the administrative user ID and password. HELP1 can perform a backup for Sara.

For information about what functions are not supported on the Web backup-archive client, refer to *Backup-Archive Clients Installation and User's Guide*.

Managing file spaces

A file space name identifies a group of files that are stored as a logical unit in server storage. Administrators manage file spaces in which IBM Tivoli Storage Manager stores each client node's data.

Administrators can perform the following activities when managing file spaces:

Task	Required Privilege Class
Determine when existing file spaces are renamed to allow for the creation of new Unicode-enabled file spaces	System, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node is assigned.
Displaying information about file spaces	Any administrator

Task	Required Privilege Class
Move selected file spaces for a single node, as well as move a node's data located in a sequential access storage pool	System, unrestricted storage, 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.
Deleting file spaces	System or unrestricted policy
Deleting file spaces assigned to specific policy domains	System, unrestricted policy, or restricted policy for those domains

Related reference

"Defining client nodes and file spaces"

Defining client nodes and file spaces

Each client is given a node name when it is registered with the server. The server views its registered nodes as clients that require services and resources from the server.

Typically, a node is equivalent to a computer as in the case of a backup-archive client installed on a user's computer for file system backups. However, multiple nodes can exist on a single computer as in the case of a SQL server containing both an application client for SQL database and transaction log backups, and a backup-archive client for file system backups.

Typically, each client file system is represented on the server as a unique file space that belongs to each client node. Therefore, the number of file spaces a node has depends on the number of file systems on the client computer. For example, a Windows desktop system may have multiple drives (file systems), such as C: and D:. In this case, the client's node has two file spaces on the server; one for the C: drive and a second for the D: drive. The file spaces can grow as a client stores more data on the server. The file spaces decrease as backup and archive file versions expire and the server reclaims the space.

IBM Tivoli Storage Manager does not allow an administrator to delete a node unless the node's file spaces have been deleted.

Mapping file spaces for clients

For client nodes running on Windows, file spaces map to logical partitions and shares. Each file space is named with the UNC name of the respective client partition or share.

For client nodes running on NetWare, file spaces map to NetWare volumes. Each file space is named with the corresponding NetWare volume name.

For clients running on Macintosh, file spaces map to Macintosh volumes. Each file space is named with the corresponding Macintosh volume name.

For clients running on UNIX or Linux, a file space name maps to a file space in storage that has the same name as the file system or virtual mount point from which the files originated. The VIRTUALMOUNTPOINT option allows users to define a virtual mount point for a file system to back up or archive files beginning with a specific directory or subdirectory. For information on the

the path no longer exist in the Tivoli Storage Manager server database. You will not be able to obtain a token ID by issuing QUERY BACKUPSETCONTENTS. To view the fully-qualified name, you can load the backup set table of contents from the client.

For more information on fully-qualified object names and using the DISPLAY OBJNAME command, see the *Administrator's Reference*.

Supporting Unicode-enabled clients

Unicode is a universal character encoding standard that supports the interchange, processing, and display of text that is written in any of the languages of the modern world.

For Windows, Macintosh OS X, and NetWare systems with the Unicode-enabled client, the server supports storing file spaces with Unicode file space names, directory names, and file names in server storage. The file spaces in server storage that have Unicode names are called Unicode-enabled file spaces. Support for Unicode names enables a client to successfully process an IBM Tivoli Storage Manager operation, even when the file spaces contain directory names or files in multiple languages, or when the client uses a different code page than the server.

New clients storing data on the server for the first time require no special setup. If the client has the latest IBM Tivoli Storage Manager client software installed, the server automatically stores Unicode-enabled file spaces for that client.

However, if you have clients that already have data stored on the server and the clients install the Unicode-enabled IBM Tivoli Storage Manager client software, you need to plan for the migration to Unicode-enabled file spaces. To allow clients with existing data to begin to store data in Unicode-enabled file spaces, IBM Tivoli Storage Manager provides a function for automatic renaming of existing file spaces. The file data itself is not affected; only the file space name is changed. After the existing file space is renamed, the operation creates a new file space that is Unicode-enabled. The creation of the new Unicode-enabled file space for clients can greatly increase the amount of space required for storage pools and the amount of space required for the server database. It can also increase the amount of time required for a client to run a full incremental backup, because the first incremental backup after the creation of the Unicode-enabled file space is a full backup.

When clients with existing file spaces migrate to Unicode-enabled file spaces, you need to ensure that sufficient storage space for the server database and storage pools is available. You also need to allow for potentially longer backup windows for the complete backups.

Attention: After the server is at the latest level of software that includes support for Unicode-enabled file spaces, you can only go back to a previous level of the server by restoring an earlier version of IBM Tivoli Storage Manager and the database.

A Unicode-enabled IBM Tivoli Storage Manager client is currently available for Windows, Macintosh OS X, and NetWare operating systems. Data in a Unicode code page from any other source, including down-level clients and API clients, will not be identified or treated as Unicode-enabled.

It is strongly recommended that users of Windows, Macintosh, and NetWare operating systems migrate their non-Unicode file spaces to Unicode-enabled file spaces. For more information see *Backup-Archive Clients Installation and User's Guide*.

Related concepts

"Unicode-enabled clients and existing backup sets" on page 408

Related tasks

"Migrating clients to Unicode-enabled file spaces"

Related reference

"Reasons for migrating clients to Unicode-enabled file spaces"

"Querying Unicode-enabled file spaces" on page 408

Reasons for migrating clients to Unicode-enabled file spaces

Without IBM Tivoli Storage Manager support for storing Unicode-enabled file spaces, some clients experience backup failures when file spaces contain names of directories or files in multiple languages, or have names that cannot be converted to the server's code page.

When IBM Tivoli Storage Manager cannot convert the code page, the client may receive one or all of the following messages if they were using the command line: ANS1228E, ANS4042E, and ANS1803E. Clients that are using the GUI may see a "Path not found" message. If you have clients that are experiencing such backup failures, then you need to migrate the file spaces for these clients to ensure that these systems are completely protected with backups. If you have a large number of clients, set the priority for migrating the clients based on how critical each client's data is to your business.

Any new file spaces that are backed up from client systems with the Unicode-enabled IBM Tivoli Storage Manager client are automatically stored as Unicode-enabled file spaces in server storage.

Objects backed up or archived with a Unicode-enabled IBM Tivoli Storage Manager client in any supported language environment can be restored or retrieved with a Unicode-enabled client in the same or any other supported language environment. This means, for example, that files backed up by a Japanese Unicode-enabled client can be restored by a German Unicode-enabled client.

Important: Objects backed up or archived by a Unicode-enabled IBM Tivoli Storage Manager client cannot be restored or retrieved by a client that is not Unicode-enabled.

Related tasks

"Migrating clients to Unicode-enabled file spaces"

Migrating clients to Unicode-enabled file spaces

To allow clients with existing data to migrate to Unicode-enabled file spaces, IBM Tivoli Storage Manager provides an automatic rename function for file spaces.

When enabled, IBM Tivoli Storage Manager uses the rename function when it recognizes that a file space that is not Unicode-enabled in server storage matches the name of a file space on a client. The existing file space in server storage is renamed, so that the file space in the current operation is then treated as a new, Unicode-enabled file space. For example, if the operation is an incremental backup at the file space level, the entire file space is then backed up to the server as a Unicode-enabled file space.

The following example shows how this process works when automatic renaming is enabled from the server, for an existing client node that has file spaces in server storage.

1. The administrator updates a client node definition by issuing an UPDATE NODE command with the parameter, AUTOFSRENAME YES.
2. The client processes an incremental back up.
3. The IBM Tivoli Storage Manager processes the backup as follows:
 - a. Renames the existing file space (_OLD)
 - b. Creates a new Unicode-enabled file space
 - c. Processes the backup in the current operation to the new Unicode-enabled file space

Attention: If you force the file space renaming for all clients at the same time, backups can contend for network and storage resources, and storage pools can run out of storage space.

Related tasks

“Planning for Unicode versions of existing client file spaces” on page 404

“Examining issues when migrating to Unicode” on page 406

“Example of a migration process” on page 407

Related reference

“Defining options for automatically renaming file spaces”

“Defining the rules for automatically renaming file spaces” on page 404

Defining options for automatically renaming file spaces:

As an administrator, you can control whether the file spaces of any existing clients are renamed to force the creation of new Unicode-enabled file spaces. By default, no automatic renaming occurs.

To control the automatic renaming, use the parameter AUTOFSRENAME when you register or update a node. You can also allow clients to make the choice. Clients can use the client option AUTOFSRENAME.

Restriction: The setting for AUTOFSRENAME affects only clients that are Unicode-enabled.

You have the following options:

- Do not allow existing file spaces to be renamed, so that Unicode-enabled file spaces are not created (AUTOFSRENAME=NO, the default).

IBM Tivoli Storage Manager does not automatically rename client file spaces when the client system upgrades to the Unicode-enabled IBM Tivoli Storage Manager client. This setting can help an administrator control how many clients' file spaces can be renamed at one time. The administrator can determine how many Unicode-enabled clients exist by issuing the QUERY NODE FORMAT=DETAILED command. The output displays the client level.

- Automatically rename existing file spaces, forcing the creation of Unicode-enabled file spaces in place of the renamed file spaces (AUTOFSRENAME=YES).

IBM Tivoli Storage Manager automatically renames client file spaces in server storage when the client upgrades to the Unicode-enabled client and runs one of the following operations: archive, selective backup, full incremental backup, or partial incremental backup. IBM Tivoli Storage Manager automatically renames

the file spaces that are specified in the current operation and creates new, Unicode-enabled file spaces where files and directories are stored to complete the operation. Other file spaces that are not specified in the current operation are not affected by the rename. This means a client can have mixed file spaces.

Attention: If you force the file space renaming for all clients at the same time, client operations can contend for network and storage resources, and storage pools can run out of storage space.

- Allow clients to choose whether to rename files spaces, in effect choosing whether new Unicode-enabled file spaces are created (AUTOFSRENAME=CLIENT).

If you use this value for a client node, the client can set its AUTOFSRENAME option in its options file. The client option determines whether file spaces are renamed (YES or NO), or whether the user is prompted for renaming at the time of an IBM Tivoli Storage Manager operation (PROMPT).

The default value for the client option is PROMPT. When the option is set for prompting, the client is presented with a choice about renaming file spaces. When a client that has existing file spaces on server storage upgrades to the Unicode-enabled client, and the client runs an IBM Tivoli Storage Manager operation with the server, the user is asked to choose whether to rename the file spaces that are involved in the current operation.

The client is prompted only once about renaming a particular file space.

If the client does not choose to rename the file space, the administrator can later rename the file space so that a new Unicode-enabled file space is created the next time the client processes an archive, selective backup, full incremental backup, or partial incremental backup.

Attention: There is no prompt for operations that run with the client scheduler. If the client is running the scheduler and the client AUTOFSRENAME option is set to PROMPT, there is no prompt and the file space is not renamed. This allows a client session to run unattended. The prompt appears during the next interactive session on the client.

The following table summarizes what occurs with different parameter and option settings.

Table 42. The effects of the AUTOFSRENAME option settings

Parameter on the server (for each client)	Option on the client	Result for file spaces	Is the file space renamed?
Yes	Yes, No, Prompt	Renamed	Yes
No	Yes, No, Prompt	Not renamed	No
Client	Yes	Renamed	Yes
	No	Not renamed	Yes
	Prompt	Command-line or GUI: The user receives a one-time only prompt about renaming Client Scheduler: Not renamed (prompt appears during the next command-line or GUI session)	Depends on the response from the user (yes or no) No

Related reference

“Defining the rules for automatically renaming file spaces” on page 404

Defining the rules for automatically renaming file spaces:

With its automatic renaming function, IBM Tivoli Storage Manager renames a file space by adding the suffix `_OLD`.

For example:

Original file space name	New file space name
<code>\\maria\c\$</code>	<code>\\maria\c\$_OLD</code>

If the new name would conflict with the name of another file space, a number is added to the suffix. For example:

Original file space name	New file space name	Other existing file spaces
<code>\\maria\c\$</code>	<code>\\maria\c\$_OLD</code>	<code>\\maria\c\$_OLD1</code> <code>\\maria\c\$_OLD2</code>

If the new name for the file space exceeds the limit of 64 characters, the file space name is truncated on the right before the suffix `_OLD` is added.

Planning for Unicode versions of existing client file spaces:

Several factors must be considered before you plan for Unicode versions of existing client file spaces.

Consider the following items when planning:

- After clients with existing file spaces start to create Unicode-enabled file spaces, they will still need to have access to the renamed file spaces that are not Unicode-enabled for some period of time.
- Your storage pool and database space requirements can double if you allow all clients to create Unicode-enabled file spaces in addition to their existing file spaces that are not Unicode-enabled.
- Because the initial backups after migration are complete backups, it can also greatly increase the time required to finish backup operations.

To minimize problems, you need to plan the storage of Unicode-enabled file spaces for clients that already have existing file spaces in server storage.

1. Determine which clients need to migrate.

Clients that have had problems with backing up files because their file spaces contain names of directories or files that cannot be converted to the server's code page should have the highest priority. Balance that with clients that are most critical to your operations. If you have a large number of clients that need to become Unicode-enabled, you can control the migration of the clients.

Change the rename option for a few clients at a time to keep control of storage space usage and processing time. Also consider staging migration for clients that have a large amount of data backed up.

2. Allow for increased backup time and network resource usage when the Unicode-enabled file spaces are first created in server storage.

Based on the number of clients and the amount of data those clients have, consider whether you need to stage the migration. Staging the migration means setting the `AUTOFSRENAME` parameter to `YES` or `CLIENT` for only a small number of clients every day.

Note: If you set the `AUTOFSRENAME` parameter to `CLIENT`, be sure to have the clients (that run the client scheduler) set their option to `AUTOFSRENAME YES`. This ensures the file spaces are renamed.

3. Check the current storage usage for the clients that need to become Unicode-enabled.

You can use the `QUERY OCCUPANCY` command to display information on how much space each client is currently using. Initially, clients will need only the amount of space used by active files. Therefore, you need to estimate how much of the current space is used by copies (different versions of the same file). Migration will result in a complete backup at the next incremental backup, so clients will need space for that backup, plus for any other extra versions that they will keep. Therefore, the amount of storage required also depends on policy (see the next step). Your IBM Tivoli Storage Manager policy specifies how files are backed up, archived, migrated from client node storage, and managed in server storage.

4. Understand how your IBM Tivoli Storage Manager policies affect the storage that will be needed.

If your policies expire files based only on the number of versions (Versions Data Exists), storage space required for each client will eventually double, until you delete the old file spaces.

If your policies expire files based only on age (Retain Extra Versions), storage space required for each client will increase initially, but will not double.

If your policies use both the number of versions and their age, each client will need less than double their current usage.

5. Estimate the effect on the database size.

The database size depends on the number of files in server storage, as well as the number of versions of those files. As Unicode-enabled file spaces are backed up, the original file spaces that were renamed remain. Therefore, the server requires additional space in the database to store information about the increased number of file spaces and files.

6. Arrange for the additional storage pool space, including space in copy storage pools and active-data pools, based on your estimate from step 3 and 4.
7. Check the server database space that is available and compare with your estimate from step 5.
8. Ensure that you have a full database backup before you proceed with migration of Unicode-enabled file spaces.
9. Consider how you will manage the renamed file spaces as they age. The administrator can delete them, or the clients can be allowed to delete their own file spaces.

Related tasks

“Estimating database space requirements” on page 599

“Backing up the server database” on page 767

Examining issues when migrating to Unicode:

When you migrate to Unicode, there are several issues that you must consider.

The server manages a Unicode-enabled client and its file spaces as follows:

- When a client upgrades to a Unicode-enabled client and logs in to the server, the server identifies the client as Unicode-enabled.

Remember: That same client (same node name) cannot log in to the server with a previous version of IBM Tivoli Storage Manager or a client that is not Unicode-enabled.

- The original file space that was renamed (_OLD) remains with both its active and inactive file versions that the client can restore if needed. The original file space will no longer be updated. The server will not mark existing active files inactive when the same files are backed up in the corresponding Unicode-enabled file space.

Important: Before the Unicode-enabled client is installed, the client can back up files in a code page other than the current locale, but cannot restore those files. After the Unicode-enabled client is installed, if the same client continues to use file spaces that are not Unicode-enabled, the client skips files that are not in the same code page as the current locale during a backup. Because the files are skipped, they appear to have been deleted from the client. Active versions of the files in server storage are made inactive on the server. When a client in this situation is updated to a Unicode-enabled client, you should migrate the file spaces for that client to Unicode-enabled file spaces.

- The server does not allow a Unicode-enabled file space to be sent to a client that is not Unicode-enabled during a restore or retrieve process.
- Clients should be aware that they will not see all their data on the Unicode-enabled file space until a full incremental backup has been processed.

When a client performs a selective backup of a file or directory and the original file space is renamed, the new Unicode-enabled file space will contain only the file or directory specified for that backup operation. All other directories and files are backed up on the next full incremental backup.

If a client needs to restore a file before the next full incremental backup, the client can perform a restore from the renamed file space instead of the new Unicode-enabled file space. For example:

- Sue had been backing up her file space, \\sue-node\d\$.
- Sue upgrades the IBM Tivoli Storage Manager client on her system to the Unicode-enabled IBM Tivoli Storage Manager client.
- Sue performs a selective backup of the HILITE.TXT file.
- The automatic file space renaming function is in effect and IBM Tivoli Storage Manager renames \\sue-node\d\$ to \\sue-node\d\$_OLD. IBM Tivoli Storage Manager then creates a new Unicode-enabled file space on the server with the name \\sue-node\d\$. This new Unicode-enabled file space contains only the HILITE.TXT file.
- All other directories and files in Sue's file system will be backed up on the next full incremental backup. If Sue needs to restore a file before the next full incremental backup, she can restore the file from the \\sue-node\d\$_OLD file space.

Refer to the *Backup-Archive Clients Installation and User's Guide* for more information.

Example of a migration process:

The example of a migration process includes one possible sequence for migrating clients.

Assumptions for this scenario are:

- The IBM Tivoli Storage Manager server database has been backed up.
- The latest server software has been installed. This installation has also performed an upgrade to the server database.
- Clients have installed the latest software.
- A few clients are file servers. Most clients are workstations used by individuals.
- Clients generally run scheduled incremental backups every night.

The following migration process is possible to perform:

1. Have all clients install the Unicode-enabled IBM Tivoli Storage Manager client software.
2. Migrate the file servers first. For clients that are file servers, update the **AUTOFSRENAME** parameter to enable automatic renaming for the file spaces. For example, if the client node names for all file servers begin with FILE, issue the following command:

```
update node file* autofsrename=yes
```

This forces the file spaces to be renamed at the time of the next backup or archive operation on the file servers. If the file servers are large, consider changing the renaming parameter for one file server each day.

3. Allow backup and archive schedules to run as usual. Monitor the results.
 - a. Check for the renamed file spaces for the file server clients. Renamed file spaces have the suffix **_OLD** or **_OLDn**, where n is a number.
 - b. Check the capacity of the storage pools. Add tape or disk volumes to storage pools as needed.
 - c. Check database usage statistics to ensure you have enough space.

Note: If you are using the client acceptor to start the scheduler, you must first modify the default scheduling mode.

4. Migrate the workstation clients. For example, migrate all clients with names that start with the letter a.

```
update node a* autofsrename=yes
```

5. Allow backup and archive schedules to run as usual that night. Monitor the results.
6. After sufficient time passes, consider deleting the old, renamed file spaces.

Related tasks

“Modifying the default scheduling mode” on page 537

Related reference

“Managing the renamed file spaces” on page 408

“Defining the rules for automatically renaming file spaces” on page 404

Managing the renamed file spaces:

The file spaces that were automatically renamed (_OLD) to allow the creation of Unicode-enabled file spaces continue to exist on the server. Users can still access the file versions in these file spaces.

Because a renamed file space is not backed up again with its new name, the files that are active (the most recent backup version) in the renamed file space remain active and never expire. The inactive files in the file space expire according to the policy settings for how long versions are retained. To determine how long the files are retained, check the values for the parameters, **Retain Extra Versions** and **Retain Only Versions**, in the backup copy group of the management class to which the files are bound.

When users no longer have a need for their old, renamed file spaces, you can delete them. If possible, wait for the longest retention time for the only version (**Retain Only Version**) that any management class allows. If your system has storage constraints, you may need to delete these file spaces before that.

Querying Unicode-enabled file spaces

You can determine which file spaces are Unicode-enabled by querying all of the file spaces.

Issue the following command:

```
query fileSpace
```

Node Name	Filespace Name	FSID	Platform	Filespace Type	Is Filespace Unicode?	Capacity (MB)	Pct Util
SUE	\\sue\c\$	1	WinNT	NTFS	Yes	2,502.3	75.2
SUE	\\sue\d\$	2	WinNT	NTFS	Yes	6,173.4	59.6
JOE	\\joe\c\$	1	WinNT	NTFS	No	12,299.7	31.7

To query a specific Unicode-enabled file space, it may be more convenient to use the file space identifier (FSID) than the file space name. File space names for Unicode-enabled file spaces may not be readable when displayed in the server's code page. Attempting to enter the name of a Unicode-enabled file space may not work because it depends on the server's code page and conversion routines that attempt to convert from the server's code page to Unicode.

Related tasks

"Displaying information about file spaces" on page 409

Unicode-enabled clients and existing backup sets

A client can have a backup set that contains both file spaces that are Unicode-enabled and file spaces that are not Unicode-enabled. The client must have the same level of IBM Tivoli Storage Manager or higher to restore the data in the backup set.

For example, a Version 5.1.0 client backs up file spaces, and then upgrades to Version 5.2.0 with support for Unicode-enabled file spaces. That same client can still restore the non-Unicode file spaces from the backup set.

Unicode-enabled file spaces in a backup set can only be accessed by a Unicode-enabled client, and not by an earlier version of the client. The server allows only Unicode-enabled clients to restore data from Unicode-enabled file spaces.

Related reference

“Restoring backup sets from a backup-archive client” on page 507

Displaying information about file spaces

You can display file space information by identifying the client node name and file space name.

You can display file space information for the following reasons:

- To identify file spaces defined to each client node, so that you can delete each file space from the server before removing the client node from the server
- To identify file spaces that are Unicode-enabled and identify their file space ID (FSID)
- To monitor the space used on workstation's disks
- To monitor whether backups are completing successfully for the file space
- To determine the date and time of the last backup

Note: File space names are case-sensitive and must be entered exactly as known to the server.

To view information about file spaces defined for client node JOE, issue the following command:

```
query filesystem joe *
```

The following figure shows the output from this command.

When you display file space information in detailed format, the Filespace Name

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

field may display file space names as "...". This indicates to the administrator that a file space does exist but could not be converted to 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.

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 Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space name or file name may display with a combination of invalid characters or blank spaces.

Refer to the *Administrator's Reference* for details.

Moving data for a client node

You can move a client node's data in a sequential-access storage pool or move selected file spaces for a single node.

Related tasks

"Moving data belonging to a client node" on page 354

Deleting file spaces

You can delete a client node from a server, but first you must delete all of that client's data from server storage by deleting any file spaces that belong to the node.

Administrators may want to delete a file space in the following cases:

- Users are not authorized to delete backed-up or archived files in storage pools.
 - You want to remove a client node from the server.
 - You want to delete a specific user's files.
1. For users who are not authorized to delete backed-up or archived files in storage pools, as an example, client node PEASE no longer needs archived files in file space `/home/pease/dir2`. However, he does not have the authority to delete those files. To delete the files in `/home/pease/dir2`, issue the following command:

```
delete filespace pease /home/pease/dir2 type=archive
```

The authority to delete backed-up or archived files from server storage is set when a client node is registered.

2. You must delete a user's files from storage pools before you can remove a client node. For example, to delete all file spaces belonging to client node DEBBYG, issue the following command:

```
delete filespace debbyg * type=any
```

After you delete all of a client node's file spaces, you can delete the node with the REMOVE NODE command.

For client nodes that support multiple users, such as UNIX or Linux, a file owner name is associated with each file on the server. The owner name is the user ID of the operating system, such as the UNIX Linux user ID. When you delete a file space belonging to a specific owner, only files that have the specified owner name in the file space are deleted.

When a node has more than one file space and you issue a DELETE FILESPACE command for only one file space, a QUERY FILESPACE command for the node during the delete process shows no file spaces. When the delete process ends, you can view the remaining file spaces with the QUERY FILESPACE command. If data retention protection is enabled, the only files which will be deleted from the file space are those which have met the retention criterion. The file space will not be deleted if one or more files within the file space cannot be deleted.

Related concepts

"Accepting default closed registration or enabling open registration" on page 368

Related tasks

"Deleting client nodes" on page 389

Managing client option files

A client node connects with the server by using the information in a client options file (dsm.opt). This file, located in the client directory, contains client options that control processing and connections with the server.

The most important option is the network address of the server, but you can add many other client options at any time. Administrators can also control client options by creating client option sets on the server that are used in conjunction with client option files on client nodes.

Related tasks

“Creating client option sets on the server”

“Managing client option sets” on page 413

Related reference

“Connecting nodes with the server” on page 372

Creating client option sets on the server

An administrator can create a set of client options to be used by a client node that is at IBM Tivoli Storage Manager Version 3 or later. The client options specified in the set are used in conjunction with the client options dsm.opt file.

Client option sets allow the administrator to specify additional options that may not be included in the client's option file (dsm.opt). You can specify which clients use the option set with the REGISTER NODE or UPDATE NODE commands. The client can use these defined options during a backup, archive, restore, or retrieve process. See the *Backup-Archive Clients Installation and User's Guide* for detailed information about individual client options.

To create a client option set and have the clients use the option set, perform the following steps:

1. Create the client option set with the DEFINE CLOPTSET command.
2. Add client options to the option set with the DEFINE CLIENTOPT command.
3. Specify which clients should use the option set with the REGISTER NODE or UPDATE NODE command.

Related reference

“Connecting nodes with the server” on page 372

Creating a client option set

When you create a client option set, you define a name for the option set and can, optionally, provide a description of the option set.

To provide a description of the option set, issue the following example command:
define cloptset engbackup description='Backup options for eng. dept.'

Tip: The option set is empty when it is first defined.

Adding client options in an option set

You can add client options in a defined client option set.

Issue the following example command to add a client option (MAXCMDRETRIES) in the ENGBACKUP option set:

```
define clientopt engbackup maxcmdretries 5
```

For a list of valid client options that you can specify, refer to the *Administrator's Reference*.

The server automatically assigns sequence numbers to the specified options, or you can choose to specify the sequence number for order of processing. This is helpful if you have defined more than one of the same option as in the following example:

```
define clientopt engbackup inclexcl "include d:\admin"  
define clientopt engbackup inclexcl "include d:\payroll"
```

A sequence number of 0 is assigned to the option include d:\admin. A sequence number of 1 is assigned to the option include d:\payroll. If you want to specifically process one option before another, include the sequence parameter as follows:

```
define clientopt engbackup inclexcl "include d:\admin" seqnumber=2"  
define clientopt engbackup inclexcl "include d:\payroll" seqnumber=1"
```

The options are processed starting with the highest sequence number.

Any include-exclude statements in the server client option set have priority over the include-exclude statements in the local client options file. The server include-exclude statements are always enforced and placed at the bottom of the include-exclude list and evaluated before the client include-exclude statements. If the server option set has several include-exclude statements, the statements are processed starting with the highest sequence number. The client can use the QUERY INCLEXCL command to view the include-exclude statements in the order they are processed. QUERY INCLEXCL also displays the source of each include-exclude statement. For more information on the processing of the include-exclude statements see the *Backup-Archive Clients Installation and User's Guide*.

The **FORCE** parameter allows an administrator to specify whether the server forces the client to use an option value. This parameter has no affect on additive options such as INCLEXCL and DOMAIN. The default value is NO. If FORCE=YES, the server forces the client to use the value, and the client cannot override the value. The following example shows how you can prevent a client from using subfile backup:

```
define clientopt engbackup subfilebackup no force=yes
```

Related reference

"The include-exclude list" on page 448

Registering client nodes and assigning them to an option set

You can register or update a client node and specify an option set for the client to use.

Issue the following command to register or update a client node:

```
register node mike pass2eng cloptset=engbackup
```

The client node MIKE is registered with the password pass2eng. When the client node MIKE performs a scheduling operation, his schedule log entries are kept for five days.

Managing client option sets

Administrators can perform several activities to manage client option sets.

Perform the following steps:

1. Update the sequence number for a client option.

You can update the sequence number for a client option to change its processing order. This is helpful if you have more than one of the same option, for example several INCLUDE options.

The following command shows you how to change the sequence number for the DATEFORMAT option from 0 to 9:

```
update clientopt engbackup dateformat 0 9
```

2. Delete an option from a client option set.

You can remove an option that is defined in a client option set.

The following example shows you how to remove the SCHEDMODE polling option from the financeschd option set:

```
delete clientopt financeschd schedmode
```

3. Copy a client option set. You can copy an existing client option to another option set.

The following example shows you how to copy the engbackup option set to financeschd option set:

```
copy cloptset engbackup financeschd
```

4. Request information about a client option set. To display information about the contents of a client option set, issue the following command:

```
query cloptset financeschd
```

5. Update the description for a client option set. You can update the description for a client option set. The following example shows you how to update the description for the engbackup option set:

```
update cloptset engbackup description='Scheduling information'
```

6. Delete a client option set. When you delete a client option set, client node references to the option set are null. The clients continue to use their existing client options file. The following example shows you how to delete the engbackup client option set:

```
delete cloptset engbackup
```

Managing IBM Tivoli Storage Manager sessions

Each time an administrator or client node connects with the server, an administrative or client session is established. IBM Tivoli Storage Manager tracks its sessions in the server database.

Backup-archive clients are eligible for client restartable restore sessions; however, application clients are not.

Tivoli Storage Manager can hold a client restore session in DSMC loop mode until one of these conditions is met:

- The device class MOUNTRETENTION limit is satisfied.
- The client IDLETIMEOUT period is satisfied.
- The loop session ends.

Administrators can perform the following activities when managing IBM Tivoli Storage Manager sessions:

Task	Required Privilege Class
Displaying information about client sessions	Any administrator
Canceling a client session	System or operator
Disabling or enabling a client session	System or operator
Freeing links for client connections	Administrator with root authority

Related concepts

“Managing client restartable restore sessions” on page 417

Displaying information about IBM Tivoli Storage Manager sessions

Each client session is assigned a unique session number.

To display information about client sessions, issue the following command:

```
query session
```

Figure 63 shows a sample client session report.

You can determine the state of the server by examining the session state and wait

Sess Number	Comm. Method	Sess State	Wait Time	Bytes Sent	Bytes Recvd	Sess Type	Platform	Client Name
471	Tcp/Ip	IdleW	36 S	592	186	Node	WinNT	JOEUSER
472	Tcp/Ip	RecvW	0 S	730	638	Node	WinNT	STATION1

Figure 63. Information about client sessions

time to determine how long (in seconds, minutes, or hours) the session has been in the current state.

Server session states

The server session state can be Start, Run, End, RecvW, SendW, MediaW, or IdleW.

See the following definitions for the server session states:

Start Connecting with a client session.

Run Executing a client request.

End Ending a client session.

RecvW

Waiting to receive an expected message from the client while a database transaction is in progress. A session in this state is subject to the COMMTIMEOUT limit.

SendW

Waiting for acknowledgment that the client has received a message sent by the server.

MediaW

Waiting for removable media to become available.

Aggregation can cause multiple media waits within a transaction and is indicated by one client message.

Important: If `QUERY SESSION FORMAT=DETAILED` is specified, the Media Access Status field displays the type of media wait state.

IdleW

Waiting for communication from the client, and a database transaction is NOT in progress. A session in this state is subject to the limit as specified in the server options file.

If a client does not initiate communication within the specified time limit set by the IDLETIMEOUT option in the server options file, then the server cancels the client session.

For example, if the IDLETIMEOUT option is set to 30 minutes, and a user does not initiate any operations within those 30 minutes, then the server cancels the client session. The client session is automatically reconnected to the server when it starts to send data again.

Related tasks

“Reclaiming space in sequential-access storage pools” on page 319

Canceling an IBM Tivoli Storage Manager session

You can cancel a client session with the `CANCEL SESSION` command and the associated session number. Canceling sessions may be necessary when a user's computer is not responding or as a prerequisite to halting the server.

Administrators can display a session number with the `QUERY SESSION` command.

Users and administrators whose sessions have been canceled must reissue their last command to access the server again.

If an operation, such as a backup or an archive process, is interrupted when you cancel the session, the server rolls back the results of the current transaction. That is, any changes made by the operation that are not yet committed to the database are undone. If necessary, the cancellation process may be delayed.

If the session is in the Run state when it is canceled, the cancel process does not take place until the session enters the SendW, RecvW, or IdleW state.

If the session you cancel is currently waiting for a media mount, the mount request is automatically canceled. If a volume associated with the client session is currently being mounted by an automated library, the cancel may not take effect until the mount is complete.

For example, to cancel a session for client MARIE:

1. Query client sessions to determine the session number. The example report displays MARIE's session number 6.
2. Cancel node MARIE's session by entering:

```
cancel session 6
```

If you want to cancel all backup and archive sessions, enter:

```
cancel session all
```

Related tasks

“Displaying information about IBM Tivoli Storage Manager sessions” on page 414

Related reference

“Server session states” on page 415

When a client session is automatically canceled

Client sessions can be automatically canceled.

The reasons are based on the settings of the following server options:

COMMTIMEOUT

Specifies how many seconds the server waits for an expected client message during a transaction that causes a database update. If the length of time exceeds this time-out, the server rolls back the transaction that was in progress and ends the client session. The amount of time it takes for a client to respond depends on the speed and processor load for the client and the network load.

IDLETIMEOUT

Specifies how many minutes the server waits for a client to initiate communication. If the client does not initiate communication with the server within the time specified, the server ends the client session. For example, the server prompts the client for a scheduled backup operation but the client node is not started. Another example can be that the client program is idle while waiting for the user to choose an action to perform (for example, backup archive, restore, or retrieve files). If a user starts the client session and does not choose an action to perform, the session will time out. The client program automatically reconnects to the server when the user chooses an action that requires server processing. A large number of idle sessions can inadvertently prevent other users from connecting to the server.

THROUGHPUTDATATHRESHOLD

Specifies a throughput threshold, in kilobytes per second, a client session must achieve to prevent being cancelled after the time threshold is reached. 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. This option is used in conjunction with the THROUGHPUTTIMETHRESHOLD server option.

THROUGHPUTTIMETHRESHOLD

Specifies the time threshold, in minutes, for a session after which it may be canceled for low throughput. The server ends a client session when it has been active for more minutes than specified and the data transfer rate is less than the amount specified in the THROUGHPUTDATATHRESHOLD server option.

Refer to the *Administrator's Reference* for more information.

Disabling or enabling access to the server

You can prevent clients from establishing sessions with the server by issuing the DISABLE SESSIONS command. This command does not cancel sessions currently in progress or system processes such as migration and reclamation.

Task	Required Privilege Class
Disabling and enabling client node access to the server	System or operator
Displaying server status	Any administrator

To disable client node access to the server, issue the following example command:
disable sessions

You continue to access the server and current client activities complete unless a user logs off or an administrator cancels a client session. After the client sessions have been disabled, you can enable client sessions and resume normal operations by issuing the following command:

enable sessions

You can issue the QUERY STATUS command to determine if the server is enabled or disabled.

Related tasks

"Locking and unlocking client nodes" on page 388

Managing client restartable restore sessions

Some large restore operations may invoke a special type of restore operation called client restartable restore sessions. These special sessions allow users to restart the restore session from where it left off if the session was interrupted.

IBM Tivoli Storage Manager identifies client restartable restore sessions by displaying message ANS1247I on the client computer when the session starts. These restore sessions can be restarted as long as the restore interval has not expired.

After a restore operation that comes directly from tape, the Tivoli Storage Manager server does not release the mount point to IDLE status from INUSE status. The server does not close the volume to allow additional restore requests to be made to that volume. However, if there is a request to perform a backup in the same session, and that mount point is the only one available, then the backup operation will stop and the server will issue message ANS1114I. You can avoid this by

closing the DSMC restore session after the restore operation completes. This releases the mount point for subsequent sessions.

When a restartable restore session is saved in the server database the file space is locked in server storage. The following rules are in effect during the file space lock:

- Files residing on sequential volumes associated with the file space cannot be moved.
- Files associated with the restore cannot be backed up. However, files not associated with the restartable restore session that are in the same file space are eligible for backup. For example, if you are restoring all files in directory A, you can still backup files in directory B from the same file space.

The `RESTOREINTERVAL` server option allows administrators to specify how long client restartable restore sessions are saved in the server database. Consider scheduled backup operations when setting this option. For more information, refer to the `RESTOREINTERVAL` server option in the *Administrator's Reference*.

Administrators can perform the following activities when managing client restartable restore sessions:

Task	Required Privilege Class
Displaying information about client restartable restore sessions	Any administrator
Canceling client restartable restore sessions	System or operator
Interrupting client restartable restore sessions	System or operator

Displaying information about a client restartable restore session

You can display information about client restartable restore sessions with the `QUERY RESTORE` command.

To determine which client nodes have eligible restartable restore sessions, issue the following example command:

```
query restore
```

Restartable restore sessions have a negative session number.

Canceling a client restartable restore session

When a client restore session is in a restartable state, the file space is locked in server storage and no files can be moved from sequential volumes. This prevents the data from being migrated, moved, reclaimed, or backed up by another operation.

These sessions will automatically expire when the specified restore interval has passed.

An administrator can cancel a restartable restore session that is in an active or restartable state. If the restore session is active, any outstanding mount requests related to the active session are automatically canceled. When a restartable restore session is canceled with the `CANCEL RESTORE` command, it cannot be restarted from the point of interruption. A restartable restore session always has a negative session number.

To cancel a restartable restore session, you must specify the session number. For example:

```
cancel restore -1
```

Interrupting an active client restartable restore session

An administrator can interrupt an active restartable restore session by canceling the session, but the session can not then be restarted.

A session that ends prematurely through an error or ends by an administrator using CTRL-C on the Tivoli Storage Manager client might be restartable. Issue the QUERY RESTORE command to show the restartable restore sessions. A session with a negative number can be restarted.

Issue the following command to cancel a session:

```
cancel session -2
```

Session -2 cannot be restarted after you issue this command.

Chapter 13. Managing IBM Tivoli Storage Manager security

Administrators can perform specific activities to manage IBM Tivoli Storage Manager security.

Related concepts

“Managing IBM Tivoli Storage Manager administrators” on page 428

“Managing levels of administrative authority” on page 430

“Managing passwords and login procedures” on page 432

“Securing the server console” on page 425

“Securing sensitive client data” on page 499

Related reference

“Managing access to the server and clients” on page 427

“Administrative authority and privilege classes” on page 425

Securing client and server communications

Communications between the Tivoli Storage Manager server and the backup-archive client, administrative command-line client (dsmadm), and client application programming interface are always password protected. However, you can add another level of data protection by using Secure Sockets Layer (SSL).

SSL is the standard technology for creating encrypted links between servers and clients. SSL provides a secure channel for servers and clients to communicate over open communications paths. With SSL, the identities of the parties are verified through the use of digital certificates.

Due to system performance concerns, use SSL only for sessions where it is needed. Consider adding additional processor resources on the Tivoli Storage Manager server to manage the increased requirements.

Remember: The SSL implementation described here is different from the Administration Center SSL, which is implemented in the Tivoli Integrated Portal. Both methods use the same SSL technology, but they have different implementations and purposes. See “Configuring SSL for the Tivoli Integrated Portal” on page 425.

Setting up SSL

The IBM Tivoli Storage Manager server and client installation procedures include the silent installation of the Global Security Kit (GSKit).

Perform the following steps to configure Tivoli Storage Manager servers and clients for Secure Sockets Layer (SSL):

1. Specify the TCP/IP port on which the server waits for SSL-enabled client communications. If the key database (`cert.kdb`) file does not exist and there is no password for it in the database, a database is created. After creating the database, the following actions occur:
 - a. The key database access password is generated and stored, with encryption, in the server database. Storing the encrypted password allows the server to open the key database and access the certificate information.

- b. A public certificate that can be used by the Tivoli Storage Manager client is extracted and put in the cert.arm file. An administrator must manually transfer the cert.arm file to the client computers. You must then add the cert.arm file to the local certificate database as a signed certificate. The administrator must ensure that the transfer method is secure. There must be no possibility to tamper with or replace the original server certificate.
- c. Tivoli Storage Manager generates a self-signed certificate and stores it in the key database in the server instance directory. The self-signed certificate is the default certificate.

Tip: This implementation of SSL requires only a server certification. There is no client certificate.

2. The client user must specify SSL YES in the client options file (dsm.opt) to enable SSL communications and update the value of the TCPSPORT option.
3. The client user imports the server public certificate (cert.arm) into the client local key database. For details, see the *Backup-Archive Clients Installation and User's Guide*.
4. If you want to use a different certificate, you must install the Certificate Authority's (CA) root certificate on all clients.

Important: You can change the key database password by issuing the SET SSLKEYRINGPW command. You need only change the key database password once. After that you can use the new password when working with SSL.

Related reference

"Specifying communication ports"

"Adding a certificate to the key database" on page 423

Specifying communication ports

The Tivoli Storage Manager server can be configured to listen on four TCP/IP ports: two for regular protocol and two for the Secure Sockets Layer (SSL) protocol.

For IPv4 or IPv6, the COMMETHOD server option must specify either TCPIP or V6TCPIP. The server options for SSL communications are SSLTCPSPORT and SSLTCPADMINPORT. The server can listen on separate ports for the following communications:

- Clients using regular protocol
- Administrators using regular protocol
- Clients using SSL protocol
- Administrators using SSL protocol

Use the TCPADMINPORT and SSLTCPADMINPORT options to separate administrative client traffic from regular client traffic on TCPSPORT and SSLTCPSPORT. If the TCPADMINPORT and SSLTCPADMINPORT options are not used, administrative traffic flows on client ports as well.

The following components support SSL:

- Command line client
- Administrative command line client
- Backup-archive client graphical user interface (GUI)
- Client API

If the ADMINONCLIENTPORT option is set to NO, SSL administrative client sessions require an SSLTCPADMINPORT option with a port number other than

that specified by the SSLTCPPORT option. The SSLTCPPORT and SSLTCPADMINPORT options do not affect the TCPSPORT or TCPADMINPORT options and their interaction with the ADMINCLIENTPORT option.

The client user decides which protocol to use and which port to specify in the dsmserv.opt file for the SSLTCPADMINPORT option. If the client requests SSL authentication, but the server is in non-SSL mode, the session fails.

Adding a certificate to the key database

To use Secure Sockets Layer (SSL), the certificate must be installed on the server, and for some root certificates, they must be installed on the client. Each server that enables SSL must obtain a unique certificate signed by a certificate authority (CA) or use a self-signed certificate.

You can use your own certificates or purchase certificates from a CA. Either can be installed and added to the key database. If you have not previously changed the key database password, the password might contain characters that cannot be reproduced. For this reason, you can change the password using the SET SSLKEYRINGPW command.

If the certificate is signed by a trusted CA, obtain the certificate, install it in the key database, and restart the server. Because the certificate is provided by a trusted authority, the certificate is accepted by Tivoli Storage Manager and communication between server and client can commence.

A trusted CA is one whose root certificate is noted as trusted in the key database. The GSKit utility includes the following trusted root certificates:

- Entrust.net Global Secure Server Certification Authority
- Entrust.net Global Client Certification Authority
- Entrust.net Client Certification Authority
- Entrust.net Certification Authority (2048)
- Entrust.net Secure Server Certification Authority
- VeriSign Class 3 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 1 Public Primary Certification Authority
- VeriSign Class 4 Public Primary Certification Authority - G2
- VeriSign Class 3 Public Primary Certification Authority - G2
- VeriSign Class 2 Public Primary Certification Authority - G2
- VeriSign Class 1 Public Primary Certification Authority - G2
- VeriSign Class 4 Public Primary Certification Authority - G3
- VeriSign Class 3 Public Primary Certification Authority - G3
- VeriSign Class 2 Public Primary Certification Authority - G3
- VeriSign Class 1 Public Primary Certification Authority - G3
- Thawte Personal Premium CA Thawte Personal Freemail CA
- Thawte Personal Basic CA Thawte Premium Server CA
- Thawte Server CA
- Rational Software Architect (RSA) Secure Server Certification Authority

Adding a CA-signed SSL certificate:

You can use a Secure Sockets Layer (SSL) certificate if the client trusts the certificate authority (CA). Trust is established when you add a signed certificate to the server key database and use a root certificate for the CA in the client key database.

Important: The Global Security Toolkit (GSKit) Version 7.0.4.27 is included in the IBM Tivoli Storage Manager server installation. The backup-archive client and server communicate with SSL through services provided by GSKit.

Complete the following steps to add a certificate to the key database using GSKit:

1. Obtain a signed, server key database certificate from your CA.
2. Issue the following example command to receive the signed certificate and make it the default for communicating with clients:

```
gsk7capicmd -cert -receive -db cert.kdb  
-pw password -file cert_signed.arm  
-default_cert yes
```

Tip: The server key database name is cert.kdb.

3. Restart the server.
4. Transfer the root certificate (ca.arm) to the client directory.
5. Issue the following example command to add the root certificate to the client key database:

```
gsk7capicmd -cert -add -db dsmcert.kdb  
-pw password -label "my CA"  
-file ca.arm -format ascii -trust enable
```

Tip: The client key database name is dsmcert.kdb.

6. Issue the following example command to verify that the client can successfully connect:

```
dsmc query session
```

Maintaining the certificate key database

It is crucial that you keep backup copies of the cert.kdb file. Secure Sockets Layer (SSL) does not start if you restore a Tivoli Storage Manager server, but do not restore its cert.kdb database file. If you have a backup copy, you can simply restore the cert.kdb file and restart the server.

If you do not have a backup copy of the cert.kdb file, perform the following steps:

1. Issue the DELETE KEYRING server command to delete the entry for it that is located in the Tivoli Storage Manager database.
2. Delete all remaining cert.* files.
3. Shut down the server.
4. Start the server. The server automatically creates a new cert.kdb file and a corresponding entry in the Tivoli Storage Manager database. If you do not issue the DELETE KEYRING command, the server attempts, on startup, to create the key database with the previous password.
5. Redistribute the new cert.arm file to all clients that are using SSL. Users must reinstall any 3rd-party certificates for the client to use.

Configuring SSL for the Tivoli Integrated Portal

The documentation for configuring Secure Sockets Layer for the Tivoli Integrated Portal is available within the Tivoli Integrated Portal Information Center (help).

After logging in to the Tivoli Integrated Portal, click **Security** and select **SSL certificate and key management**. All SSL-based information for the Tivoli Integrated Portal is located in the Administration Center. Additional information about SSL is available in the Tivoli Integrated Portal Information Center. Click **Help** in the menu to open the Information Center. To find the SSL information, click **Using the console** → **Application server topics** → **SSL certificate and key management**. You can also look for “SSL” in the Information Center search field.

Securing the server console

At installation, the server console is defined with a special user ID, which is named `SERVER_CONSOLE`. This name is reserved and cannot be used by another administrator.

An administrator with system privilege can revoke or grant new privileges to the `SERVER_CONSOLE` user ID. However, an administrator cannot update, lock, rename, or remove the `SERVER_CONSOLE` user ID. The `SERVER_CONSOLE` user ID does not have a password.

Therefore, you cannot use the user ID from an administrative client unless you set authentication to off.

Administrative authority and privilege classes

After administrators are registered, they can perform a limited set of tasks. By default, administrators can request command-line help and issue queries.

To perform other tasks, administrators must be granted authority by being assigned one or more administrative privilege classes. Privilege classes determine the authority level for an administrator. Figure 64 on page 426 illustrates the privilege classes. An administrator with system privilege class can perform any task with the server. Administrators with policy, storage, operator, or node privileges can perform subsets of tasks.

Important: Two server options give you additional control over the ability of administrators to perform tasks.

- `QUERYAUTH` allows you to select the privilege class that an administrator must have to issue `QUERY` and `SELECT` commands. By default, no privilege class is required. You can change the requirement to one of the privilege classes, including system.
- `REQSYSAUTHOUTFILE` allows you to specify that system authority is required for commands that cause the server to write to an external file (for example, `BACKUP DB`). By default, system authority is required for such commands.

See the *Administrator's Reference* for details on server options.

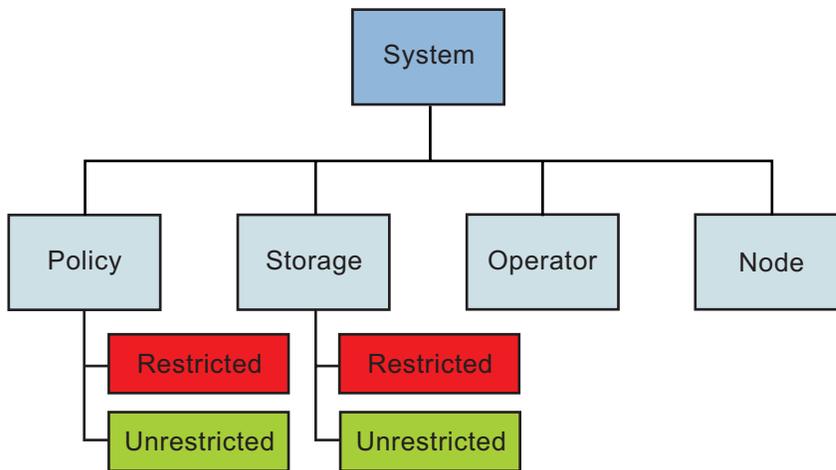


Figure 64. Administrative privilege classes

Table 43 summarizes the privilege classes, and gives examples of how to set privilege classes.

Table 43. Authority and privilege classes

Privilege Class	Capabilities
System grant authority rocko classes=system	Perform any administrative task with the server. <ul style="list-style-type: none"> • System-wide responsibilities • Manage the enterprise • Manage IBM Tivoli Storage Manager security
Unrestricted Policy grant authority smith classes=policy	Manage the backup and archive services for nodes assigned to any policy domain. <ul style="list-style-type: none"> • Manage nodes • Manage policy • Manage schedules
Restricted Policy grant authority jones domains=engpoldom	Same capabilities as unrestricted policy except authority is limited to specific policy domains.
Unrestricted Storage grant authority coyote classes=storage	Manage server storage, but not definition or deletion of storage pools. <ul style="list-style-type: none"> • Manage the database and recovery log • Manage IBM Tivoli Storage Manager devices • Manage IBM Tivoli Storage Manager storage
Restricted Storage grant authority holland stgpools=tape*	Manage server storage, but limited to specific storage pools. <ul style="list-style-type: none"> • Manage IBM Tivoli Storage Manager devices • Manage IBM Tivoli Storage Manager storage

Table 43. Authority and privilege classes (continued)

Privilege Class	Capabilities
Operator grant authority bill classes=operator	Control the immediate operation of the server and the availability of storage media. <ul style="list-style-type: none"> • Manage the IBM Tivoli Storage Manager server • Manage client sessions • Manage tape operations
Node \ grant authority help1 classes=node node=labclient	Access a Web backup-archive client to perform backup and restore operations.

Related concepts

“Overview of remote access to web backup-archive clients” on page 393

“Managing levels of administrative authority” on page 430

Managing access to the server and clients

An administrator can control access to the server and clients through a number of methods.

Table 44 describes the typical tasks for managing access to the server and clients.

Table 44. Managing access

Task	Details
Allow a new administrator to access the server	<ol style="list-style-type: none"> 1. “Registering administrators” on page 428 2. “Granting authority to administrators” on page 431
Modify authority for registered administrators	“Managing levels of administrative authority” on page 430
Give a user authority to access a client remotely	“Managing client access authority levels” on page 395
Give an administrator authority to create a backup set for a client node	“Generating client backup sets on the server” on page 503
Prevent administrators from accessing the server	“Locking and unlocking administrators from the server” on page 430
Prevent new sessions with the server, but allow current sessions to complete	“Disabling or enabling access to the server” on page 417
Prevent clients from accessing the server	“Locking and unlocking client nodes” on page 388
Change whether passwords are required to access IBM Tivoli Storage Manager	“Disabling the default password authentication” on page 433
Change requirements for passwords	<ul style="list-style-type: none"> • “Modifying the default password expiration period” on page 432 • “Setting a limit for invalid password attempts” on page 433 • “Setting a minimum length for a password” on page 433

Table 44. Managing access (continued)

Task	Details
Prevent clients from initiating sessions within a firewall	“Server-initiated sessions” on page 379
Tip: For information on connecting with IBM Tivoli Storage Manager across a firewall, refer to the <i>Installation Guide</i> .	

Managing IBM Tivoli Storage Manager administrators

The administrator is responsible for registering other administrators, granting levels of authority to administrators, renaming or removing administrators, and for locking and unlocking administrators from the server.

Task	Required Privilege Class
Registering an administrator	System
Granting administrative authority	System
Updating information about other administrators	System
Updating information about yourself	Any administrator
Displaying information about administrators	Any administrator
Renaming an administrator user ID	System
Removing administrators	System
Locking or unlocking administrators from the server	System

Registering administrators

An administrator registers other administrators using the REGISTER ADMIN command.

To register an administrator with a user ID of DAVEHIL, the password birds, and a password expiration period of 120 days, issue the following REGISTER ADMIN command:

```
register admin davehil birds passexp=120 contact='backup team'
```

Updating information about other administrators

An administrator can reset another administrator's password by issuing the UPDATE ADMINISTRATOR command.

For example, administrator DAVEHIL changes his password to ganymede by issuing the following command:

```
update admin davehil ganymede
```

Note: The SERVER_CONSOLE administrator's ID and contact information cannot be updated.

Renaming an administrator

You can rename an administrator ID when an employee wants to be identified by a new ID, or you want to assign an existing administrator ID to another person. You cannot rename an administrator ID to one that already exists on the system.

For example, if administrator HOLLAND leaves your organization, you can assign administrative privilege classes to another user by completing the following steps:

1. Assign HOLLAND's user ID to WAYNESMITH by issuing the RENAME ADMIN command:

```
rename admin holland waynesmith
```

By renaming the administrator's ID, you remove HOLLAND as a registered administrator from the server. In addition, you register WAYNESMITH as an administrator with the password, contact information, and administrative privilege classes previously assigned to HOLLAND.

2. Change the password to prevent the previous administrator from accessing the server by entering:

```
update admin waynesmith new_password contact="development"
```

Note: The administrator SERVER_CONSOLE cannot be renamed.

Related concepts

“Securing the server console” on page 425

Removing administrators

You can remove administrators from the server so that they no longer have access to administrative functions.

To remove registered administrator ID SMITH, issue the following example command:

```
remove admin smith
```

Important:

1. You cannot remove the last system administrator from the system.
2. You cannot remove the administrator SERVER_CONSOLE.

Related concepts

“Securing the server console” on page 425

Displaying information about administrators

Any administrator can query the server to display administrator information. You can restrict the query to all administrators authorized with a specific privilege class.

To query the system for a detailed report on administrator ID DAVEHIL, issue the following example QUERY ADMIN command:

```
query admin davehil format=detailed
```

Figure 65 on page 430 displays a detailed report.

```

Administrator Name: DAVEHIL
Last Access Date/Time: 2002.09.04 17.10.52
Days Since Last Access: <1
Password Set Date/Time: 2002.09.04 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: 05/09/2002 23:54:20
Registering Administrator: SERVER_CONSOLE
Managing profile:
Password Expiration Period: 90 Day (s)

```

Figure 65. A detailed administrator report

Locking and unlocking administrators from the server

You can lock out other administrators to temporarily prevent them from accessing IBM Tivoli Storage Manager by issuing the LOCK ADMIN command.

For example, administrator MARYSMITH takes a leave of absence from your business.

1. Lock her out by entering the following example command:

```
lock admin marysmith
```

2. When she returns, any system administrator can unlock her administrator ID by entering:

```
unlock admin marysmith
```

MARYSMITH can now access the server to complete administrative tasks. You cannot lock or unlock the SERVER_CONSOLE ID from the server.

Related concepts

“Securing the server console” on page 425

Managing levels of administrative authority

A privilege class is a level of authority granted to an administrator. The privilege class determines which administrative tasks the administrator can perform.

See the *Administrator's Reference* about the activities that administrators can perform with each privilege class.

You can perform the following activities to manage levels of authority:

Task	Required Privilege Class
Granting a level of authority to an administrator	System
Modifying the level of authority for an administrator	System

Related reference

“Administrative authority and privilege classes” on page 425

Granting authority to administrators

You can grant authority by issuing the GRANT AUTHORITY command.

To grant restricted policy privilege to administrator JONES for the domain ENGPOLDOM, issue the following example command:

```
grant authority jones domains=engpoldom
```

Extending authority for administrators

You can grant and extend authority by issuing the GRANT AUTHORITY command. If an ID already has some level of authority, granting additional authority adds to any existing privilege classes; it does not override those classes.

For example, JONES has restricted policy privilege for policy domain ENGPOLDOM.

1. To extend JONES' authority to policy domain MKTPOLDOM and add operator privilege, issue the following example command:

```
grant authority jones domains=mktpoldom classes=operator
```

2. As an additional example, assume that three tape storage pools exist: TAPEPOOL1, TAPEPOOL2, and TAPEPOOL3. To grant restricted storage privilege for these storage pools to administrator HOLLAND, you can issue the following command:

```
grant authority holland stgpools=tape*
```

3. HOLLAND is restricted to managing storage pools with names that begin with TAPE, if the storage pools existed when the authority was granted. HOLLAND is not authorized to manage any storage pools that are defined after authority has been granted. To add a new storage pool, TAPEPOOL4, to HOLLAND's authority, issue the following command:

```
grant authority holland stgpools=tapepool4
```

Reducing authority for administrators

You can revoke part of an administrator's authority by issuing the REVOKE AUTHORITY command.

For example, rather than revoking all of the privilege classes for administrator JONES, you want to revoke only the operator authority and the policy authority to policy domain MKTPOLDOM.

Issue the following command to revoke only the operator authority and the policy authority to policy domain MKTPOLDOM:

```
revoke authority jones classes=operator domains=mktpoldom
```

JONES still has policy privilege to the ENGPOLDOM policy domain.

Reducing privilege classes

You can reduce an administrator's authority simply by revoking one or more privilege classes and granting one or more other classes.

For example, administrator HOGAN has system authority. To reduce authority for HOGAN to the operator privilege class, perform the following steps:

1. Revoke the system privilege class by issuing the following command:
`revoke authority hogan classes=system`
2. Grant operator privilege class by issuing the following command:
`grant authority hogan classes=operator`

Revoking authority for administrators

You can revoke an administrator's authority by issuing the `REVOKE AUTHORITY` command. To revoke all administrative privilege classes, do not specify any privilege classes, policy domains, or storage pools.

For example, to revoke both the storage and operator privilege classes from administrator JONES, issue the following command:

```
revoke authority jones
```

Managing passwords and login procedures

By default, IBM Tivoli Storage Manager requires authorized administrators and nodes to identify themselves to the server with a password.

Administrators can perform the following activities to manage passwords and login procedures:

Task	Required Privilege Class
Modifying the default timeout period for the Administration Center	System
Modifying the default password expiration period	System
Setting the limit for invalid password attempts	System
Setting the minimum length for passwords	System
Disabling the default password authentication	System

Modifying the default password expiration period

By default, the server sets a password expiration of 90 days. The expiration period begins when an administrator or client node is first registered to the server. If a user password is not changed within this period, the server prompts the user to change the password the next time the user tries to access the server.

To set the password expiration period for selected administrators or client nodes, you must specify the administrator or node names with the **ADMIN** or **NODE** parameter with the `SET PASSEXP` command. If you set the expiration period only for selected users, you may set the expiration period from 0-9999 days. A value of 0 means that user's password never expires.

To set the expiration period of client node LARRY to 120 days, issue the following example command:

```
set passexp 120 node=larry
```

After you have explicitly set a password expiration for a node or administrator, it is not modified if you later set a password expiration for all users. You can use the RESET PASSEXP command to reset the password expiration period to the common expiration period. Use the QUERY STATUS command to display the common password expiration period, which at installation is set to 90 days.

Setting a limit for invalid password attempts

By default, IBM Tivoli Storage Manager does not check the number of times a user attempts to log in with an invalid password. You can set a limit on consecutive invalid password attempts for all client nodes. When the limit is exceeded, the server locks the node.

To set a system-wide limit of three consecutive invalid password attempts, issue the following example command:

```
set invalidpwlimit 3
```

The default value at installation is 0. A value of 0 means that invalid password attempts are not checked. You can set the value from 0 to 9999 attempts. If you initially set a limit of 4 and then change the limit to a lower number, some clients may fail verification during the next login attempt. After a client node has been locked, only a storage administrator with proper authority can unlock the node.

An administrator can also force a client to change their password on the next login by specifying the FORCEPWRESET=YES parameter on the UPDATE NODE or UPDATE ADMIN command. For more information, refer to *Administrator's Reference*.

Related tasks

“Locking and unlocking client nodes” on page 388

“Locking and unlocking administrators from the server” on page 430

Setting a minimum length for a password

By default, IBM Tivoli Storage Manager does not check the length of a password. The administrator can specify a minimum password length that is required for IBM Tivoli Storage Manager passwords.

To set the minimum password length to eight characters, issue the following example command:

```
set minpwlenght 8
```

The default value at installation is 0. A value of 0 means that the password length is not checked. You can set the length value from 0 to 64.

Disabling the default password authentication

By default, the server automatically sets password authentication on. With password authentication set to on, all users must enter a password when accessing the server.

To allow administrators and client nodes to access the server without entering a password, issue the following command:

```
set authentication off
```

Attention: Setting password authentication to “off” reduces data security.

Chapter 14. Implementing policies for client data

Policies are rules that you set at the IBM Tivoli Storage Manager server to help you manage client data. Policies control how and when client data is stored.

For example:

- How and when files are backed up and archived to server storage
- How space-managed files are migrated to server storage
- The number of copies of a file and the length of time copies are kept in server storage

IBM Tivoli Storage Manager provides a standard policy that sets rules to provide a basic amount of protection for data on workstations. If this standard policy meets your needs, you can begin using Tivoli Storage Manager immediately.

The server process of expiration is one way that the server enforces policies that you define. Expiration processing determines when files are no longer needed, that is, when the files are expired. For example, if you have a policy that requires only four copies of a file be kept, the fifth and oldest copy is expired. During expiration processing, the server removes entries for expired files from the database, effectively deleting the files from server storage.

You may need more flexibility in your policies than the standard policy provides. To accommodate individual user's needs, you may fine tune the STANDARD policy, or create your own policies. Some types of clients or situations require special policy. For example, you may want to enable clients to restore backed-up files to a specific point-in-time.

Policy can be distributed from a configuration manager to managed servers.

You can also perform Tivoli Storage Manager tasks from the Administration Center.

Related concepts

- “More on management classes” on page 446
- “How client migration works with backup and archive” on page 456
- “Protection and expiration of archive data” on page 474
- “Distributing policy using enterprise configuration” on page 489

Related tasks

- “Client operations controlled by policy” on page 440
- “Getting users started” on page 438
- “Changing policy” on page 439
- “Assigning client nodes to a policy domain” on page 472
- “Policy configuration scenarios” on page 482
- “Configuring policy for direct-to-tape backups” on page 482

Related reference

- “Basic policy planning”
- “Reviewing the standard policy” on page 437
- “File expiration and expiration processing” on page 439
- “The parts of a policy” on page 443
- “How Tivoli Storage Manager selects files for policy operations” on page 451
- “Creating your own policies” on page 456
- “Defining and updating a policy domain” on page 458
- “Defining and updating a management class” on page 461
- “Validating and activating a policy set” on page 470
- “Running expiration processing to delete expired files” on page 472
- “Configuring policy for Tivoli Storage Manager application clients” on page 483
- “Policy for logical volume backups” on page 483
- “Configuring policy for NDMP operations” on page 485
- “Configuring policy for LAN-free data movement” on page 486
- “Policy for Tivoli Storage Manager servers as clients” on page 488
- “Setting policy to enable point-in-time restore for clients” on page 488
- “Querying policy” on page 489
- “Deleting policy” on page 492

Basic policy planning

Start out simply to plan your policy. You may be able to use the default policy that comes with the server.

Ask the following questions:

- How many backup versions do clients need?
- How long do clients need the backup versions?

Examine the default policy to see if it meets your needs:

- Up to two backup versions of a file on the client’s system are retained in server storage.
- The most recent backup version is retained for as long as the original file is on the client file system. All other versions are retained for up to 30 days after they become inactive.

- One backup version of a file that has been deleted from the client’s system is retained in server storage for 60 days.
- An archive copy is kept for up to 365 days.

The server manages files based on whether the files are active or inactive. The most current backup or archived copy of a file is the active version. All other versions are called inactive versions. An active version of a file becomes inactive when:

- A new backup is made
- A user deletes that file on the client node and then runs an incremental backup

Policy determines how many inactive versions of files the server keeps, and for how long. When files exceed the criteria, the files expire. Expiration processing can then remove the files from the server database.

Related reference

“File expiration and expiration processing” on page 439

“Running expiration processing to delete expired files” on page 472

“Reviewing the standard policy”

Reviewing the standard policy

The standard policy consists of a standard policy domain, policy set, management class, backup copy group, and archive copy group. Each of these parts is named STANDARD.

The attributes of the default policy are as follows:

Table 45. Summary of default policy

Policy	Object where the policy is set
Backup Policies	
Files are backed up to the default disk storage pool, BACKUPPOOL .	STANDARD backup copy group, DESTINATION parameter
An incremental backup is performed only if the file has changed since the last backup.	STANDARD backup copy group, MODE parameter
Files cannot be backed up while they are being modified.	STANDARD backup copy group, SERIALIZATION parameter
Up to two backup versions of a file on the client’s system are retained in server storage. The most recent backup version is retained for as long as the original file is on the client file system. All other versions are retained for up to 30 days after they become inactive.	STANDARD backup copy group, the following parameters: VEREXISTS RETEXTRA REONLY
One backup version of a file that has been deleted from the client’s system is retained in server storage for 60 days.	STANDARD backup copy group, VERDELETED parameter
When a backed up file is no longer associated with a backup copy group, it remains in server storage for 30 days (backup retention grace period).	STANDARD policy domain, BACKRETENTION parameter
Archive Policies	
Files are archived in the default disk storage pool, ARCHIVEPOOL .	STANDARD archive copy group, DESTINATION parameter
Files cannot be archived while they are being modified.	STANDARD archive copy group, SERIALIZATION parameter

Table 45. Summary of default policy (continued)

Policy	Object where the policy is set
An archive copy is kept for up to 365 days.	STANDARD archive copy group, RETVER parameter
When an archived file is no longer associated with an archive copy group, it remains in server storage for 365 days (archive retention grace period).	STANDARD policy domain, ARCHRETENTION parameter
General	
The default management class is STANDARD.	STANDARD policy set (ACTIVE), ASSIGN DEFMGMTCLASS command
Tivoli Storage Manager for Space Management (HSM) Policy	
Client files are not space-managed (there are no HSM clients).	STANDARD management class, SPACEMGTECHNIQUE parameter

Related reference

“The parts of a policy” on page 443

Getting users started

When you register a client node, the default is to assign the node to the STANDARD policy domain. If users register their own workstations during open registration, they are also assigned to the STANDARD policy domain.

To help users take advantage of IBM Tivoli Storage Manager, you can further tune the policy environment by performing the following tasks:

- Define sets of client options for the different groups of users.
- Help users with creating the include-exclude list. For example:
 - Create include-exclude lists to help inexperienced users who have simple file management needs. One way to do this is to define a basic include-exclude list as part of a client option set. This also gives the administrator some control over client usage.
 - Provide a sample include-exclude list to users who want to specify how the server manages their files. You can show users who prefer to manage their own files how to:
 - Request information about management classes
 - Select a management class that meets backup and archive requirements
 - Use include-exclude options to select management classes for their files

For information on the include-exclude list, see the user’s guide for the appropriate client.

- Automate incremental backup procedures by defining schedules for each policy domain. Then associate schedules with client nodes in each policy domain.

Related tasks

“Creating client option sets on the server” on page 411

Chapter 16, “Scheduling operations for client nodes,” on page 525

Related reference

“The include-exclude list” on page 448

Changing policy

Some types of clients and situations require policy changes. For example, if you need to direct client data to storage pools different from the default storage pools, you need to change policy.

Other situations may also require policy changes. See “Policy configuration scenarios” on page 482 for details.

To change policy that you have established in a policy domain, you must replace the ACTIVE policy set. You replace the ACTIVE policy set by activating another policy set. Perform the following steps:

1. Create or modify a policy set so that it contains the policy that you want to implement.
 - Create a new policy set either by defining a new policy set or by copying a policy set.
 - Modify an existing policy set (it cannot be the ACTIVE policy set).

Note: You cannot directly modify the ACTIVE policy set. If you want to make a small change to the ACTIVE policy set, copy the policy to modify it and follow the steps here.

2. Make any changes that you need to make to the management classes, backup copy groups, and archive copy groups in the new policy set.
3. Validate the policy set.
4. Activate the policy set. The contents of your new policy set becomes the ACTIVE policy set.

Related tasks

“Defining and updating an archive copy group” on page 468

“Policy configuration scenarios” on page 482

Related reference

“Validating a policy set” on page 470

“Activating a policy set” on page 471

“Defining and updating a management class” on page 461

“Defining and updating a backup copy group” on page 462

File expiration and expiration processing

An expired file is a file that the server no longer needs to keep, according to policy.

Files expire under the following conditions:

- Users delete file spaces from client nodes
- Users expire files by using the EXPIRE command on the client
- A file that is a backup version exceeds the criteria in the backup copy group (how long a file is kept and how many inactive versions of a file are kept)
- An archived file exceeds the time criteria in the archive copy group (how long archived copies are kept)
- A backup set exceeds the retention time that is specified for it

Important:

1. A base file is not eligible for expiration until all of its dependent subfiles have been expired.

2. An archive file is not eligible for expiration if there is a deletion hold on it. If a file is not held, it will be handled according to existing expiration processing.

The server deletes expired files from the server database only during expiration processing. After expired files are deleted from the database, the server can reuse the space in the storage pools that was occupied by expired files. You should ensure that expiration processing runs periodically to allow the server to reuse space.

Expiration processing also removes from the database any restartable restore sessions that exceed the time limit set for such sessions by the `RESTOREINTERVAL` server option.

Related concepts

“Managing client restartable restore sessions” on page 417

“Deletion hold” on page 475

“Expiration processing of base files and subfiles” on page 513

Related tasks

“Reclaiming space in sequential-access storage pools” on page 319

Related reference

“Running expiration processing to delete expired files” on page 472

Client operations controlled by policy

IBM Tivoli Storage Manager policies govern the backup and restore, archive and retrieve, and client migration and recall client operations.

Related concepts

“Backup and restore”

“Archive and retrieve” on page 441

“Client migration and recall” on page 441

Backup and restore

Backup-archive clients can back up and restore files and directories. Backup-archive clients on UNIX, Linux, and Windows systems can also back up and restore logical volumes.

Backups allow users to preserve different versions of files as they change.

Backup

To guard against the loss of information, the backup-archive client can copy files, subdirectories, and directories to media controlled by the server. Backups can be controlled by administrator-defined policies and schedules, or users can request backups of their own data.

The backup-archive client provides two types of backup:

Incremental backup

The backup of files according to policy defined in the backup copy group of the management class for the files. An incremental backup typically backs up all files that are new or that have changed since the last incremental backup.

Selective backup

Backs up only files that the user specifies. The files must also meet some of the policy requirements defined in the backup copy group.

See *Backup-Archive Clients Installation and User's Guide* for details on backup-archive clients that can also back up logical volumes. The logical volume must meet some of the policy requirements that are defined in the backup copy group.

Related reference

"Policy for logical volume backups" on page 483

Restore

When a user restores a backup version of a file, the server sends a copy of the file to the client node. The backup version remains in server storage. Restoring a logical volume backup works the same way.

If more than one backup version exists, a user can restore the active backup version or any inactive backup versions.

If policy is properly set up, a user can restore backed-up files to a specific time.

Related reference

"Setting policy to enable point-in-time restore for clients" on page 488

Archive and retrieve

To preserve files for later use or for records retention, a user with a backup-archive client can archive files, subdirectories, and directories on media controlled by the server. When users archive files, they can choose to have the backup-archive client erase the original files from their workstation after the client archives the files.

When a user retrieves a file, the server sends a copy of the file to the client node. The archived file remains in server storage.

Client migration and recall

When the Tivoli Storage Manager for Space Management product is on the workstation, a user can migrate files from workstation storage to server storage and recall those files as needed.

Tivoli Storage Manager for Space Management frees space for new data and makes more efficient use of your storage resources. The installed Tivoli Storage Manager for Space Management product is also called the space manager client or the HSM client. Files that are migrated and recalled with the HSM client are called space-managed files.

For details about using Tivoli Storage Manager for Space Management, see *Space Management for UNIX: User's Guide*.

Migration

When a file is migrated to the server, it is replaced on the client node with a small stub file of the same name as the original file. The stub file contains data needed to locate the migrated file on server storage.

Tivoli Storage Manager for Space Management provides selective and automatic migration. Selective migration lets users migrate files by name. The two types of automatic migration are:

Threshold

If space usage exceeds a high threshold set at the client node, migration begins and continues until usage drops to the low threshold also set at the client node.

Demand

If an out-of-space condition occurs for a client node, migration begins and continues until usage drops to the low threshold.

To prepare for efficient automatic migration, Tivoli Storage Manager for Space Management copies a percentage of user files from the client node to the IBM Tivoli Storage Manager server. The premigration process occurs whenever Tivoli Storage Manager for Space Management completes an automatic migration. The next time free space is needed at the client node, the files that have been pre-migrated to the server can quickly be changed to stub files on the client. The default premigration percentage is the difference between the high and low thresholds.

Files are selected for automatic migration and premigration based on the number of days since the file was last accessed and also on other factors set at the client node.

Recall

Tivoli Storage Manager for Space Management provides selective and transparent recall. Selective recall lets users recall files by name. Transparent recall occurs automatically when a user accesses a migrated file.

When recalling active file versions, the server searches in an active-data storage pool associated with a FILE device class, if such a pool exists.

Related concepts

“Active-data pools as sources of active file versions for server operations” on page 225

Reconciliation

Migration and premigration can create inconsistencies between stub files on the client node and space-managed files in server storage.

For example, if a user deletes a migrated file from the client node, the copy remains at the server. At regular intervals set at the client node, IBM Tivoli Storage Manager compares client node and server storage and reconciles the two by deleting from the server any outdated files or files that do not exist at the client node.

The parts of a policy

Policy administrators use IBM Tivoli Storage Manager policy to specify how files are backed up, archived, migrated from client node storage, and managed in server storage.

Figure 66 shows the parts of a policy and the relationships among the parts.

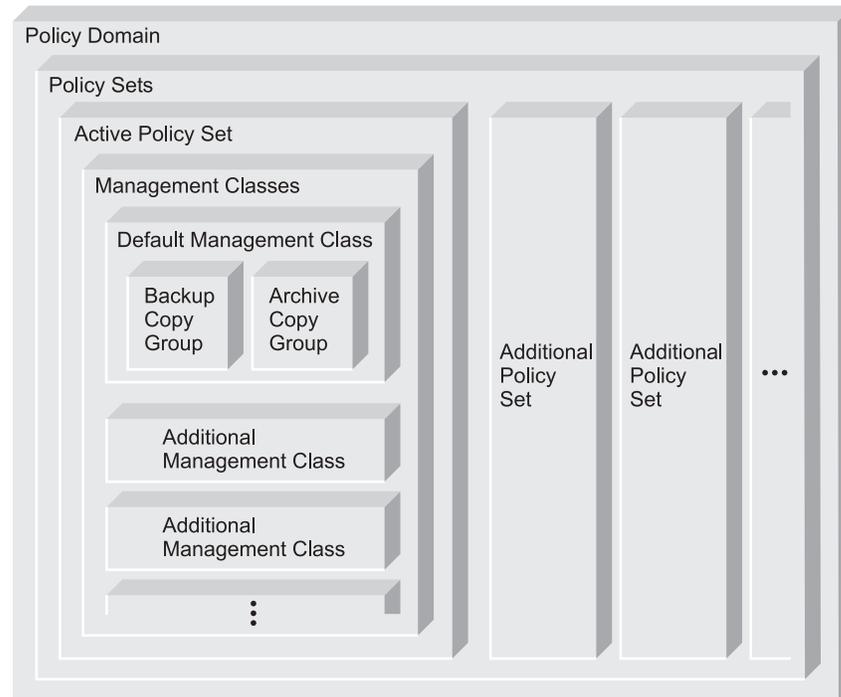


Figure 66. IBM Tivoli Storage Manager Policy

Backup copy group

Controls the backup processing of files associated with the management class. A backup copy group determines the following items:

- How frequently a file can be backed up
- How to handle files that are in use during a backup
- Where the server initially stores backup versions of files and directories
- How many backup versions the server keeps of files and directories
- How long the server keeps backup versions of files and directories

Archive copy group

Controls the archive processing of files associated with the management class. An archive copy group determines the following items:

- How to handle files that are in use during archive
- Where the server stores archived copies of files
- How long the server keeps archived copies of files

Management class

Associates backup and archive groups with files, and specifies if and how client node files are migrated to storage pools. A management class can contain one backup or archive copy group, both a backup and archive

copy group, or no copy groups. Users can bind (that is, associate) their files to a management class through the include-exclude list.

Policy set

Specifies the management classes that are available to groups of users. Policy sets contain one or more management classes. You must identify one management class as the default management class. Only one policy set, the ACTIVE policy set, controls policy operations.

Policy domain

Lets an administrator group client nodes by the policies that govern their files and by the administrators who manage their policies. A policy domain contains one or more policy sets, but only one policy set (named ACTIVE) can be active at a time. The server uses only the ACTIVE policy set to manage files for client nodes assigned to a policy domain.

You can use policy domains to:

- Group client nodes with similar file management requirements
- Provide different default policies for different groups of clients
- Direct files from different groups of clients to different storage hierarchies based on need (different file destinations with different storage characteristics)
- Restrict the number of management classes to which clients have access

Related concepts

“More on management classes” on page 446

Related reference

“Example: sample policy objects” on page 457

“Running expiration processing to delete expired files” on page 472

Relationships among clients, storage, and policy

The relationship among the physical device environment, storage and policy objects, and clients are represented in a figure.

The numbers in the following list correspond to the numbers in the figure.

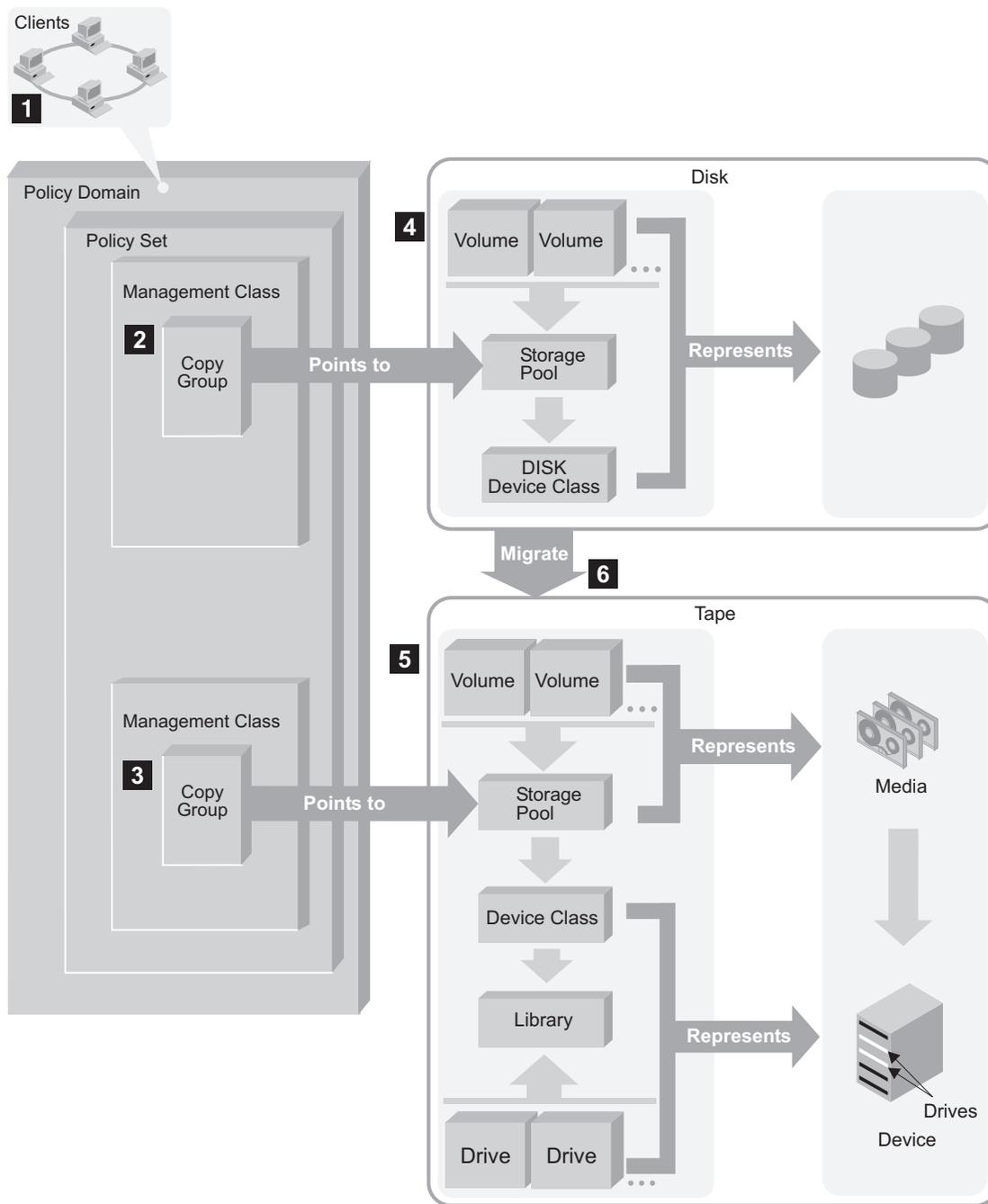


Figure 67. How clients, server storage, and policy work together

- 1** When clients are registered, they are associated with a policy domain. Within the policy domain are the policy set, management class, and copy groups.
- 2, 3** When a client backs up, archives, or migrates a file, it is bound to a management class. A management class and the backup and archive copy groups within it specify where files are stored and how they are managed when they are backed up, archived, or migrated from the client.
- 4, 5** Storage pools are the destinations for backed-up, archived, or

space-managed files. Copy groups specify storage pools for backed-up or archived files. Management classes specify storage pools for space-managed files.

Storage pools are mapped to device classes, which represent devices. The storage pool contains volumes of the type indicated by the associated device class. The example below illustrates this concept:

- A storage pool that is mapped to a device class with a device type of 8 MM contains only 8 mm tapes.

6

Files that are initially stored on disk storage pools can migrate to the following place if the pools are set up in a storage hierarchy:

- Tape storage pools

Figure 67 on page 445 summarizes the relationships among the physical device environment, IBM Tivoli Storage Manager storage and policy objects, and clients.

More on management classes

Management classes are the key connection between client files and policy. Each client node is assigned to a single policy domain, and the client node has access only to the management classes contained in the active policy set.

The management classes specify whether client files are migrated to storage pools (hierarchical storage management). The copy groups in these management classes specify the number of backup versions retained in server storage and the length of time to retain backup versions and archive copies.

For example, if a group of users needs only one backup version of their files, you can create a policy domain that contains only one management class whose backup copy group allows only one backup version. Then you can assign the client nodes for these users to the policy domain.

Related tasks

“Registering nodes with the server” on page 368

Related reference

“Contents of a management class”

“Default management classes” on page 447

“The include-exclude list” on page 448

“How files and directories are associated with a management class” on page 449

Contents of a management class

A management class contains policy for backup, archive, and space management operations by clients. You can specify if and how a Tivoli Storage Manager for Space Management client can migrate files to server storage with parameters in the management class.

For clients using the server for backup and archive, you can choose what a management class contains from the following options:

A backup copy group and an archive copy group

Typical end users need to back up and archive documents, spreadsheets, and graphics.

A backup copy group only

Some users only want to back up files (such as working documents, database, log, or history files that change daily). Some application clients need only a backup copy group because they never archive files.

An archive copy group only

A management class that contains only an archive copy group is useful for users who create:

- Point-in-time files. For example, an engineer can archive the design of an electronic component and the software that created the design. Later, the engineer can use the design as a base for a new electronic component.
- Files that are rarely used but need to be retained for a long time. A client can erase the original file without affecting how long the archive copy is retained in server storage. Examples include legal records, patient records, and tax forms.

Attention: A management class that contains neither a backup nor an archive copy group prevents a file from ever being backed up or archived. This type of management class is not recommended for most users. Use such a management class carefully to prevent users from mistakenly selecting it. If users bind their files to a management class without copy groups, IBM Tivoli Storage Manager issues warning messages.

Default management classes

Each policy set must include a default management class.

The default management class is used for the following purposes:

- To manage files that are not bound to a specific management class, as defined by the INCLUDE option in the include-exclude list.
- To manage existing backup versions when an administrator deletes a management class or a backup copy group from the server.
- To manage existing archive copies when an administrator deletes a management class or an archive copy group from the server. The server does not rebind archive copies, but does use the archive copy group (if one exists) in the default management class.
- To manage files when a client node is assigned to a new policy domain and the active policy set does not have management classes with the same names as that to which the node's files are bound.

A typical default management class should perform the following things:

- Meet the needs of most users
- Contain both a backup copy group and an archive copy group
- Set serialization static or shared static to ensure the integrity of backed up and archived files
- Retain backup versions and archive copies for a sufficient amount of time
- Retain directories for at least as long as any files are associated with the directory

Other management classes can contain copy groups tailored either for the needs of special sets of users or for the needs of most users under special circumstances.

Related reference

“How files and directories are associated with a management class” on page 449

The include-exclude list

A user can define an include-exclude list to specify which files are eligible for the different processes that the client can run. Include and exclude options in the list determine which files are eligible for backup and archive services and which files can be migrated from the client (space-managed).

The options also include how the server controls symbolic links and processing such as image, compression and encryption.

If a user does not create an include-exclude list, the following default conditions apply:

- All files belonging to the user are eligible for backup and archive services.
- The default management class governs backup, archive, and space-management policies.

Figure 68 shows an example of an include-exclude list. The statements in this example list perform the following actions:

- Excludes certain files or directories from backup, archive, and client migration operations

Line 1 in Figure 68 means that the SSTEINER node ID excludes all core files from being eligible for backup and client migration.

- Includes some previously excluded files

Line 2 in Figure 68 means that the files in the following directory are excluded:

– /home/ssteiner

The include statement that follows on line 3, however, means that the options.scr file in that directory is eligible for backup and client migration.

- Binds a file to a specific management class

Line 4 in Figure 68 means that all files and subdirectories belonging to the following directory are managed by the policy defined in the MCENGBK2 management class:

– /home/ssteiner/driver5

```
exclude ../../core
exclude /home/ssteiner/*
include /home/ssteiner/options.scr
include /home/ssteiner/driver5/../../* mcengbk2
```

Figure 68. Example of an include-exclude list

IBM Tivoli Storage Manager processes the include-exclude list from the bottom up, and stops when it finds an include or exclude statement that matches the file it is processing. Therefore, the order in which the include and exclude options are listed affects which files are included and excluded. For example, suppose you switch the order of two lines in the example, as follows:

```
include /home/ssteiner/options.scr
exclude /home/ssteiner/*
```

The exclude statement comes last, and excludes all files in the following directory:

- /home/ssteiner

When IBM Tivoli Storage Manager is processing the include-exclude list for the `options.scr` file, it finds the exclude statement first. This time, the `options.scr` file is excluded.

Some options are evaluated after the more basic include and exclude options. For example, options that exclude or include files for compression are evaluated after the program determines which files are eligible for the process being run.

You can create include-exclude lists as part of client options sets that you define for clients.

For detailed information on the include and exclude options, see the user's guide for the appropriate client.

Related tasks

"Creating client option sets on the server" on page 411

How files and directories are associated with a management class

Binding is the process of associating a file with a management class. The policies defined in the management class then apply to the bound files. The server binds a file to a management class when a client backs up, archives, or migrates the file.

A client chooses a management class as follows:

- For backing up a file, a client can specify a management class in the client's include-exclude list (include-exclude options file for UNIX and Linux clients), or can accept the default management class.
- For backing up directories, the client can specify a management class by using the DIRMC option in the client options file.

Important: It is recommended that you define a default management class. If no management class is specified for a directory, the server chooses the management class with the longest retention period in the backup copy group (retention period for the only backup version). When two or more management classes have the same, "longest" retention period, the Tivoli Storage Manager client selects the management class whose name is last in alphabetical order.

- For backing up a file system or logical volume, a client can specify a management class in the client's include-exclude list (include-exclude options file for UNIX and Linux clients), or can accept the default management class.
- For archiving a file, the client can do one of the following tasks:
 - Specify a management class in the client's include-exclude list (with either an include option or an include.archive option)
 - Specify a management class with the ARCHMC option on the archive command
 - Accept the default management class
- For archiving directories, the client can specify a management class with the archiving options, or the ARCHMC option.

Important: It is recommended that you define a default management class. If the client does not specify any archiving options, the server assigns the default management class to the archived directory. If the default management class has no archive copy group, the server assigns the management class that currently has the archive copy group with the shortest retention time. When two or more

management classes have the same, "shortest" retention period, the Tivoli Storage Manager client selects the management class whose name is last in alphabetical order.

- For migrating a file, a client can specify a management class in the client's include-exclude options file, or can accept the default management class.

The default management class is the management class identified as the default in the active policy set.

A management class specified with a simple include option can apply to one or more processes on the client. More specific include options (such as `include.archive`) allow the user to specify different management classes. Some examples of how this works:

- If a client backs up, archives, and migrates a file to the same server, and uses only a single include option, the management class specified for the file applies to all three operations (backup, archive, and migrate).
- If a client backs up and archives a file to one server, and migrates the file to a different server, the client can specify one management class for the file for backup and archive operations, and a different management class for migrating.
- Clients can specify a management class for archiving that is different from the management class for backup.

See the user's guide for the appropriate client for more details.

Effects of changing a management class

A file remains bound to a management class even if the attributes of the management class or its copy groups change.

The following scenario illustrates this process:

1. A file named `REPORT.TXT` is bound to the default management class that contains a backup copy group specifying that up to three backup versions can be retained in server storage.
2. During the next week, three backup versions of `REPORT.TXT` are stored in server storage. The active and two inactive backup versions are bound to the default management class.
3. The administrator assigns a new default management class that contains a backup copy group specifying only up to two backup versions.
4. The administrator then activates the policy set, and the new default management class takes effect.
5. `REPORT.TXT` is backed up again, bringing the number of versions to four. The server determines that according to the new backup copy group only two versions are to be retained. Therefore, the server marks the two oldest versions for deletion (expired).
6. Expiration processing occurs. `REPORT.TXT` is still bound to the default management class, which now includes new retention criteria. Therefore, the two versions marked for deletion are purged, and one active and one inactive backup version remain in storage.

Related reference

"Running expiration processing to delete expired files" on page 472

Rebinding files to management classes

Rebinding is the process of associating a file or a logical volume image with a new management class.

Backup versions

The server rebinds backup versions of files and logical volume images in some cases.

The following list highlights the cases when a server rebinds backup versions of files:

- The user changes the management class specified in the include-exclude list and does a backup.
- An administrator activates a policy set in the same policy domain as the client node, and the policy set does not contain a management class with the same name as the management class to which a file is currently bound.
- An administrator assigns a client node to a different policy domain, and the active policy set in that policy domain does not have a management class with the same name.

Backup versions of a directory can be rebound when the user specifies a different management class using the DIRMC option in the client option file, and when the directory gets backed up.

If a file is bound to a management class that no longer exists, the server uses the default management class to manage the backup versions. When the user does another backup, the server rebinds the file and any backup versions to the default management class. If the default management class does not have a backup copy group, the server uses the backup retention grace period specified for the policy domain.

Archive copies

Archive copies are never rebound because each archive operation creates a different archive copy. Archive copies remain bound to the management class name specified when the user archived them.

If the management class to which an archive copy is bound no longer exists or no longer contains an archive copy group, the server uses the default management class. If you later change or replace the default management class, the server uses the updated default management class to manage the archive copy.

If the default management class does not contain an archive copy group, the server uses the archive retention grace period specified for the policy domain.

How Tivoli Storage Manager selects files for policy operations

The IBM Tivoli Storage Manager selects files for full and partial incremental backups, selective backups, logical volume backups, archives, and automatic migration from an HSM client (Tivoli Storage Manager for Space Management).

Incremental backup

Backup-archive clients can choose to back up their files using full or partial incremental backup. A full incremental backup ensures that clients' backed-up files are always managed according to policies. Clients are urged to use full incremental backup whenever possible.

If the amount of time for backup is limited, clients may sometimes need to use partial incremental backup. A partial incremental backup should complete more quickly and require less memory. When a client uses partial incremental backup, only files that have changed since the last incremental backup are backed up. Attributes in the management class that would cause a file to be backed up when doing a full incremental backup are ignored. For example, unchanged files are not backed up even when they are assigned to a management class that specifies absolute mode and the minimum days between backups (frequency) has passed.

The server also does less processing for a partial incremental backup. For example, the server does not expire files or rebind management classes to files during a partial incremental backup.

If clients must use partial incremental backups, they should periodically perform full incremental backups to ensure that complete backups are done and backup files are stored according to policies. For example, clients can do partial incremental backups every night during the week, and a full incremental backup on the weekend.

Performing full incremental backups is important if clients want the ability to restore files to a specific time. Only a full incremental backup can detect whether files have been deleted since the last backup. If full incremental backup is not done often enough, clients who restore to a specific time may find that many files that had actually been deleted from the workstation get restored. As a result, a client's file system may run out of space during a restore process.

Related reference

“Setting policy to enable point-in-time restore for clients” on page 488

Full incremental backup

When a user requests a full incremental backup, the IBM Tivoli Storage Manager determines its eligibility.

The IBM Tivoli Storage Manager ensures the following items are identified:

1. Checks each file against the user's include-exclude list:
 - Files that are excluded are not eligible for backup.
 - If files are not excluded and a management class is specified with the INCLUDE option, IBM Tivoli Storage Manager uses that management class.
 - If files are not excluded but a management class is not specified with the INCLUDE option, IBM Tivoli Storage Manager uses the default management class.
 - If no include-exclude list exists, all files in the client domain are eligible for backup, and IBM Tivoli Storage Manager uses the default management class.
2. Checks the management class of each included file:
 - If there is a backup copy group, the process continues with step 3.
 - If there is no backup copy group, the file is not eligible for backup.
3. Checks the mode, frequency, and serialization defined in the backup copy group.

Mode Specifies whether the file is backed up only if it has changed since the last backup (modified) or whenever a backup is requested (absolute).

Frequency

Specifies the minimum number of days that must elapse between backups.

Tip: For Windows this attribute is ignored during a journal-based backup.

Serialization

Specifies how files are handled if they are modified while being backed up and what happens if modification occurs.

- If the mode is modified and the minimum number of days have elapsed since the file was last backed up, IBM Tivoli Storage Manager determines if the file has been changed since it was last backed up:
 - If the file has been changed and the serialization requirement is met, the file is backed up.
 - If the file has not been changed, it is not backed up.
- If the mode is modified and the minimum number of days have not elapsed since the file was last backed up, the file is not eligible for backup.
- If the mode is absolute, the minimum number of days have elapsed since the file was last backed up, and the serialization requirement is met, the file is backed up.
- If the mode is absolute and the minimum number of days have not elapsed since the file was last backed up, the file is not eligible for backup.

Partial incremental backup

When a user requests a partial incremental backup, the IBM Tivoli Storage Manager determines its eligibility.

IBM Tivoli Storage Manager ensures the following items are identified:

1. Checks each file against the user's include-exclude list:
 - Files that are excluded are not eligible for backup.
 - If files are not excluded and a management class is specified with the INCLUDE option, the server uses that management class.
 - If files are not excluded but a management class is not specified with the INCLUDE option, the server uses the default management class.
 - If no include-exclude list exists, all files in the client domain are eligible for backup, and the server uses the default management class.
2. Checks the management class of each included file:
 - If there is a backup copy group, the process continues with step 3.
 - If there is no backup copy group, the file is not eligible for backup.
3. Checks the date and time of the last incremental backup by the client, and the serialization requirement defined in the backup copy group. (Serialization specifies how files are handled if they are modified while being backed up and what happens if modification occurs.)
 - If the file has not changed since the last incremental backup, the file is not backed up.
 - If the file has changed since the last incremental backup and the serialization requirement is met, the file is backed up.

Selective backup

When a user requests a selective backup, the IBM Tivoli Storage Manager ensures its eligibility.

IBM Tivoli Storage Manager ensures the following items are identified:

1. Checks the file against any include or exclude statements contained in the user include-exclude list:
 - Files that are not excluded are eligible for backup. If a management class is specified with the INCLUDE option, IBM Tivoli Storage Manager uses that management class.
 - If no include-exclude list exists, the files selected are eligible for backup, and IBM Tivoli Storage Manager uses the default management class.
2. Checks the management class of each included file:
 - If the management class contains a backup copy group and the serialization requirement is met, the file is backed up. Serialization specifies how files are handled if they are modified while being backed up and what happens if modification occurs.
 - If the management class does not contain a backup copy group, the file is not eligible for backup.

An important characteristic of selective backup is that a file is backed up without regard for whether the file has changed. This result may not always be what you want. For example, suppose a management class specifies to keep three backup versions of a file. If the client uses incremental backup, the file is backed up only when it changes, and the three versions in storage will be at different levels. If the client uses selective backup, the file is backed up regardless of whether it has changed. If the client uses selective backup on the file three times without changing the file, the three versions of the file in server storage are identical. Earlier, different versions are lost.

Logical volume backup

When a user requests a logical volume backup, the IBM Tivoli Storage Manager determines its eligibility.

IBM Tivoli Storage Manager ensures the following items are identified:

1. Checks the specification of the logical volume against any include or exclude statements contained in the user include-exclude list:
 - If no include-exclude list exists, the logical volumes selected are eligible for backup, and IBM Tivoli Storage Manager uses the default management class.
 - Logical volumes that are not excluded are eligible for backup. If the include-exclude list has an INCLUDE option for the volume with a management class specified, IBM Tivoli Storage Manager uses that management class. Otherwise, the default management class is used.
2. Checks the management class of each included logical volume:
 - If the management class contains a backup copy group and the logical volume meets the serialization requirement, the logical volume is backed up. Serialization specifies how logical volumes are handled if they are modified while being backed up and what happens if modification occurs.
 - If the management class does not contain a backup copy group, the logical volume is not eligible for backup.

Archive

When a user requests the archiving of a file or a group of files, the IBM Tivoli Storage Manager determine its eligibility.

IBM Tivoli Storage Manager ensures the following items are identified:

1. Checks the files against the user's include-exclude list to see if any management classes are specified:
 - IBM Tivoli Storage Manager uses the default management class for files that are not bound to a management class.
 - If no include-exclude list exists, IBM Tivoli Storage Manager uses the default management class unless the user specifies another management class. See the user's guide for the appropriate client for details.
2. Checks the management class for each file to be archived.
 - If the management class contains an archive copy group and the serialization requirement is met, the file is archived. Serialization specifies how files are handled if they are modified while being archived and what happens if modification occurs.
 - If the management class does not contain an archive copy group, the file is not archived.

Remember: If you need to frequently create archives for the same data, consider using instant archive (backup sets) instead. Frequent archive operations can create a large amount of metadata in the server database resulting in increased database growth and decreased performance for server operations such as expiration. Frequently, you can achieve the same objectives with incremental backup or backup sets. Although the archive function is a powerful way to store inactive data with fixed retention, it should not be used on a frequent and large scale basis as the primary backup method.

Related concepts

"Creating and using client backup sets" on page 502

Automatic migration from a client node

A file is eligible for automatic migration from an HSM client (Tivoli Storage Manager for Space Management) if it meets certain criteria.

The criteria for a file to be eligible for automatic migration from an HSM client are displayed in the following list:

- It resides on a node on which the root user has added and activated hierarchical storage management. It must also reside in a local file system to which the root user has added space management, and not in the root (/) or /tmp file system.
- It is not excluded from migration in the include-exclude list.
- It meets management class requirements for migration:
 - The file is not a character special file, a block special file, a FIFO special file (that is, a named pipe file) or a directory.
 - The file is assigned to a management class that calls for space management.
 - The management class calls for automatic migration after a specified number of days, and that time has elapsed.
 - A backup version of the file exists if the management class requires it.
 - The file is larger than the stub file that would replace it (plus one byte) or the file system block size, whichever is larger.

How client migration works with backup and archive

As an administrator, you can define a management class that specifies automatic migration from the client under certain conditions.

Note: The situation described is valid only when Space Management is installed and configured. You can perform automatic migration only when using the Space Management client.

For example, if the file has not been accessed for at least 30 days and a backup version exists, the file is migrated. You can also define a management class that allows users to selectively migrate whether or not a backup version exists. Users can also choose to archive files that have been migrated. IBM Tivoli Storage Manager manages the following situations:

- If the file is backed up or archived to the server to which it was migrated, the server copies the file from the migration storage pool to the backup or archive storage pool. For a tape-to-tape operation, each storage pool must have a tape drive.
- If the file is backed up or archived to a different server, Tivoli Storage Manager accesses the file by using the migrate-on-close recall mode. The file resides on the client node only until the server stores the backup version or the archived copy in a storage pool.

When a client restores a backup version of a migrated file, the server deletes the migrated copy of the file from server storage the next time reconciliation is run.

When a client archives a file that is migrated and does not specify that the file is to be erased after it is archived, the migrated copy of the file remains in server storage. When a client archives a file that is migrated and specifies that the file is to be erased, the server deletes the migrated file from server storage the next time reconciliation is run.

The Tivoli Storage Manager default management class specifies that a backup version of a file must exist before the file is eligible for migration.

Creating your own policies

You can create your own policies by defining the parts of a policy and specifying each attribute, or by copying existing policy parts and updating only those attributes that you want to change.

Task	Required Privilege Class
Define or copy a policy domain	System
Update a policy domain over which you have authority	Restricted policy
Define, update, or copy policy sets and management classes in any policy domain	System or unrestricted policy
Define, update, or copy policy sets and management classes in policy domains over which you have authority	Restricted policy
Define or update copy groups in any policy domain	System or unrestricted policy

Task	Required Privilege Class
Define or update copy groups that belong to policy domains over which you have authority	Restricted policy
Assign a default management class to a nonactive policy set in any policy domain	System or unrestricted policy
Assign a default management class to a nonactive policy set in policy domains over which you have authority	Restricted policy
Validate and activate policy sets in any policy domain	System or unrestricted policy
Validate and activate policy sets in policy domains over which you have authority	Restricted policy
Start inventory expiration processing	System

Table 46 shows that an advantage of copying existing policy parts is that some associated parts are copied in a single operation.

Table 46. Cause and effect of copying existing policy parts

If you copy this...	Then you create this...
Policy Domain	A new policy domain with: <ul style="list-style-type: none"> • A copy of each policy set from the original domain • A copy of each management class in each original policy set • A copy of each copy group in each original management class
Policy Set	A new policy set in the same policy domain with: <ul style="list-style-type: none"> • A copy of each management class in the original policy set • A copy of each copy group in the original management class
Management Class	A new management class in the same policy set and a copy of each copy group in the management class

Example: sample policy objects

The sample policy objects example is used in several scenarios.

Figure 69 on page 458 shows the policies for an engineering department.

The domain contains two policy sets that are named STANDARD and TEST. The administrator activated the policy set that is named STANDARD. When you activate a policy set, the server makes a copy of the policy set and names it ACTIVE. Only one policy set can be active at a time.

The ACTIVE policy set contains two management classes: MCENG and STANDARD. The default management class is STANDARD.

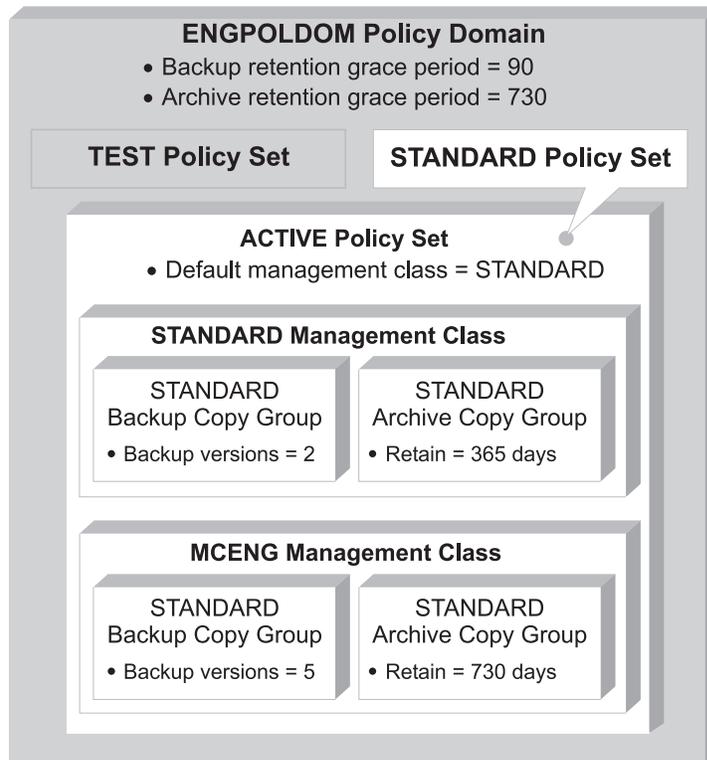


Figure 69. An example of policy objects defined for an engineering department

Related tasks

“Defining and updating an archive copy group” on page 468

Related reference

“Defining and updating a policy domain”

“Defining and updating a policy set” on page 460

“Defining and updating a management class” on page 461

“Defining and updating a backup copy group” on page 462

“Assigning a default management class” on page 470

“Activating a policy set” on page 471

“Running expiration processing to delete expired files” on page 472

Defining and updating a policy domain

When you update or define a policy domain, you specify the backup retention grace period, the archive retention grace period, and the destination for active backup data.

See the following definitions:

Backup Retention Grace Period

Specifies the number of days to retain an inactive backup version when the server cannot rebind the file to an appropriate management class. The backup retention grace period protects backup versions from being immediately expired when the management class to which a file is bound no longer exists or no longer contains a backup copy group, and the default management class does not contain a backup copy group.

Backup versions of the file managed by the grace period are retained in server storage only for the backup retention grace period. This period starts from the day of the backup. For example, if the backup retention grace period for the STANDARD policy domain is used and set to 30 days, backup versions using the grace period expire in 30 days from the day of the backup.

Backup versions of the file continue to be managed by the grace period unless one of the following events occur:

- The client binds the file to a management class containing a backup copy group and then backs up the file
- A backup copy group is added to the file's management class
- A backup copy group is added to the default management class

Archive Retention Grace Period

Specifies the number of days to retain an archive copy when the management class for the file no longer contains an archive copy group and the default management class does not contain an archive copy group. The retention grace period protects archive copies from being immediately expired.

The archive copy of the file managed by the grace period is retained in server storage for the number of days specified by the archive retention grace period. This period starts from the day on which the file is first archived. For example, if the archive retention grace period for the policy domain STANDARD is used, an archive copy expires 365 days from the day the file is first archived.

The archive copy of the file continues to be managed by the grace period unless an archive copy group is added to the file's management class or to the default management class.

Destination for Active Backup Data

Specifies the names active-data pools that store active versions of backup data for nodes assigned to the domain. Before the IBM 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 defined in the list of active-data pools. 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 client backup is performed using simultaneous-write operations to an active-data pool, the server performs the verification during backup operations by IBM Tivoli Storage Manager backup-archive clients or by application clients using the IBM Tivoli Storage Manager API. The verification is also performed when active data is being copied using the COPY ACTIVEDATA command.

Example: defining a policy domain

To create a new policy domain you can either copy an existing policy domain and update the new domain, or define a new policy domain from scratch.

Important: When you copy an existing domain, you also copy any associated policy sets, management classes, and copy groups.

For example, perform the following steps to copy and update an existing domain:

1. Copy the STANDARD policy domain to the ENGPOLDOM policy domain 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.
2. 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 assigned to the domain. Use `engactivedata` as the name of the active-data pool, as in the following example:

```
update domain engpoldom description='Engineering Policy Domain'  
backretention=90 archretention=730 activedestination=engactivedata
```

Defining and updating a policy set

When you define or update a policy set, you must specify the policy domain name.

See the following definition:

Policy domain name

Names the policy domain to which the policy set belongs

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

Related reference

“Activating a policy set” on page 471

Example: defining a policy set

An administrator must develop new policies based on the existing STANDARD policy set.

To create the TEST policy set in the ENGPOLDOM policy domain, the administrator performs the following steps:

1. Copy the STANDARD policy set and name the new policy set TEST:

```
copy policyset engpoldom standard test
```

Note: When you copy an existing policy set, you also copy any associated management classes and copy groups.

2. Update the description of the policy set named TEST:

```
update policyset engpoldom test  
description='Policy set for testing'
```

Defining and updating a management class

When you define or update a management class, you must specify the policy domain name, the policy set name, and the description.

See the following definitions:

Policy domain name

Names the policy domain to which the management class belongs.

Policy set name

Names the policy set to which the management class is assigned.

Description

Describes the management class. A clear description can help users to choose an appropriate management class for their use.

The following four parameters apply only to HSM clients (Tivoli Storage Manager for Space Management):

Whether space management is allowed

Specifies that the files are eligible for both automatic and selective migration, only selective migration, or no migration.

How frequently files can be migrated

Specifies the minimum number of days that must elapse since a file was last accessed before it is eligible for automatic migration.

Whether backup is required

Specifies whether a backup version of a file must exist before the file can be migrated.

Where migrated files are to be stored

Specifies the name of the storage pool in which migrated files are stored. Your choice could depend on factors such as:

- The number of client nodes migrating 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 users need immediate access to migrated versions, you can specify a disk storage pool as the destination.

Attention: You cannot specify a copy storage pool or an active-data pool as a destination.

Example: define a new management class

There are just two steps to creating a new management class.

Perform the following steps to create a new management class:

1. Define a new management class MCENG by entering:

```
define mgmtclass engpoldom standard mceng
```
2. Update the description of the MCENG management class by entering:

```
update mgmtclass engpoldom standard mceng  
description='Engineering Management Class for Backup and Archive'
```

Defining and updating a backup copy group

When you are defining and updating a backup copy group, you will have to know where to store it, how to manage files that are modified during backup, how to designate the frequency of your backups, and how to retain the backup versions.

Related reference

“Where to store backed-up files”

“How to manage files that are modified during backup”

“Defining the frequency of backing up files” on page 463

“Retaining backup versions” on page 464

Where to store backed-up files

Specify a storage pool where the server initially stores the files associated with this backup copy group. This is called the destination.

Your choice can depend on factors such as the following items:

- Whether the server and the client nodes have access to shared devices on a storage area network (SAN).
- The number of client nodes backing up to the storage pool. When many user files are stored in the same storage pool, volume contention can occur as users try to back up to or restore files from the storage pool.
- How quickly the files must be restored. If users need immediate access to backup versions, you may want to specify a disk storage pool as the destination.

Attention: You cannot specify a copy storage pool or an active-data pool as the destination.

How to manage files that are modified during backup

You can use the `SERIALIZATION` attribute on the `DEFINE COPYGROUP` command to specify how files are managed if they are modified during a backup.

This attribute can be one of four values: `STATIC`, `SHRSTATIC` (shared static), `DYNAMIC`, or `SHRDYNAMIC` (shared dynamic).

The value you choose depends on how you want IBM Tivoli Storage Manager to manage files that are modified while they are being backed up.

Do not back up files that are modified during the backup

You will want to prevent the server from backing up a file while it is being modified. Use one of the following values:

STATIC

Specifies that IBM Tivoli Storage Manager will attempt to back up the file only once. If the file or directory is modified during a backup, the server does not back it up.

SHRSTATIC (Shared static)

Specifies that if the file or directory is modified during a backup, the server retries the backup as many times as specified by the `CHANGINGRETRIES` option in the client options file. If the file is modified during the last attempt, the file or directory is not backed up.

Back up files that are modified during the backup

Some files are in constant use, such as an error log. Consequently, these

files may never be backed up when serialization is set to `STATIC` or `SHRSTATIC`. To back up files that are modified during the backup, use one of the following values:

DYNAMIC

Specifies that a file or directory is backed up on the first attempt, even if the file or directory is modified during the backup.

SHRDYNAMIC (Shared dynamic)

Specifies that if a file or directory is modified during a backup, the server retries the backup as many times as specified by the `CHANGINGRETRIES` option in the client options file. The server backs up the file on the last attempt, even if the file or directory is being modified.

Attention:

- If a file is modified during backup and `DYNAMIC` or `SHRDYNAMIC` is specified, then the backup may not contain all the changes and may not be usable. For example, the backup version may contain a truncated record. Under some circumstances, it may be acceptable to capture a dynamic or “fuzzy” backup of a file (the file was changed during the backup). For example, a dynamic backup of an error log file that is continuously appended may be acceptable. However, a dynamic backup of a database file may not be acceptable, since restoring such a backup could result in an unusable database. Carefully consider dynamic backups of files as well as possible problems that may result from restoring potentially “fuzzy” backups.
- When certain users or processes open files, they may deny any other access, including “read” access, to the files by any other user or process. When this happens, even with serialization set to `DYNAMIC` or `SHRDYNAMIC`, IBM Tivoli Storage Manager will not be able to open the file at all, so the server cannot back up the file.

Defining the frequency of backing up files

You can specify how frequently files can be backed up with two parameters, `FREQUENCY` and `MODE`.

See the following definitions:

Frequency

The frequency is the minimum number of days that must elapse between full incremental backups.

Mode The mode parameter specifies whether a file or directory must have been modified to be considered for backup during a full incremental backup process. IBM Tivoli Storage Manager does not check this attribute when a user requests a partial incremental backup, a selective backup for a file, or a backup of a logical volume. You can select from two modes:

Modified

A file is considered for full incremental backup only if it has changed since the last backup. A file is considered changed if any of the following items is different:

- Date on which the file was last modified
- File size
- File owner
- File permissions

Absolute

A file is considered for full incremental backup regardless of whether it has changed since the last backup.

The server considers both parameters to determine how frequently files can be backed up. For example, if frequency is 3 and mode is Modified, a file or directory is backed up only if it has been changed and if three days have passed since the last backup. If frequency is 3 and mode is Absolute, a file or directory is backed up after three days have passed whether or not the file has changed.

Use the Modified mode when you want to ensure that the server retains multiple, different backup versions. If you set the mode to Absolute, users may find that they have three identical backup versions, rather than three different backup versions.

Absolute mode can be useful for forcing a full backup. It can also be useful for ensuring that extended attribute files are backed up, because Tivoli Storage Manager does not detect changes if the size of the extended attribute file remains the same.

When you set the mode to Absolute, set the frequency to 0 if you want to ensure that a file is backed up each time full incremental backups are scheduled for or initiated by a client.

Retaining backup versions

Multiple versions of files are useful when users continually update files and sometimes need to restore the original file from which they started. The most current backup version of a file is called the active version. All other versions are called inactive versions.

You can specify the number of versions to keep by:

- Directly specifying the number of versions
You specify the number of backup versions with two parameters:
 - **Versions Data Exists** (number of versions to keep when the data still exists on the client node)
 - **Versions Data Deleted** (number of versions to keep when the data no longer exists on the client node)
- Specifying the number of days to keep each backup version
You specify the number of days to keep backup versions with two parameters:
 - **Retain Extra Versions** (how many days to keep inactive backup versions; the days are counted from the day that the version became inactive)
 - **Retain Only Versions** (how many days to keep the last backup version of a file that has been deleted)
- Specifying a combination of the number of versions and the days to keep them
Use a combination of the four parameters: **Versions Data Exists**, **Versions Data Deleted**, **Retain Extra Versions**, and **Retain Only Versions**.

These parameters interact to determine the backup versions that the server retains. When the number of inactive backup versions exceeds the number of versions allowed (**Versions Data Exists** and **Versions Data Deleted**), the oldest version expires and the server deletes the file from the database the next time expiration processing runs. How many inactive versions the server keeps is also related to the parameter for how long inactive versions are kept (**Retain Extra Versions**). Inactive

versions expire when the number of days that they have been inactive exceeds the value specified for retaining extra versions, even when the number of versions is not exceeded.

Important: A base file is not eligible for expiration until all its dependent subfiles have been expired.

For example, see Table 47 and Figure 70. A client node has backed up the file REPORT.TXT four times in one month, from March 23 to April 23. The settings in the backup copy group of the management class to which REPORT.TXT is bound determine how the server treats these backup versions. Table 48 on page 466 shows some examples of how different copy group settings would affect the versions. The examples show the effects as of April 24 (one day after the file was last backed up).

Table 47. Status of REPORT.TXT as of April 24

Version	Date Created	Days the Version Has Been Inactive
Active	April 23	(not applicable)
Inactive 1	April 13	1 (since April 23)
Inactive 2	March 31	11 (since April 13)
Inactive 3	March 23	24 (since March 31)

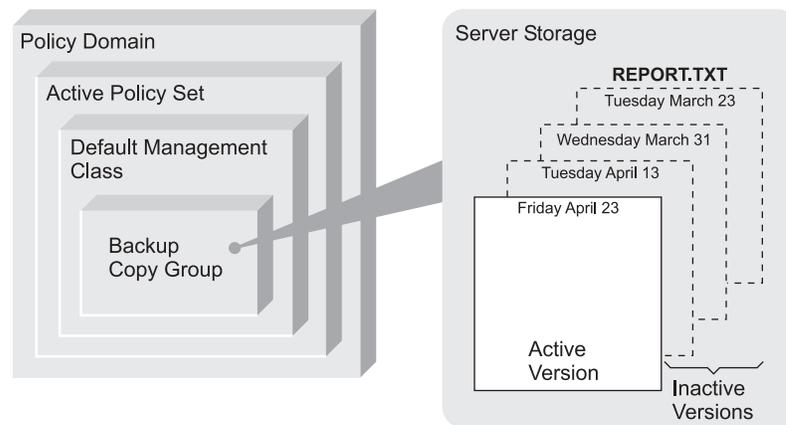


Figure 70. Active and inactive versions of REPORT.TXT

Table 48. Effects of backup copy group policy on backup versions for REPORT.TXT as of april 24. One day after the file was last backed up.

Versions Data Exists	Versions Data Deleted	Retain Extra Versions	Retain Only Version	Results
4 versions	2 versions	60 days	180 days	<p>Versions Data Exists and Retain Extra Versions control the expiration of the versions. The version created on March 23 is retained until the client node backs up the file again (creating a fourth inactive version), or until that version has been inactive for 60 days.</p> <p>If the user deletes the REPORT.TXT file from the client node, the server notes the deletion at the next full incremental backup of the client node. From that point, the Versions Data Deleted and Retain Only Version parameters also have an effect. All versions are now inactive. Two of the four versions expire immediately (the March 23 and March 31 versions expire). The April 13 version expires when it has been inactive for 60 days (on June 23). The server keeps the last remaining inactive version, the April 23 version, for 180 days after it becomes inactive.</p>
NOLIMIT	2 versions	60 days	180 days	<p>Retain Extra Versions controls expiration of the versions. The inactive versions (other than the last remaining version) are expired when they have been inactive for 60 days.</p> <p>If the user deletes the REPORT.TXT file from the client node, the server notes the deletion at the next full incremental backup of the client node. From that point, the Versions Data Deleted and Retain Only Version parameters also have an effect. All versions are now inactive. Two of the four versions expire immediately (the March 23 and March 31 versions expire) because only two versions are allowed. The April 13 version expires when it has been inactive for 60 days (on June 22). The server keeps the last remaining inactive version, the April 23 version, for 180 days after it becomes inactive.</p>
NOLIMIT	NOLIMIT	60 days	180 days	<p>Retain Extra Versions controls expiration of the versions. The server does not expire inactive versions based on the maximum number of backup copies. The inactive versions (other than the last remaining version) are expired when they have been inactive for 60 days.</p> <p>If the user deletes the REPORT.TXT file from the client node, the server notes the deletion at the next full incremental backup of the client node. From that point, the Retain Only Version parameter also has an effect. All versions are now inactive. The three of four versions will expire after each of them has been inactive for 60 days. The server keeps the last remaining inactive version, the April 23 version, for 180 days after it becomes inactive.</p>
4 versions	2 versions	NOLIMIT	NOLIMIT	<p>Versions Data Exists controls the expiration of the versions until a user deletes the file from the client node. The server does not expire inactive versions based on age.</p> <p>If the user deletes the REPORT.TXT file from the client node, the server notes the deletion at the next full incremental backup of the client node. From that point, the Versions Data Deleted parameter controls expiration. All versions are now inactive. Two of the four versions expire immediately (the March 23 and March 31 versions expire) because only two versions are allowed. The server keeps the two remaining inactive versions indefinitely.</p>

See *Administrator's Reference* for details about the parameters. The following list gives some tips on using the NOLIMIT value:

Versions Data Exists

Setting the value to NOLIMIT may require increased storage, but that value may be needed for some situations. For example, to enable client nodes to restore files to a specific point in time, set the value for **Versions Data Exists** to NOLIMIT. Setting the value this high ensures that the server retains versions according to the **Retain Extra Versions** parameter for the copy group.

Versions Data Deleted

Setting the value to NOLIMIT may require increased storage, but that value may be needed for some situations. For example, set the value for **Versions Data Deleted** to NOLIMIT to enable client nodes to restore files to a specific point in time. Setting the value this high ensures that the server retains versions according to the **Retain Extra Versions** parameter for the copy group.

Retain Extra Versions

If NOLIMIT is specified, inactive backup versions are deleted based on the **Versions Data Exists** or **Versions Data Deleted** parameters.

To enable client nodes to restore files to a specific point in time, set the parameters **Versions Data Exists** or **Versions Data Deleted** to NOLIMIT. Set the value for **Retain Extra Versions** to the number of days that you expect clients may need versions of files available for possible point-in-time restoration. For example, to enable clients to restore files from a point in time 60 days in the past, set **Retain Extra Versions** to 60.

Retain Only Version

If NOLIMIT is specified, the last version is retained forever unless a user or administrator deletes the file from server storage.

Related concepts

"Enabling clients to use subfile backup" on page 511

Example: define a backup copy group

Define a backup copy group belonging to the MCENG management class in the STANDARD policy set belonging to the ENGPOLDOM policy domain.

This new copy group must be able to do the following tasks:

- Let users back up changed files, regardless of how much time has elapsed since the last backup, using the default value 0 for the **Frequency** parameter (frequency parameter not specified)
- Retain up to four inactive backup versions when the original file resides on the user workstation, using the **Versions Data Exists** parameter (verexists=5)
- Retain up to four inactive backup versions when the original file is deleted from the user workstation, using the **Versions Data Deleted** parameter (verdeleted=4)
- Retain inactive backup versions for no more than 90 days, using the **Retain Extra Versions** parameter (retextra=90)
- If there is only one backup version, retain it for 600 days after the original is deleted from the workstation, using the **Retain Only Version** parameter (reonly=600)
- Prevent files from being backed up if they are in use, using the **Serialization** parameter (serialization=static)

- Store files in the ENGBACK1 storage pool, using the **Destination** parameter (destination=engback1)

To define the backup copy group, enter:

```
define copygroup engpoldom standard mceng standard
destination=engback1 serialization=static
verexists=5 verdeleted=4 retextra=90 retonly=600
```

Defining and updating an archive copy group

To define or update an archive copy group, you must specify where the archived files are to be stored, if files can be modified during archive, how long to retain an archived copy, and the minimum amount of days to retain an object.

1. Where archived files are to be stored, specify a defined storage pool as the initial destination. Your choice can depend on factors such as:
 - Whether the server and the client nodes have access to shared devices on a SAN
 - The number of client nodes archiving files to the storage pool. When many user files are stored in the same storage pool, volume contention can occur as users archive files to and retrieve files from the storage pool.
 - How quickly the files must be restored. If users need immediate access to archive copies, you could specify a disk storage pool as the destination.
 - Whether the archive copy group is for a management class that is the default for a policy domain. The default management class is used by clients registered in the policy domain, when they do not specify a management class for a file. This includes servers that are registered as clients to this server.

Note: You cannot specify a copy storage pool or an active-data pool as a destination.

2. If files can be modified during archive, specify how files are handled if they are modified while being archived. This attribute, called serialization, can be one of four values:

Static Specifies that if the file is modified during an archiving process, the server does not archive it. IBM Tivoli Storage Manager does not retry the archive.

Shared Static

Specifies that if the file is modified during an archive process, the server does not archive it. However, IBM Tivoli Storage Manager retries the archive process as many times as specified by the CHANGINGRETRIES option in the client options file.

Dynamic

Specifies that a file is archived on the first attempt, even if the file is being modified during the archive process.

Shared Dynamic

Specifies that if the file is modified during the archive attempt, the server archives it on its last try even if the file is being modified. IBM Tivoli Storage Manager retries the archive process as many times as specified by the CHANGINGRETRIES option in the client options file.

For most files, set serialization to either static or shared static to prevent the server from archiving a file while it is being modified.

However, you may want to define a copy group with a serialization of shared dynamic or dynamic for files where log records are continuously added, such

as an error log. If you only have copy groups that use static or shared static, these files may never be archived because they are constantly in use. With shared dynamic or dynamic, the log files are archived. However, the archive copy may contain a truncated message.

Attention: If a file is archived while it is in use (shared dynamic or dynamic serialization), the copy may not contain all the changes and may not be usable.

Note: When certain users or processes open files, they deny read access to the files for any other user or process. When this happens, even with serialization set to dynamic or shared dynamic, the server does not back up the file.

3. How long to retain an archived copy specifies the number of days to retain an archived copy in storage. When the time elapses, the archived copy expires and the server deletes the file the next time expiration processing runs.

When a user archives directories, the server uses the default management class unless the user specifies otherwise. If the default management class does not have an archive copy group, the server binds the directory to the management class that currently has the shortest retention time for archive. When you change the retention time for an archive copy group, you may also be changing the retention time for any directories that were archived using that copy group.

The user can change the archive characteristics by using Archive Options in the interface or by using the ARCHMC option on the command.

4. The **RETMIN** parameter in archive copy groups specifies the minimum number of days an object will be retained after the object is archived. For objects that are managed by event-based retention policy, this parameter ensures that objects are retained for a minimum time period regardless of when an event triggers retention

After you have defined an archive copy group, using the **RETMIN=n** parameter, ensure that the appropriate archive data will be bound to the management class with this archive copy group. You can do this either by using the default management class or by modifying the client options file to specify the management class for the appropriate archive data.

Placing a deletion hold on an object does not extend its retention period. For example, if an object is thirty days away from the end of its retention period and it is placed on hold for ninety days, it will be eligible for expiration immediately upon the hold being released.

Related concepts

“Deletion hold” on page 475

Related tasks

“Using virtual volumes to store data on another server” on page 712

Example: define an archive copy group

Define an archive copy group belonging to the MCENG class.

This copy group must:

- Allow users to archive a file if it is not in use (`serialization=static`)
- Retain the archive copy for 730 days (`retver=730`)
- Store files in the ENGARCH1 storage pool (`destination=engarch1`)

To define a STANDARD archive copy group to the MCENG management class in the STANDARD policy set belonging to the ENGPOLDOM policy domain, enter:

```
define copygroup engpoldom standard mceng standard
type=archive destination=engarch1 serialization=static
retver=730
```

Assigning a default management class

After you have defined a policy set and the management classes that it contains, you must assign a default management class for the policy set.

Related reference

“Default management classes” on page 447

Example: assign a default management class

The example is to assign the STANDARD management class as the default management class for the TEST policy set in the ENGPOLDOM policy domain.

Issue the following command to complete the example:

```
assign defmgmtclass engpoldom standard standard
```

The STANDARD management class was copied from the STANDARD policy set to the TEST policy set. Before the new default management class takes effect, you must activate the policy set.

Related tasks

“Example: defining a policy set” on page 460

Validating and activating a policy set

After you have defined a policy set and defined management classes to it, you can validate the policy set and activate the policy set for the policy domain. Only one policy set is active in a policy domain.

Validating a policy set

When you validate a policy set, the server examines the management class and copy group definitions in the policy set and reports on conditions that need to be considered if the policy set is activated.

Validation fails if the policy set does not contain a default management class. Validation results in warning messages if any of the following conditions exist.

Condition	Reason for warning
The storage destinations specified for backup, archive, or migration do not refer to defined storage pools.	A backup, archive, or migration operation will fail when the operation involves storing a file in a storage pool that does not exist.
A storage destination specified for backup, archive, or migration is a copy storage pool or an active-data pool.	The storage destination must be a primary storage pool.
The default management class does not contain a backup or archive copy group.	When the default management class does not contain a backup or archive copy group, any user files bound to the default management class are not backed up or archived.

Condition	Reason for warning
The current ACTIVE policy set names a management class that is not defined in the policy set being validated.	<p>When users back up files that were bound to a management class that no longer exists in the active policy set, backup versions are rebound to the default management class.</p> <p>When the management class to which an archive copy is bound no longer exists and the default management class does not contain an archive copy group, the archive retention grace period is used to retain the archive copy.</p>
The current ACTIVE policy set contains copy groups that are not defined in the policy set being validated.	When users perform a backup and the backup copy group no longer exists in the management class to which a file is bound, backup versions are managed by the default management class. If the default management class does not contain a backup copy group, backup versions are managed by the backup retention grace period, and the workstation file is not backed up.
A management class specifies that a backup version must exist before a file can be migrated from a client node, but the management class does not contain a backup copy group.	The contradictions within the management classes can cause problems for Tivoli Storage Manager for Space Management (HSM) users.

Related reference

“How files and directories are associated with a management class” on page 449
 “Defining and updating a policy domain” on page 458

Activating a policy set

To activate a policy set, specify a policy domain and policy set name.

When you activate a policy set, the server performs a final validation of the contents of the policy set and copies the original policy set to the ACTIVE policy set.

You cannot update the ACTIVE policy set; the original and the ACTIVE policy sets are two separate objects. For example, updating the original policy set has no effect on the ACTIVE policy set. To change the contents of the ACTIVE policy set, you must create or change another policy set and then activate that policy set.

If data retention protection is active, the following rules apply during policy set validation and activation. The server can be a managed server and receive policy definitions via enterprise configuration, but it will not be possible to activate propagated policy sets if these rules are not satisfied.

- All management classes in the policy set to be validated and 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 validated and activated.
- If an archive copy group exists in the active policy set, the corresponding copy group in the policy set to be validated and activated must have RETVER and RETMIN values at least as large as the corresponding values in the active copy group.

Related concepts

“Data retention protection” on page 474

Related tasks

“Changing policy” on page 439

Example: validating and activating a policy set

Validating and activating the STANDARD policy set in the ENGPOLDOM policy domain requires a two-step process.

Perform the following steps to complete the example:

1. To validate the STANDARD policy set, enter:

```
validate policysset engpoldom standard
```

Examine any messages that result and correct the problems.

2. To activate the STANDARD policy set, enter:

```
activate policysset engpoldom standard
```

Assigning client nodes to a policy domain

You can assign existing client nodes to a new policy domain, or create new client nodes to be associated with an existing policy domain.

For example, to assign the client node APPCLIENT1 to the ENGPOLDOM policy domain, issue the following command:

```
update node appclient1 domain=engpoldom
```

To create a new client node, NEWUSER, and assign it to the ENGPOLDOM policy domain, issue the following command:

```
register node newuser newuser domain=engpoldom
```

Running expiration processing to delete expired files

Expiration processing deletes expired client files from the server storage. Expiration processing also removes from the database any restartable restore sessions that exceed the time limit for saving such sessions.

You can run expiration processing either automatically or by command. You should ensure that expiration processing runs periodically to allow the server to reuse storage pool space that is occupied by expired client files.

Note:

1. A base file is not eligible for expiration until all of its dependent subfiles have been expired.
2. An archive file is not eligible for expiration if there is a deletion hold on it. If a file is not held, it will be handled according to existing expiration processing.

Related concepts

“Expiration processing of base files and subfiles” on page 513

“Deletion hold” on page 475

Running expiration processing automatically

You control automatic expiration processing by using the expiration interval option (EXPINTERVAL) in the server options file (dmserv.opt). You can also control when restartable restore sessions expire with another server option, RESTOREINTERVAL.

You can set the options by editing the dmserv.opt file (see the *Administrator's Reference*).

If you use the server options file to control automatic expiration, the server runs expiration processing each time you start the server. After that, the server runs expiration processing at the interval you specified with the option, measured from the start time of the server.

Using commands and scheduling to control expiration processing

You can manually start expiration processing.

Issue the following command:

```
expire inventory
```

Expiration processing then deletes expired files from the database. You can schedule this command by using the DEFINE SCHEDULE command. If you schedule the EXPIRE INVENTORY command, set the expiration interval to 0 (zero) in the server options so that the server does not run expiration processing when you start the server.

You can control how long the expiration process runs by using the **DURATION** parameter with the EXPIRE INVENTORY command. You can also run several (up to 10) expiration processes in parallel by specifying RESOURCE=*x*, where *x* equals the number of nodes that you want to process.

When expiration processing runs, the server normally sends detailed messages about policy changes made since the last time expiration processing ran. You can reduce those messages by using the **QUIET=YES** parameter with the EXPIRE INVENTORY command, or the following options:

- The EXPQUIET server option

When you use the quiet option or parameter, the server issues messages about policy changes during expiration processing only when files are deleted, and either the default management class or retention grace period for the domain has been used to expire the files.

Additional expiration processing with disaster recovery manager

If you have disaster recovery manager (DRM), one or more database backup volumes may also be deleted during expiration processing.



These volumes may be deleted if the following conditions are true:

- The volume has a device type of SERVER
- The volume is not part of the most recent database backup series
- The last volume of the database backup series has exceeded the expiration value specified with the SET DRMDBBACKUPEXPIREDAYS command

Related tasks

“Moving copy storage pool and active-data pool volumes on-site” on page 817

Protection and expiration of archive data

There are two separate ways to protect Tivoli Storage Manager archive objects so that they will not be inadvertently deleted. One method is to activate data retention protection on a server level.

For example, securities brokers and other regulated institutions enforce retention requirements for certain records, including electronic mail, customer statements, trade settlements, check images and new account forms. Data retention protection prevents deliberate or accidental deletion of data until its specified retention criterion is met.

Another method of additional protection is to place a deletion hold on an object using the client API. For example, federal regulatory requirements allow a broker-dealer to delete records when the regulatory retention period has lapsed, but other legal requirements might mandate that the records continue to be maintained. By using deletion hold, you ensure that data is not deleted until the hold is released.

Data retention protection

Data retention protection ensures that archive objects are not deleted from the Tivoli Storage Manager server until policy-based retention requirements for that object have been satisfied. It is enforced at the server level via the SET ARCHIVERETENTIONPROTECTION command.

See the *Administrator's Reference* for more information.

Retention protection can only be activated on a new server that does not already have stored objects (backup, archive, or space-managed). Activating retention protection applies to all archive objects subsequently stored on that server. After retention protection has been set, the server cannot store backup objects, space-managed objects, or backupsets. Retention protection cannot be added for an object that was previously stored on a Tivoli Storage Manager server. After an object is stored with retention protection, retention protection cannot be removed.

Retention protection is based on the retention criterion for each object, which is determined by the **RETVER** parameter of the archive copy group of the management class to which the object is bound. If an object uses event-based retention, the object will not expire until whatever comes later: either the date the object was archived plus the number of days in the **RETMIN** parameter or the date the event was signalled plus the number of days specified in the **RETVER** parameter. On servers which have retention protection enabled, the following operations will not delete objects whose retention criterion has not been satisfied:

- Requests from the client to delete an archive object
- DELETE FILESPACE (from either a client or administrative command)
- DELETE VOLUME DISCARDDATA=YES

- AUDIT VOLUME FIX=YES

Important: A cached copy of data can be deleted, but data in primary storage pools, copy storage pools, and active-data pools can only be marked damaged and is never deleted.

If your server has data retention protection activated, the following items are restrictions:

- A registered node cannot be reassigned to a different policy domain.
- You cannot define a device class with a device type of SERVER.
- You can export data but it will not be retention protected after it is imported.
- You cannot import data to a retention protected server.

For servers which have data retention protection enabled, consider using mirroring for the Tivoli Storage Manager database and log. You should also consider using roll forward log mode and performing frequent database backups. This way if a database restore is needed, the database can be brought back to the current point in time without loss of data.

Related tasks

Chapter 25, “Protecting and recovering your server,” on page 755

Deletion hold

If a hold is placed on an object through the client API, the object is not deleted until the hold is released.

See the *Backup-Archive Clients Installation and User's Guide* for more information. There is no limit to how often you alternate holding and releasing an object. An object can have only one hold on it at a time, so if you attempt to hold an object that is already held, you will get an error message.

If an object with event-based policy is on hold, an event can still be signalled. The hold will not extend the retention period for an object. If the retention period specified in the **RETVER** and **RETMIN** parameters expires while the object is on hold, the object will be eligible for deletion whenever the hold is released.

If an object is held, it will not be deleted whether or not data retention protection is active. If an object is not held, it is handled according to existing processing such as normal expiration, data retention protection, or event-based retention. Data that is in deletion hold status can be exported. The hold status will be preserved when the data is imported to another system.

The following deletion operations are prevented if a hold is on an object:

- Requests from the client to delete an archive object
- DELETE FILESPACE (from either a client or administrative command)
- DELETE VOLUME DISCARDDATA=YES
- AUDIT VOLUME FIX=YES

Note: A cached copy of data can be deleted, but data in primary storage pools, copy storage pools, and active-data pools can only be marked damaged and is never deleted.

Protecting data using the NetApp SnapLock licensed feature

The NetApp SnapLock licensed feature helps meet federal regulatory requirements for archived data. The SnapLock feature allows Tivoli Storage Manager to set a retention date for files and to commit a file to a WORM (write once, read many) state.

Data stored with a retention date cannot be deleted from the file system before the retention period expires. The SnapLock feature can only be used by Tivoli Storage Manager servers that have data retention protection enabled.

Data archived by data retention protection servers and stored to NetApp NAS file servers is stored as Tivoli Storage Manager FILE volumes. At the end of a write transaction, a retention date is set for the FILE volume, through the SnapLock interface. This date is calculated by using the **RETVER** and **RETMIN** parameters of the archive copy group used when archiving the data. Having a retention date associated with the FILE volume gives it a characteristic of WORM media by not allowing the data to be destroyed or overwritten until the retention date has passed. These FILE volumes are referred to as WORM FILE volumes. After a retention date has been set, the WORM FILE volume cannot be deleted until the retention date has passed. System Storage Archive Manager combined with WORM FILE volume reclamation ensures protection for the life of the data.

Storage pools can be managed either by threshold or by data retention period. The **RECLAMATIONTYPE** storage pool parameter indicates that a storage pool is managed based on a data retention period. When a traditional storage pool is queried with the **FORMAT=DETAILED** parameter, this output is displayed:

```
Reclamation Type: THRESHOLD
```

Tivoli Storage Manager servers that have data retention protection enabled through System Storage Archive Manager and have access to a NetApp filer with the SnapLock licensed feature can define a storage pool with **RECLAMATIONTYPE** set to **SNAPLOCK**. This means that data created on volumes in this storage pool are managed by retention date. When a SnapLock storage pool is queried with the **FORMAT=DETAILED** parameter, the output displayed indicates that the storage pools are managed by data retention period.

```
Reclamation Type: SNAPLOCK
```

See the NetApp document *Data ONTAP Storage Management Guide* for details on the SnapLock filer. Note this is NetApp documentation.

Attention: It is not recommended that you use this feature to protect data with a retention period of less than three months.

Related concepts

“Data retention protection” on page 474

Reclamation and the SnapLock feature

It is recommended that you set the NetApp default retention period to 30 days to match the WORM FILE default reclamation period. Tivoli Storage Manager reclaims any remaining data on a WORM FILE volume just before the retention date expiration.

The reclamation of a WORM FILE volume to another WORM FILE volume before the retention date expiration ensures that data is always protected by the SnapLock feature.

Because this protection is at a Tivoli Storage Manager volume level, the data on the volumes can be managed by Tivoli Storage Manager policy without consideration of where the data is stored. Data stored on WORM FILE volumes is protected both by data retention protection and by the retention period stored with the physical file on the SnapLock volume. If a Tivoli Storage Manager administrator issues a command to delete the data, the command fails. If someone attempt to delete the file through a series of network file system calls, the SnapLock feature prevents the data from being deleted.

During reclamation processing, if the Tivoli Storage Manager server cannot move data from an expiring SnapLock volume to a new SnapLock volume, a warning message is issued.

Retention periods

Tivoli Storage Manager policies manage the retention time for the WORM FILE volume. The retention of some files might exceed the retention time for the WORM FILE volume they were stored on. This could require moving them to another volume to ensure that the files are stored on WORM media.

Some objects on the volume might need to be retained longer than other objects on the volume for the following reasons:

- They are bound to management classes with different retention times.
- They cannot be removed because of a deletion hold.
- They are waiting for an event to occur before expiring.
- The retention period for a copy group is increased, requiring a longer retention time than that specified in the SnapLock feature when the WORM FILE volume was committed.

Use the DEFINE STGPOOL command to set up a storage pool for use with the SnapLock feature. Selecting RECLAMATIONTYPE=SNAPLOCK enables Tivoli Storage Manager to manage FILE volumes by a retention date. After a storage pool has been set up as a SnapLock storage pool, the RECLAMATIONTYPE parameter cannot be updated to THRESHOLD. When a SnapLock storage pool is defined, a check is made to ensure that the directories specified in the device class are SnapLock WORM volumes. When a file class is defined and storage pools are created with the reclamation type of SNAPLOCK, all volumes must be WORM volumes or the operation fails. If a device class is updated to contain additional directories and there are SnapLock storage pools assigned to it, the same check is made to ensure all directories are SnapLock WORM volumes.

There are three retention periods available in the NetApp SnapLock feature. These must be configured correctly so that the Tivoli Storage Manager server can properly manage WORM data stored in SnapLock volumes. The Tivoli Storage Manager server sets the retention period for data being stored on NetApp SnapLock volumes based on the values in the copy group for the data being

archived. The NetApp filer should not conflict with the ability of the Tivoli Storage Manager server to set the retention period. The following settings are the Tivoli Storage Manager recommendations for retention periods in the NetApp filer:

1. Minimum Retention Period Set the higher value: either 30 days or the minimum number of days specified by any copy group (using a NetApp SnapLock filer for WORM FILE storage) for the data retention period. The copy group is the one in use storing data on NetApp SnapLock volumes.
2. Maximum Retention Period Leave default of 30 years. This allows the Tivoli Storage Manager server to set the actual volume retention period based on the settings in the archive copy group.
3. Default Retention Period Set to 30 days. If you do not set this value and you do not set the maximum retention period, each volume's retention period will be set to 30 years. If this occurs, the Tivoli Storage Manager server's ability to manage expiration and reuse of NetApp SnapLock volumes will be largely defeated in that no volume will be able to be reused for thirty years.

With the NetApp SnapLock retention periods appropriately set, Tivoli Storage Manager can manage the data in SnapLock storage pools with maximum efficiency. For each volume that is in a SNAPLOCK storage pool, a Tivoli Storage Manager reclamation period is created. The Tivoli Storage Manager reclamation period has a start date, BEGIN RECLAIM PERIOD, and an end date, END RECLAIM PERIOD. View these dates by issuing the QUERY VOLUME command with the FORMAT=DETAILED parameter on a SnapLock volume. For example:

```
Begin Reclaim Period: 09/05/2010
End Reclaim Period: 10/06/2010
```

When Tivoli Storage Manager archives files to a SnapLock volume, it keeps track of the latest expiration date of those files, and the BEGIN RECLAIM PERIOD is set to that latest expiration date. When more files are added to the SnapLock volume, the starting date is set to that later date if there is a file with a later expiration date than the one currently on the volume. The start date is set to the latest expiration date for any file on that volume. The expectation is that all files on that volume should have already either expired, or should be expiring on that day and the following day there should be no valid data left on that volume.

The END RECLAIM PERIOD is set to a month later than the BEGIN RECLAIM PERIOD. The retention date set in the NetApp filer for that volume is set to the END RECLAIM PERIOD date. This means the NetApp filer will prevent any deletion of that volume until the END RECLAIM PERIOD date has passed. This is approximately a month after the data has actually expired in the Tivoli Storage Manager server. If an END RECLAIM PERIOD date is calculated by the Tivoli Storage Manager server for a volume, and the date is later than the current END RECLAIM PERIOD, the new date will be reset in the NetApp filer for that volume to the later date. This guarantees that the Tivoli Storage Manager WORM FILE volume will not be deleted until all data on the volume has expired, or the data has been moved to another SnapLock volume.

The Tivoli Storage Manager reclamation period is the amount of time between the begin date and the end date. It is also the time period which the Tivoli Storage Manager server has to delete volumes on which all the data has expired, or to move files which have not expired on expiring SnapLock volumes to new SnapLock volumes with new dates. This month is critical to how the server safely and efficiently manages the data on WORM FILE volumes. Data on a SnapLock volume typically expires by the time the beginning date arrives, and the volume

should be empty. When the end date arrives, the volume can be safely deleted from the Tivoli Storage Manager inventory and the SnapLock filer.

However, some events may occur which mean that there is still valid data on a SnapLock volume:

1. Expiration processing in the Tivoli Storage Manager server for that volume may have been delayed or has not completed yet.
2. The retention parameters on the copy group or associated management classes may have been altered for a file after it was archived, and that file is not going to expire for some period of time.
3. A deletion hold may have been placed on one or more of the files on the volume.
4. Reclamation processing has either been disabled or is encountering errors moving data to new SnapLock volumes on a SnapLock storage pool.
5. A file is waiting for an event to occur before the Tivoli Storage Manager server can begin the expiration of the file.

If there are files which have not expired on a SnapLock volume when the beginning date arrives, they must be moved to a new SnapLock volume with a new begin and end date. This will properly protect that data. However, if expiration processing on the Tivoli Storage Manager server has been delayed, and those files will expire as soon as expiration processing on the Tivoli Storage Manager server runs, it is inefficient to move those files to a new SnapLock volume. To ensure that unnecessary data movement does not occur for files which are due to expire, movement of files on expiring SnapLock volumes will be delayed some small number of days after the BEGIN RECLAIM PERIOD date. Since the data is protected in the SnapLock filer until the END RECLAIM PERIOD date, there is no risk to the data in delaying this movement. This allows Tivoli Storage Manager expiration processing to complete. After that number of days, if there is still valid data on an expiring SnapLock volume, it will be moved to a new SnapLock volume, thus continuing the protection of the data.

Since the data was initially archived, there may have been changes in the retention parameters for that data (for example, changes in the management class or copy pool parameters) or there may be a deletion hold on that data. However, the data on that volume will only be protected by SnapLock until the END RECLAIM PERIOD date. Data that has not expired is moved to new SnapLock volumes during the Tivoli Storage Manager reclamation period. If errors occur moving data to a new SnapLock volume, a distinct warning message is issued indicating that the data will soon be unprotected. If the error persists, it is recommended that you issue a MOVE DATA command for the problem volume.

Attention: Disabling reclamation processing on a SnapLock storage pool is not recommended because after the processing is disabled, the Tivoli Storage Manager server has no way to issue warning messages that data will become unprotected. This situation can also occur if reclamation and migration is disabled for the entire server (for example, NOMIGRRECL set in the server options file). Be very careful when managing SnapLock storage pools so that data doesn't inadvertently become unprotected.

Configuring SnapLock for event-based retention

Data stored in SnapLock volumes that are managed by System Storage Archive Manager and event-based retention can result in excessive reclamation, which causes performance degradation of the server.

If data is managed by event-based retention, Tivoli Storage Manager initially sets the retention period to the greater of the RETVER and RETMIN values for the archive copy group. When the volume enters the reclamation period and data that remains on the volume is moved, the retention period for the target volume is set to the remaining retention period of the data, which is typically 0. The new volume then enters the reclamation period shortly after receiving the data, resulting in the reclamation of volumes that were just created.

You can avoid this situation by using the RETENTIONEXTENSION server option. This option allows the server to set or extend the retention date of a SnapLock volume. You can specify from 30 to 9999 days. The default is 365 days.

When selecting volumes in a SnapLock storage pool for reclamation, the server checks if the volume is within the reclamation period.

- If the volume is not within the reclamation period, no action is taken. The volume is not reclaimed, and the retention date is unchanged
- If the volume is within the reclamation period, the server checks if the percent of reclaimable space on the volume is greater than the reclamation threshold of the storage pool or of the threshold percentage passed in on the THRESHOLD parameter of a RECLAIM STGPOOL command.
 - If the reclaimable space is greater than the threshold, the server reclaims the volume and sets the retention date of the target volume is set to the greater of these values:
 - The remaining retention time of the data plus 30 days for the reclamation period.
 - The RETENTIONEXTENSION value plus 30 days for the reclamation period.
 - If the reclaimable space is not greater than the threshold, the server resets the retention date of the volume by the amount specified in the RETENTIONEXTENSION option. The new retention period is calculated by adding the number of days specified to the current date.

In the examples described below, the SnapLock volume, VolumeA, is in a storage pool whose reclamation threshold is set to 60%. The RETENTIONEXTENSION server option is set to 365 days. The retention period VolumeA is in the reclamation period. The following situations show how retention is affected:

- The reclaimable space on VolumeA is less than 60%. The retention date of VolumeA is extended by 365 days.
- The reclaimable space on VolumeA is greater than 60%, and the remaining retention time of the data is more than 365 days. VolumeA is reclaimed, and the retention date of the target volume is set based on the remaining retention of the data plus 30 days for the reclamation period.
- The reclaimable space on VolumeA is greater than 60%, and the retention time of the data is less than 365 days. VolumeA is reclaimed, and its retention date is set to 365 days, the RETENTIONEXTENSION value, plus 30 days for the reclamation period.

Ensuring continuous data protection

Data that is stored on a volume with the SnapLock feature enabled and moved or copied to a non-SnapLock volume loses the unique hardware protection that is available through the NetApp WORM volumes.

The Tivoli Storage Manager server allows this type of movement, but if data is moved from a WORM FILE volume to another type of media, the data may no longer be protected from inadvertent or malicious deletion. If this data is on WORM volumes to meet data retention and protection requirements for certain legal purposes and is moved to other media, the data may no longer meet those requirements. You should configure your storage pools so this type of data is kept in storage pools which consist of SnapLock WORM volumes during the entire data retention period.

Set up SnapLock volumes as Tivoli Storage Manager WORM FILE volumes

When defining or updating configurations that involve SnapLock storage pools, you should ensure that the storage pools selected for the **NEXTSTGPOOL**, **RECLAIMSTGPOOL**, and **COPYSTGPOOLS** parameters have the **RECLAMATIONTYPE=SNAPLOCK** option specified.

By configuring the storage pools in this way, you ensure that your data will be properly protected. If you define a next, reclaim, copy storage pool, or active-data pool without selecting the **RECLAMATIONTYPE=SNAPLOCK** option, you will not have a protected storage pool. The command will succeed but a warning message will be issued.

Perform the following steps to set up a SnapLock volume for use as a Tivoli Storage Manager WORM FILE volume.

1. Install and set up SnapLock on the NetApp filer. See NetApp documentation for more information.
2. Properly configure the minimum, maximum, and default retention periods. If these retention periods are not configured properly, Tivoli Storage Manager will not be able to properly manage the data and volumes.
3. Install and configure a Tivoli Storage Manager server with data retention protection. Ensure the **SET ARCHIVERETENTIONPROTECTION** command is activated.
4. Set up policy by using the **DEFINE COPYGROUP** command. Select **RETVER** and **RETMIN** values in the archive copy group which will meet your requirements for protecting this data in WORM storage. If the **RETVER** or **RETMIN** values are not set, the default management classes values will be used.
5. Set up storage by using the **DEFINE DEVCLASS** command.
 - Use the **FILE** device class.
 - Specify the **DIRECTORY** parameter to point to the directory or directories on the SnapLock volumes.
6. Define a storage pool using the device class you defined above.
 - Specify **RECLAMATIONTYPE=SNAPLOCK**.
7. Update the copy group to point to the storage pool you just defined.
8. Use the Tivoli Storage Manager API to archive your objects into the SnapLock storage pool. This feature is not available on standard Tivoli Storage Manager backup-archive clients.

Related reference

“Retention periods” on page 477

Policy configuration scenarios

The scenarios that are published are designed to show you some cases for which policy changes may be needed.

Related tasks

“Configuring policy for direct-to-tape backups”

Related reference

“Configuring policy for Tivoli Storage Manager application clients” on page 483

“Policy for logical volume backups” on page 483

“Configuring policy for NDMP operations” on page 485

“Configuring policy for LAN-free data movement” on page 486

“Policy for Tivoli Storage Manager servers as clients” on page 488

“Setting policy to enable point-in-time restore for clients” on page 488

Configuring policy for direct-to-tape backups

The server default policy enables client nodes to back up data to disk storage pools on the server. As an alternative, you may configure a policy to store client data directly in tape storage pools to reduce contention for disk resources.

If you back up directly to tape, the number of clients that can back up data at the same time is equal to the number of drives available to the storage pool (through the mount limit of the device class). For example, if you have one drive, only one client at a time can back up data.

The direct-to-tape backup eliminates the need to migrate data from disk to tape. However, performance of tape drives is often lower when backing up directly to tape than when backing up to disk and then migrating to tape. Backing up data directly to tape usually means more starting and stopping of the tape drive. Backing up to disk then migrating to tape usually means the tape drive moves more continuously, meaning better performance.

1. Double-click the desktop icon for the Tivoli Storage Manager Console.
2. Expand the tree until the Tivoli Storage Manager server you want to work with is displayed. Expand the server and click **Wizards**. The list of wizards appears in the right pane.
3. Select the Client Node Configuration wizard and click **Start**. The Client Node Configuration wizard appears.
4. Progress through the wizard to the “Define Tivoli Storage Manager client nodes and policy” page.
5. By default, client nodes are associated with BACKUPPOOL. This storage pool is set to immediately migrate any data it receives. Drag BACKUPPOOL and drop it on a tape storage pool.

Note: You can also select a client, click **Edit** → **New** to create a new policy domain that will send client data directly to any storage pool.

6. Finish the wizard.

At the server command line, you may define a new policy domain that enables client nodes to back up or archive data directly to tape storage pools. For example, you may define a policy domain named DIR2TAPE with the following steps:

1. Copy the default policy domain STANDARD as a template:

```
copy domain standard dir2tape
```

This command creates the DIR2TAPE policy domain that contains a default policy set, management class, backup and archive copy group, each named STANDARD.

2. Update the backup or archive copy group in the DIR2TAPE policy domain to specify the destination to be a tape storage pool. For example, to use a tape storage pool named TAPEPOOL for backup, issue the following command:

```
update copygroup dir2tape standard standard destination=tapepool
```

To use a tape storage pool named TAPEPOOL for archive, issue the following command:

```
update copygroup dir2tape standard standard type=archive
destination=tapepool
```

3. Activate the changed policy set.

```
activate policysset dir2tape standard
```

4. Assign client nodes to the DIR2TAPE policy domain. For example, to assign a client node named TAPEUSER1 to the DIR2TAPE policy domain, issue the following command:

```
update node tapeuser1 domain=dir2tape
```

Configuring policy for Tivoli Storage Manager application clients

The Tivoli Storage Manager application clients using the server to store data may require that you configure policy to make the most efficient use of server storage. See the user's guide for each application client for policy requirements.

Some of the application clients include a time stamp in each database backup. Because the default policy for the server keeps one backup version of each unique file, database backups managed by default policy are never deleted because each backup is uniquely named with its time stamp. To ensure that the server deletes backups as required, configure policy as recommended in the user's guide for the application client.

Policy for logical volume backups

Consider defining a management class specifically for logical volume backups. To enable clients to restore a logical volume and then reconcile the results of any file backup operations since the logical volume backup was made, you must set up management classes with the backup copy group set up differently from the STANDARD.

The **Versions Data Exists**, **Versions Data Deleted**, and **Retain Extra Versions** parameters work together to determine over what time period a client can restore a logical volume image and reconcile later file backups. Also, you may have server storage constraints that require you to control the number of backup versions allowed for logical volumes. The server handles logical volume backups the same as regular incremental or selective backups. Logical volume backups differ from selective, incremental, or archive operations in that each file space that is backed up is treated as a single large file.

Backups of logical volumes are intended to help speed the restoration of a computer. One way to use the capability is to have users periodically (for example, once a month) perform a logical volume backup, and schedule daily full incremental backups. If a user restores a logical volume, the program first restores the logical volume backup and then any files that were changed since the backup (incremental or other file backup processes). The user can also specify that the restore process reconcile any discrepancies that can result when files are deleted.

For example, a user backs up a logical volume, and the following week deletes one or more files from the volume. At the next incremental backup, the server records in its database that the files were deleted from the client. When the user restores the logical volume, the program can recognize that files have been deleted since the backup was created. The program can delete the files as part of the restore process. To ensure that users can use the capability to reconcile later incremental backups with a restored logical volume, you need to ensure that you coordinate policy for incremental backups with policy for backups for logical volumes.

For example, you decide to ensure that clients can choose to restore files and logical volumes from any time in the previous 60 days. You can create two management classes, one for files and one for logical volumes. Table 49 shows the relevant parameters. In the backup copy group of both management classes, set the Retain Extra Versions parameter to 60 days.

In the management class for files, set the parameters so that the server keeps versions based on age rather than how many versions exist. More than one backup version of a file may be stored per day if clients perform selective backups or if clients perform incremental backups more than once a day. The **Versions Data Exists** parameter and the **Versions Data Deleted** parameter control how many of these versions are kept by the server. To ensure that any number of backup versions are kept for the required 60 days, set both the **Versions Data Exists** parameter and the **Versions Data Deleted** parameter to NOLIMIT for the management class for files. This means that the server retains backup versions based on how old the versions are, instead of how many backup versions of the same file exist.

For logical volume backups, the server ignores the frequency attribute in the backup copy group.

Table 49. Example of backup policy for files and logical volumes

Parameter (backup copy group in the management class)	Management Class for Files	Management Class for Logical Volumes
Versions Data Exists	NOLIMIT	3 versions
Versions Data Deleted	NOLIMIT	1
Retain Extra Versions	60 days	60 days
Retain Only Version	120 days	120 days

Configuring policy for NDMP operations

You can register a network-attached storage (NAS) file server as a node, using network data management protocol (NDMP) operations. Under the direction of the Tivoli Storage Manager server, the NAS file server performs backup and restore of file system and directory images to a tape library.

The Tivoli Storage Manager server initiates the backup, allocates a drive, and selects and mounts the media. The NAS file server then transfers the data to tape.

Because the NAS file server performs the backup, the data is stored in its own format. For most NAS file servers, the data is stored in the NDMPDUMP data format. For NetApp file servers, the data is stored in the NETAPPDUMP data format. For EMC file servers, the data is stored in the CELERRADUMP data format. To manage NAS file server image backups, copy groups for NAS nodes must point to a storage pool that has a data format of NDMPDUMP, NETAPPDUMP, or CELERRADUMP.

The following backup copy group attributes are ignored for NAS images:

- Frequency
- Mode
- Retain Only Versions
- Serialization
- Versions Data Deleted

To set up the required policy for NAS nodes, you can define a new, separate policy domain.

Backups for NAS nodes can be initiated from the server, or from a client that has at least client owner authority over the NAS node. For client-initiated backups, you can use client option sets that contain include and exclude statements to bind NAS file system or directory images to a specific management class. The valid options that can be used for a NAS node are: include.fs.nas, exclude.fs.nas, and domain.nas. NAS backups initiated from the Tivoli Storage Manager server with the BACKUP NODE command ignore client options specified in option files or client option sets. For details on the options see the *Backup-Archive Clients Installation and User's Guide* for your particular client platform.

When the Tivoli Storage Manager server creates a table of contents (TOC), you can view a collection of individual files and directories backed up via NDMP and select which to restore. To establish where to send data and store the table of contents, policy should be set so that:

- Image backup data is sent to a storage pool with a NDMPDUMP, NETAPPDUMP or CELERRADUMP format.
- The table of contents is sent to a storage pool with either NATIVE or NONBLOCK format.

Related tasks

“Creating client option sets on the server” on page 411

Related reference

Chapter 8, “Using NDMP for operations with NAS file servers,” on page 167

Configuring policy for LAN-free data movement

For LAN-free data movement, you can set up a SAN configuration in which a client directly accesses a storage device to read or write data. LAN-free data movement requires setup on the server and on the client, and the installation of a storage agent on the client computer.

The storage agent transfers data between the client and the storage device. See *Storage Agent User's Guide* for details. See the Web site for details on clients that support the feature: http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

One task in configuring your systems to use this feature is to set up policy for the clients. Copy groups for these clients must point to the storage pool that is associated with the SAN devices. If you have defined a path from the client to a drive on the SAN, drives in this storage pool can then use the SAN to send data directly to the device for backup, archive, restore, and retrieve.

To set up the required policy, either define a new, separate policy domain, or define a new management class in an existing policy domain.

Related tasks

"Define a new policy domain"

"Configuring IBM Tivoli Storage Manager for LAN-free data movement" on page 124

Related reference

"Define a new management class in an existing policy domain" on page 487

Define a new policy domain

One way to configure policy for clients is to define a separate policy domain in which the active policy set has a default management class with the required settings. Then register all clients using SAN data transfer to that domain.

Perform the following steps to define a new policy domain:

1. Create the policy domain for the clients. For example, to define a policy domain that is named SANCLIENTS, issue the following command:

```
define domain sanclients
  description='Policy domain for clients using SAN devices'
```

2. Create a policy set in that domain. For example, to define the policy set that is named BASE in the SANCLIENTS policy domain, issue the following command:

```
define policyset sanclients base
```

3. Create the default management class for the policy set. First define the management class, then assign the management class as the default for the policy set.

For example, to define the management class that is named SANCLIENTMC, issue the following command:

```
define mgmtclass sanclients base sanclientmc
```

Then assign the new management class as the default:

```
assign defmgmtclass sanclients base sanclientmc
```

4. Define the backup copy group in the default management class, as follows:
 - Specify the DESTINATION, the name of the storage pool that is associated with the SAN devices on the server.

The storage pool must already be set up. The storage pool must point to a device class that is associated with the library for the SAN devices.

- Accept the default settings for all remaining parameters.

For example, to define the backup copy group for the SANCLIENTMC management class, issue the following command:

```
define copygroup sanclients base sanclientmc standard destination=sanpool
```

5. Define the archive copy group in the default management class, as follows:

- Specify the DESTINATION, the name of the storage pool that is associated with the SAN devices on the server.

The storage pool must already be set up. The storage pool must point to a device class that is associated with the library for the SAN devices.

- Accept the default settings for all remaining parameters.

For example, to define the archive copy group for the SANCLIENTMC management class, issue the following command:

```
define copygroup sanclients base sanclientmc standard  
type=archive destination=sanpool
```

6. Activate the policy set.

For example, to activate the BASE policy set in the SANCLIENTS policy domain, issue the following command:

```
activate policyset sanclients base
```

7. Register or update the application clients to associate them with the new policy domain.

For example, to update the node SANCLIENT1, issue the following command:

```
update node sanclient1 domain=sanclients
```

Related tasks

“Configuring IBM Tivoli Storage Manager for LAN-free data movement” on page 124

Define a new management class in an existing policy domain

If you choose not to define a separate policy domain with the appropriate management class as the default, you must define a new management class within an existing policy domain and activate the policy set.

Because the new management class is not the default for the policy domain, you must add an include statement to each client options file to bind objects to that management class.

For example, suppose `sanclientmc` is the name of the management class that you defined for clients that are using devices on a SAN. You want the client to be able to use the SAN for backing up any file on the `c` drive. Put the following line at the end of the client's include-exclude list:

```
include c:* sanclientmc
```

For details on the include-exclude list, see *Backup-Archive Clients Installation and User's Guide*.

Policy for Tivoli Storage Manager servers as clients

One server (a source server) can be registered as a client to another server (the target server). Data stored by the source server appears as archived files on the target server. The source server is registered to a policy domain on the target server, and uses the default management class for that policy domain.

In the default management class, the destination for the archive copy group determines where the target server stores data for the source server. Other policy specifications, such as how long to retain the data, do not apply to data stored for a source server.

Related tasks

“Using virtual volumes to store data on another server” on page 712

Setting policy to enable point-in-time restore for clients

To enable clients to restore backed-up files to a specific point in time, you must set up the backup copy group differently from the STANDARD. The **Versions Data Exists**, **Versions Data Deleted**, and **Retain Extra Versions** parameters work together to determine over what time period a client can perform a point-in-time restore operation.

For example, you decide to ensure that clients can choose to restore files from anytime in the previous 60 days. In the backup copy group, set the **Retain Extra Versions** parameter to 60 days. More than one backup version of a file may be stored per day if clients perform selective backups or if clients perform incremental backups more than once a day. The **Versions Data Exists** parameter and the **Versions Data Deleted** parameter control how many of these versions are kept by the server. To ensure that any number of backup versions are kept for the required 60 days, set both the **Versions Data Exists** parameter and the **Versions Data Deleted** parameter to NOLIMIT. This means that the server essentially determines the backup versions to keep based on how old the versions are, instead of how many backup versions of the same file exist.

Keeping backed-up versions of files long enough to allow clients to restore their data to a point in time can mean increased resource costs. Requirements for server storage increase because more file versions are kept, and the size of the server database increases to track all of the file versions. Because of these increased costs, you may want to choose carefully which clients can use the policy that allows for point-in-time restore operations.

Clients need to run full incremental backup operations frequently enough so that IBM Tivoli Storage Manager can detect files that have been deleted on the client file system. Only a full incremental backup can detect whether files have been deleted since the last backup. If full incremental backup is not done often enough, clients who restore to a specific time may find that many files that had actually been deleted from the workstation get restored. As a result, a client's file system may run out of space during a restore process.

Important: The server will not attempt to retrieve client files from an active-data pool during a point-in-time restore. Point-in-time restores require both active and inactive file versions. Active-data pools contain only active file versions. For optimal efficiency during point-in-time restores and to avoid switching between active-data pools and primary or copy storage pools, the server retrieves both active and inactive versions from the same storage pool and volumes.

Distributing policy using enterprise configuration

If you set up one Tivoli Storage Manager server as a configuration manager, you can distribute policy to other Tivoli Storage Manager servers.

To distribute policy, you associate a policy domain with a profile. Managed servers that subscribe to the profile then receive the following definitions:

- The policy domain itself
- Policy sets in that domain, except for the ACTIVE policy set
- Management classes in the policy sets
- Backup and archive copy groups in the management classes
- Client schedules associated with the policy domain

The names of client nodes and client-schedule associations are not distributed. The ACTIVE policy set is also not distributed.

The distributed policy becomes managed objects (policy domain, policy sets, management classes, and so on) defined in the database of each managed server. To use the managed policy, you must activate a policy set on each managed server. If storage pools specified as destinations in the policy do not exist on the managed server, you receive messages pointing out the problem when you activate the policy set. You can create new storage pools to match the names in the policy set, or you can rename existing storage pools.

On the managed server you also must associate client nodes with the managed policy domain and associate client nodes with schedules.

Related tasks

“Setting up enterprise configurations” on page 685

Querying policy

You can request information about the contents of policy objects. You might want to do this before creating new objects or when helping users to choose policies that fit their needs.

Task	Required Privilege Class
Query any policy domain, policy set, management class, or copy group	Any administrator

You can specify the output of a query in either standard or detailed format. The examples in this section are in standard format.

On a managed server, you can see whether the definitions are managed objects. Request the detailed format in the query and check the contents of the “Last update by (administrator)” field. For managed objects, this field contains the string `$$CONFIG_MANAGER$$`.

Querying copy groups

You can request information about backup copy groups through a command.

Issue the following command to request information about the backup copy group (the default) in the ENGPOLDOM engineering policy domain:

```
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

To request information about archive copy groups in the ENGPOLDOM engineering policy domain, enter:

```
query copygroup engpoldom * type=archive
```

The following data shows the output from the query.

Policy Domain Name	Policy Set Name	Mgmt Class Name	Copy Group Name	Retain Version
ENGPOLDOM	ACTIVE	MCENG	STANDARD	730
ENGPOLDOM	ACTIVE	STANDARD	STANDARD	365
ENGPOLDOM	STANDARD	MCENG	STANDARD	730
ENGPOLDOM	STANDARD	STANDARD	STANDARD	365
ENGPOLDOM	TEST	STANDARD	STANDARD	365

Querying management classes

You can request information about management classes through a command.

Issue the following command to request information about management classes in the ENGPOLDOM engineering policy domain:

```
query mgmtclass engpoldom * *
```

The following figure is the output from the query. It shows that the ACTIVE policy set contains the MCENG and STANDARD management classes.

Policy Domain Name	Policy Set Name	Mgmt Class Name	Default Mgmt Class ?	Description
ENGPOLDOM	ACTIVE	MCENG	No	Engineering Management Class with Backup and Archive Copy Groups
ENGPOLDOM	ACTIVE	STANDARD	Yes	Installed default management class
ENGPOLDOM	STANDARD	MCENG	No	Engineering Management Class with Backup and Archive Copy Groups
ENGPOLDOM	STANDARD	STANDARD	Yes	Installed default management class
ENGPOLDOM	TEST	STANDARD	Yes	Installed default management class

Querying policy sets

You can request information about policy sets through a command.

Issue the following command to request information about policy sets in the ENGPOLDOM engineering policy domain:

```
query policysset engpoldom *
```

The following figure is the output from the query. It shows an ACTIVE policy set and two inactive policy sets, STANDARD and TEST.

Policy Domain Name	Policy Set Name	Default Mgmt Class Name	Description
ENGPOLDOM	ACTIVE	STANDARD	Installed default policy set
ENGPOLDOM	STANDARD	STANDARD	Installed default policy set
ENGPOLDOM	TEST	STANDARD	Policy set for testing

Querying policy domains

You can request information about policy domains through a command.

Issue the following command to request information about a policy domain (for example, to determine if any client nodes are registered to that policy domain):

```
query domain *
```

The following figure is the output from the query. It shows that both the ENGPOLDOM and STANDARD policy domains have client nodes assigned to them.

Policy Domain Name	Activated Policy Set	Activated Default Mgmt Class	Number of Registered Nodes	Description
-----	-----	-----	-----	-----
APPLIEN-TS	BASE	APPLIEN-TMC	1	Policy domain for application clients
ENGPOLDOM	STANDARD	STANDARD	21	Engineering Policy Domain
STANDARD	STANDARD	STANDARD	18	Installed default policy domain.

Deleting policy

When you delete a policy object, you also delete any objects belonging to it. For example, when you delete a management class, you also delete the copy groups in it.

You cannot delete the ACTIVE policy set or objects that are part of that policy set.

Task	Required Privilege Class
Delete policy domains	System
Delete any policy sets, management classes, or copy groups	System or unrestricted policy
Delete policy sets, management classes, or copy groups that belong to policy domains over which you have authority	Restricted policy

You can delete the policy objects named STANDARD that come with the server. However, all STANDARD policy objects are restored whenever you reinstall the server.

Related concepts

“Protection and expiration of archive data” on page 474

Deleting copy groups

You can delete a backup or archive copy group if it does not belong to a management class in the ACTIVE policy set.

For example, to delete the backup and archive copy groups belonging to the MCENG and STANDARD management classes in the STANDARD policy set, enter:

```
delete copygroup engpoldom standard mceng type=backup
delete copygroup engpoldom standard standard type=backup
delete copygroup engpoldom standard mceng type=archive
delete copygroup engpoldom standard standard type=archive
```

Deleting management classes

You can delete a management class if it does not belong to the ACTIVE policy set.

For example, to delete the MCENG and STANDARD management classes from the STANDARD policy set, enter:

```
delete mgmtclass engpoldom standard mceng
delete mgmtclass engpoldom standard standard
```

When you delete a management class from a policy set, the server deletes the management class and all copy groups that belong to the management class in the specified policy domain.

Deleting policy sets

Authorized administrators can delete any policy set other than the ACTIVE policy set.

For example, to delete the TEST policy set from the ENGPOLDOM policy domain, enter:

```
delete policyset engpoldom test
```

When you delete a policy set, the server deletes all management classes and copy groups that belong to the policy set within the specified policy domain.

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.

Deleting policy domains

You can delete a policy domain only if the domain has no client nodes registered to it. 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.

For example, to delete the STANDARD policy domain, perform the following steps:

1. Request a list of all client nodes assigned to the STANDARD policy domain by issuing the following command:

```
query node * domain=standard
```

2. If client nodes are assigned to the policy domain, remove them in one of the following ways:

- a. Assign each client node to a new policy domain. For example, enter the following commands:

```
update node htang domain=engpoldom
update node tomc domain=engpoldom
update node pease domain=engpoldom
```

If the ACTIVE policy set in ENGPOLDOM does not have the same management class names as in the ACTIVE policy set of the STANDARD policy domain, then backup versions of files may be bound to a different management class name.

- b. Delete each node from the STANDARD policy domain by first deleting all file spaces belonging to the nodes, then deleting the nodes.
3. Delete the policy domain by issuing the following command:
`delete domain standard`

When you delete a policy domain, the server deletes the policy domain and all policy sets (including the ACTIVE policy set), management classes, and copy groups that belong to the policy domain.

Related reference

“How files and directories are associated with a management class” on page 449

Chapter 15. Managing data for client nodes

You might need help to generate backup sets and enable subfile backups for client nodes. You can also use data validation for client nodes so that any data corruption is identified when data is sent over the network between the client and server.

Tasks:
“Validating a node's data during a client session” on page 496
“Encrypting data on tape” on page 496
“Setting up shredding” on page 500
“Generating client backup sets on the server” on page 503
“Restoring backup sets from a backup-archive client” on page 507
“Moving backup sets to other servers” on page 508
“Managing client backup sets” on page 509
“Enabling clients to use subfile backup” on page 511
“Optimizing restore operations for clients” on page 514
“Managing storage usage for archives” on page 521
Concepts:
“Performance considerations for data validation” on page 496
“Securing sensitive client data” on page 499
“Creating and using client backup sets” on page 502

Validating a node's data

Data validation can identify data corruption during a client session when data is sent between a client and the server. IBM Tivoli Storage Manager provides the option of specifying whether a cyclic redundancy check (CRC) is performed during a client session to validate data sent over the network between a client or a storage agent and the server.

Cyclic redundancy checking is performed at the client when the client requests services from the server. For example, the client issues a query, backup, or archive request. The server also performs a CRC operation on the data sent by the client and compares its value with the value calculated by the client. If the CRC values do not match, the server will issue an error message once per session. Depending on the operation, the client may attempt to automatically retry the operation.

After Tivoli Storage Manager completes the data validation, the client and server discard the CRC values generated in the current session.

Data validation can be enabled for one or all of the following items:

- Tivoli Storage Manager client nodes.
- Tivoli Storage Manager storage agents. For details, refer to the storage agent user's guide for your particular operating system.

See “Choosing when to enable data validation” on page 785 to help you determine where to enable data validation.

Performance considerations for data validation

Consider the impact on performance when you decide whether data validation is necessary for all nodes or some nodes. Data validation impacts performance because additional CPU overhead is required on both the client and server to calculate and compare CRC values.

This type of validation is independent from validating data written to a storage pool volume. See “Data validation during audit volume processing” on page 784.

Validating a node's data during a client session

You can enable data validation for a node by using either the REGISTER NODE or UPDATE NODE command. By default, data validation is set to NO.

Methods for enabling data validation for a node include choosing data validation for individual nodes, specifying a set of nodes by using a wildcard search string, or specifying a group of nodes in a policy domain.

For example, to enable data validation for existing node, ED, you can issue an UPDATE NODE command. This user backs up the company payroll records weekly and you have decided it is necessary to have all the user data validated: the data itself and metadata.

```
update node ed validateprotocol=all
```

Later, the network has shown to be stable and no data corruption has been identified when user ED has processed backups. You can then disable data validation to minimize the performance impact of validating all of ED's data during a client session. For example:

```
update node ed validateprotocol=no
```

Encrypting data on tape

It is often critical to secure client data, especially when that data might be of a sensitive nature. To ensure that data for off-site volumes is protected, IBM tape encryption technology is available.

This technology uses a stronger level of encryption by requiring 256-bit Advanced Encryption Standard (AES) encryption keys. Keys are passed to the drive by a key manager to encrypt and decrypt data.

IBM tape technology supports three different methods of drive encryption for IBM 3592 generation 2 and generation 3 devices, IBM linear tape open (LTO) generation 4 devices, and Sun StorageTek T10000B devices.

Restriction: The only method of drive encryption that is available for an HP LTO-4 device is application.

Application encryption

Encryption keys are managed by the application, in this case, Tivoli Storage Manager. Tivoli Storage Manager generates and stores the keys in the server database. Data is encrypted during WRITE operations, when the encryption key is passed from the server to the drive. Data is decrypted for READ operations.

Attention: When using application encryption, you must take extra care to secure database backups because the encryption keys that are used to encrypt and decrypt data are stored in the server database. To restore your data, you must have the correct database backup and corresponding encryption keys to access your information. Ensure that you back up the database frequently and safeguard the backups to prevent data loss or theft. Anyone who has access to both the database backup and the encryption keys has access to your data.

Use application-managed encryption for only storage pool volumes. Other volumes such as backup-set tapes, export volumes, and database backups are not encrypted using the application method.

Library encryption

Encryption keys are managed by the library. Keys are stored in an encryption key manager and provided to the drive. If you set up the hardware to use the library encryption, you can use this method by setting the **DRIVEENCRYPTION** parameter in the device class definition to **ALLOW**.

Restriction: Only certain IBM libraries support IBM LTO-4 library encryption.

System encryption

System encryption is available on AIX. Encryption keys that are provided to the drive are managed by the device driver or operating system and stored in an encryption key manager. If the hardware is set up to use system encryption, you can use this method by setting the **DRIVEENCRYPTION** parameter in the device class definition to **ALLOW**.

The methods of drive encryption that you can use with Tivoli Storage Manager are set up at the hardware level. Tivoli Storage Manager cannot control or change which encryption method is used in the hardware configuration. If the hardware is set up for the application encryption method, Tivoli Storage Manager can turn encryption on or off depending on the **DRIVEENCRYPTION** value on the device class. For more information about specifying this parameter, see the following topics:

- “Encrypting data with 3592 generation 2 and generation 3 drives” on page 206
- “Encrypting data using LTO generation 4 tape drives” on page 213
- “Enabling ECARTRIDGE drive encryption” on page 216 and “Disabling ECARTRIDGE drive encryption” on page 216

Choosing an encryption method

Deciding on which encryption method you want to use depends on how you want to manage your data. If you only want to encrypt storage pool volumes and eliminate some encryption processing on your system, the Application method should be enabled.

This method allows Tivoli Storage Manager to manage the encryption keys. When using Application encryption, you must take extra care to secure database backups since the encryption keys are stored in the server database. Without access to database backups and matching encryption keys, you will not be able to restore your data.

If you want to encrypt all of your data in a particular logical library or encrypt data on more than just storage pool volumes, the System or Library method can be

utilized. These methods are virtually transparent to the server. Tivoli Storage Manager is aware of them being used and displays informational messages when writing to an empty volume.

Library managed encryption allows you to control which volumes are encrypted through the use of their serial numbers. You can specify a range or set of volumes to encrypt. With Application managed encryption, you can create dedicated storage pools that only contain encrypted volumes. This way, you can utilize storage pool hierarchies and policies to manage the way data is encrypted.

The Library and System methods of encryption can share the same encryption key manager, which allows the two modes to be interchanged. However, this can only occur if the encryption key manager is set up to share keys. Tivoli Storage Manager cannot currently verify if encryption key managers for both methods are the same. Neither can Tivoli Storage Manager share or utilize encryption keys between the application method and either library or system methods of encryption.

To determine whether or not a volume is encrypted and which method was used, you can issue the `QUERY VOLUME` command with `FORMAT=DETAILED`. For more information on data encryption using the backup-archive client, see the *Backup-Archive Clients Installation and User's Guide*.

Changing your encryption method and hardware configuration

If you want to change the encryption method for a given set of volumes, the volumes need to be returned to scratch status. Updating the parameter value will only affect empty volumes.

For example, if you currently have Application managed encryption enabled, and you decide that you don't want encryption enabled at all, only empty volumes will be impacted by the change. Filling volumes will continue to be encrypted while new volumes will not. If you do not want currently filling volumes to continue being encrypted, the volume status should be changed to `READONLY`. This will ensure that Tivoli Storage Manager does not append any more encrypted data to the volumes. You can use the `MOVE DATA` command to transfer the data to a new volume after the update of the `DRIVEENCRYPTION` parameter. The data will then be available in an un-encrypted format.

When migrating from one hardware configuration to another, you will need to move your data from the old volumes to new volumes with new encryption keys and key managers. You can do this by setting up two logical libraries and storage pools (each with a different encryption method) and migrating the data from the old volumes to the new volumes. This will eliminate volumes that were encrypted using the original method. Assume that you have volumes that were encrypted using the Library method and you want to migrate to the Application method. Tivoli Storage Manager will be unable to determine which encryption keys are needed for data on these volumes because the library's encryption key manager stores these keys and Tivoli Storage Manager does not have access to them. Table 50 on page 499 illustrates considerations for changing your hardware encryption method.

Table 50. Hardware and encryption method compatibility

	Volumes with No Encryption	Volumes with Application Managed Encryption	Volumes with Library Managed Encryption	Volumes with System Managed Encryption
Desired Hardware Method = None	No Special Consideration	Incompatible. Scratch tape labels will be unreadable and need to be relabeled	Incompatible. Scratch tape labels will be unreadable and need to be relabeled	Incompatible. Scratch tape labels will be unreadable and need to be relabeled
Desired Hardware Method = Application	No Special Consideration	No Special Consideration	Incompatible	Incompatible
Desired Hardware Method = Library	No Special Consideration	Incompatible	No Special Consideration	Ensure the same key bank/server is still used
Desired Hardware Method = System	No Special Consideration	Incompatible	Ensure the same key bank/server is still used	No Special Consideration

Securing sensitive client data

After client data has been deleted, it might still be possible to recover it. For sensitive data, this condition is a potential security exposure. The destruction of deleted data, also known as shredding, lets you store sensitive data so that it is overwritten one or more times after it is deleted.

This process increases the difficulty of discovering and reconstructing the data later. Tivoli Storage Manager performs shredding only on data in random-access disk storage pools. You can configure the server to ensure that sensitive data is stored only in storage pools in which shredding is enforced (shred pools).

Shredding occurs only after a data deletion commits, but it is not necessarily completed immediately after the deletion. The space occupied by the data to be shredded remains occupied while the shredding takes place, and is not available as free space for new data until the shredding is complete. When sensitive data is written to server storage and the write operation fails, the data that was already written is shredded.

Shredding performance is affected by the amount of data to be shredded, the number of times that data is to be overwritten, and the speed of the disk and server hardware. You can specify that the data is to be overwritten up to ten times. The greater the number of times, the greater the security but also the greater the impact on server performance. It is strongly recommended that write caching be disabled for any disk devices used to store sensitive data. If write caching is enabled, the overwrite operations are adversely affected.

Shredding can be done either automatically after the data is deleted or manually by command. The advantage of automatic shredding is that it is performed without administrator intervention whenever deletion of data occurs. This limits the time that sensitive data might be compromised. Automatic shredding also

limits the time that the space used by deleted data is occupied. The advantage of manual shredding is that it can be performed when it will not interfere with other server operations.

Setting up shredding

You must configure Tivoli Storage Manager so that data identified as sensitive is stored only in storage pools that will enforce shredding after that data is deleted.

Perform the following steps to set up your shredding configuration:

1. Specify that you want data to be shredded either automatically after it is deleted or manually by an administrator. You can specify how shredding is to be done by setting the SHREDDING server option.

```
shredding automatic
```

You can also set the shredding option dynamically by using the SETOPT command.

2. Set up one or more random access disk storage pool hierarchies that will enforce shredding and specify how many times the data is to be overwritten after deletion. For example,

```
define stgpool shred2 disk shred=5
define stgpool shred1 disk nextstgpool=shred2 shred=5
```

3. Define volumes to those pools, and specify disks for which write caching can be disabled.

```
define volume shred1
/var/storage/bf.dsm formatsize=100
define volume shred2
/var/storage/bg.dsm formatsize=100
```

4. Define and activate a policy for the sensitive data. The policy will bind the data to a management class whose copy groups specify shred storage pools.

```
define domain shreedom
define policyset shreedom shredpol
define mgmtclass shreedom shredpol shredclass
define copygroup shreedom shredpol shredclass type=backup
destination=shred1
define copygroup shreedom shredpol shredclass type=archive
destination=shred1
activate policyset shreedom shredpol
```

5. Identify those client nodes whose data should be shredded after deletion, and assign them to the new domain.

```
update node engineering12 domain=shreedom
```

If you have specified manual shredding with the SHREDDING server option, you can start the shredding process by issuing the SHRED DATA command. This command lets you specify how long the process will run before it is canceled and how the process responds to an I/O error during shredding. For objects that cannot be shredded, the server reports each object.

Note: 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.

To see the status and amount of data waiting to be shredded, you can issue the `QUERY SHREDSTATUS` command. The server reports a summary of the number and size of objects waiting to be shredded. To display detailed information about data shredding on the server, issuing the following command:

```
query shredstatus format=detailed
```

Figure 71 displays a detailed report for the storage pool.

Shredding	Objects	Occupied	Writes to
Active	Awaiting	Space	Complete
-----	Shred	(MB)	Shred (MB)
NO	4	182	364

Figure 71. Querying shredding status

When data shredding completes, a message is issued that reports the amount of data that was successfully shredded and the amount of data that was skipped, if any.

Ensuring that shredding is enforced

It is important to ensure that sensitive data is stored only in shred storage pools. Only data that is in a shred pool is shredded after being deleted.

Some changes to objects and some server operations involving the moving or copying of data could result in sensitive data that cannot be shredded. This would compromise the intent and value of shredding.

Table 51 describes these types of operations.

Table 51. Commands affecting the shredding process

Command	Operation
BACKUP STGPOOL	To back up a shred pool to a copy storage pool you must set the SHREDTONOSHRED parameter to YES. If this value is not specified, the server issues an error message and does not allow the backup. If this value is specified, the server does not issue a warning message when the shred pool is backed up, and that data cannot be shredded.
COPY ACTIVATEDATA	To copy data from a shred pool to an active-data pool you must set the SHREDTONOSHRED parameter to YES. If this value is not specified, the server issues an error message and does not allow the data to be copied. If this value is specified, the server does not issue a warning when data from the shred pool is copied, and that data cannot be shredded.
DEFINE STGPOOL	Tivoli Storage Manager does not require that the next storage pool for a shred pool also be a shred pool. When you define a storage pool and you specify a non-shred pool as the next storage pool, a warning message is issued, but you can choose to continue with the definition. After the storage pool is defined and migration occurs, no message is issued. However, the data in the next storage pool cannot be shredded.

Table 51. Commands affecting the shredding process (continued)

Command	Operation
EXPIRE INVENTORY DELETE FILESPACE DELETE VOLUME	Data in a shred pool is deleted after inventory expiration and after deletion of a file space or volume. After the data is deleted, it is shredded. However, if data is deleted that has copies in both shred and non-shred pools, the server shreds only those copies in the shred pools and does not issue a warning message before deletion. The data non-shred pools cannot be shredded.
EXPORT NODE EXPORT SERVER	To export data from a shred pool you must set the ALLOWSHREDDABLE parameter to YES. If this value is specified, and the exported data includes data from shred pools, that data cannot be shredded. If the export operation includes data from shred pools, the server does not issue a warning message.
GENERATE BACKUPSET	To include data from a shred pool when you generate a backup set, you must set the ALLOWSHREDDABLE parameter to YES. If this value is specified and the backup set includes data from shred pools, that data cannot be shredded, and no warning message is issued.
MOVE DATA	You cannot move data from a shred pool to a destination that is not another shred pool unless you set the SHREDTONOSHRED parameter to YES. When the move is complete, the original data is shredded but the data in the destination storage pool cannot be shredded.
UPDATE STGPOOL	The server issues a warning message for any of the storage pool updates listed below. You can choose to continue with the operation, but the result is that some or all sensitive data cannot be shredded. <ul style="list-style-type: none"> • For a shred pool, add or change a non-shred pool as the next pool in the hierarchy. This includes copy storage pools and active data storage pools. • Change a pool from non-shred to shred when the NEXTSTGPOOL attribute specifies a non-shred pool. • Change a pool with COPYSTGPools or ACTIVEDATAPOOLS attributes from non-shred to shred. • Change a pool from shred to non-shred by setting the SHRED parameter to 0.

Creating and using client backup sets

A backup set is a collection of backed-up data from one client, stored and managed as a single object on specific media in server storage. 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 the GENERATE BACKUPSET command, and consolidates them onto sequential media.

Currently, the backup object types supported for backup sets include directories, files, and image data. If you are upgrading from Tivoli Storage Manager Express®, backup sets can also contain data from Data Protection for Microsoft SQL and Data Protection for Microsoft Exchange servers. The backup set process is also called instant archive.

You can generate backup sets on the server for individual client nodes or for groups of nodes. A node group is a group of client nodes that are acted upon as a single entity. If you specify one or more node groups, the server generates a backup set for each node and places all of the backup sets together on a single set of output volumes. To create a node group, use the `DEFINE NODEGROUP` command, and then use the `DEFINE NODEGROUPMEMBER` command to add nodes to the group. For details, see the *Administrator's Reference*. The client node for which a backup set is generated must be registered to the server.

The media may be directly readable by something such as the following device:

- A CD-ROM, JAZ, or ZIP drive attached to a client's computer.

Administrators can generate multiple copies of backup sets that correspond to some point-in-time. The backup sets can be retained for various time periods. This is an efficient way to create long-term storage of periodic backups, without requiring the data to be sent over the network again.

While an administrator can generate a backup set from any client's backed up files, backup sets can only be used by a backup-archive client.

You cannot generate a backup set for a NAS (network attached storage) node.

When generating backup sets, the server searches for active file versions in an active-data storage pool associated with a FILE device class, if such a pool exists. For details about the complete storage-pool search-and-selection order, see "Active-data pools as sources of active file versions for server operations" on page 225.

Data from a shred storage pool will not be included in a backup set unless you explicitly permit it by setting the `ALLOWSHREDDABLE` parameter to YES in the `GENERATE BACKUPSET` command. If this value is specified, and the client node data includes data from shred pools, that data cannot be shredded. The server will not issue a warning if the backup set operation includes data from shred pools. See "Securing sensitive client data" on page 499 for more information about shredding.

For details about creating and using backup sets, see the following sections:

- "Generating client backup sets on the server"
- "Restoring backup sets from a backup-archive client" on page 507
- "Moving backup sets to other servers" on page 508
- "Managing client backup sets" on page 509

Generating client backup sets on the server

You can generate backup sets on the server for client nodes. The client node for which a backup set is generated must be registered to the server. An incremental backup must be completed for a client node before the server can generate a backup set for the client node.

Task	Required Privilege Class
Generate a backup set	System or restricted policy over the domain to which the node is assigned

The `GENERATE BACKUPSET` command runs as a background process on the server. If you cancel the background process created by this command, the media may not contain a complete backup set.

Generate backup set processing attempts to process all available objects onto the backup set media. However, objects may be skipped due to being unavailable on the server or other errors (I/O, media, hardware) that can occur at the time of backup set generation. Some errors may lead to termination of processing before all available data can be processed. For example, if the source data for a backup set is on multiple sequential volumes and the second or subsequent segment of an object spanning volumes is on a volume that is unavailable, processing is terminated.

If objects are skipped or other problems occur to terminate processing, review all of the messages associated with the process to determine whether or not it should be run again. To obtain a complete backup set, correct any problems that are indicated and reissue the `GENERATE BACKUPSET` command.

To improve performance when generating backup sets, you can do one or both of the following tasks:

- Collocate the primary storage pool in which the client node data is stored. 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.
- Store active backup data in an active-data pool associated with a `FILE` device class. When generating a backup set, the server will search this type of active-data pool for active file versions before searching other possible sources.

See the following sections:

- “Choosing media for generating the backup set”
- “Selecting a name for the backup set” on page 505
- “Setting a retention period for the backup set” on page 505
- “Example: generating a client backup set” on page 505
- “Generating backup sets to a specific point-in-time” on page 506
- “Generating backup sets with multiple data types” on page 506
- “Creating a single set of backup volumes with data from multiple nodes” on page 507

Choosing media for generating the backup set

To generate a backup set, you must specify a device class that is associated with the media to which the backup set will be written.

You can write backup sets to sequential media: sequential tape and device class `FILE`. The tape volumes containing the backup set are not associated with storage pools and, therefore, are not migrated through the storage pool hierarchy.

For device class `FILE`, the server creates each backup set with a file extension of `OST`. You can copy `FILE` device class volumes to removable media that is associated with `CD-ROM`, `JAZ`, or `ZIP` devices, by using the `REMOVABLEFILE` device type.

You can determine whether to use scratch volumes when you generate a backup set. If you do not use specific volumes, the server uses scratch volumes for the backup set.

You can use specific volumes for the backup set. If there is not enough space to store the backup set on the volumes, the server uses scratch volumes to store the remainder of the backup set.

Consider the following items when you select a device class for writing the backup set:

- Generate the backup set on any sequential access devices whose device types are supported on both the client and server. If you do not have access to compatible devices, you will need to define a device class for a device type that is supported on both the client and server.
- Ensure that the media type and recording format used for generating the backup set is supported by the device that will be reading the backup set.

For more information, see “Removable file device configuration” on page 118.

Selecting a name for the backup set

The server adds a unique suffix to the name you specify for the backup set. For example, if you name the backup set mybackupset, the server adds a unique extension, such as 3099, to the name. This allows you to create backup sets with the same name without overwriting previous backup sets.

To later display information about this backup set, you can include a wildcard character with the name, such as mybackupset*, or you can specify the fully qualified name, such as mybackupset.3099.

Setting a retention period for the backup set

You can set the retention period, specified as a number of days, to retain the backup set on the server. You can specify a number between zero and 30,000 days.

Backup sets are retained on the server for 365 days if you do not specify a value. The server uses the retention period to determine when to expire the volumes on which the backup set resides.

Example: generating a client backup set

Generate a backup set on CD-ROM that the client can later use to restore the data. Because CD-ROM is a read-only medium, you must create a backup set on a device that the server can write to before you create the backup set on the CD-ROM.

Do not exceed the CD-ROM data capacity. Use the following steps to generate a backup set on a CD-ROM:

1. Define a device class whose device type is FILE. Name the device class CDFILE:

```
define devclass cdfile devtype=file maxcapacity=640M dir=/backupset
```
2. Create a backup set with a device type of FILE for client node JOHNSON. Name the backup set PROJECT and retain it for 90 days.

```
generate backupset johnson project devclass=file  
volumes=BK1,BK2,BK3 retention=90
```

Attention: Volumes=BK1,BK2,BK3 specifies that the backup set is put into files named BK1, BK2 and BK3. This assumes that the backup set is large enough to require three 650MB files.

3. Use any CD-ROM authoring software to put these files onto CD-ROMs. For this example, the CD-ROM volume names are BK1, BK2, and BK3.
 - The authoring software must label each CD-ROM.
 - The label must exactly match the name of the file.

For an example of using the backup set on the CD-ROM, see “Moving backup sets to other servers” on page 508.

Generating backup sets to a specific point-in-time

You can generate a backup set to a specific point-in-time by using the **PITDATE** and **PITTIME** parameters on the **GENERATE BACKUPSET** command. When these dates are specified, the new backup set will contain files that were active at the specified date and time, even if the files are inactive at the time the **GENERATE BACKUPSET** command is issued.

Backup sets are generated to a point-in-time by using one of two date and time specifications: the date and time specified on the **GENERATE BACKUPSET** command, or the date and time that the **GENERATE BACKUPSET** command was issued.

Point-in-time backup set generation works best if a recent date and time are specified. Files that have expired, or are marked as expire-immediately cannot be included in the backup set.

Generating backup sets with multiple data types

You can use the **GENERATE BACKUPSET** command to generate backup sets that contain either file or image data. You can use the **DATATYPE** parameter to specify the types of data to be included. A separate backup set is generated for each specified data type, but all the backup sets are stored together on a single set of output media.

You can use the **DATATYPE** parameter to limit the backup set to only one data type. For example, you might do this if you don't want to store redundant data on the backup set media. Alternatively, you can specify that both file and image backup data be included from a machine in order to reduce the number of tapes that must be included in your off-site tape rotation.

Image backup sets include the image and all files and directories changed or deleted since the image was backed up so that all backup sets on the media represent the same point in time. Tables of contents are automatically generated for any backup sets that contain image or application data. If the **GENERATE BACKUPSET** command cannot generate a table of contents for one of these backup sets, then it will fail.

For file level backup sets, the table of contents generation is optional. By default, the command attempts to create a table of contents for file level backup sets, but it will not fail if a table of contents is not created. You can control the table of contents option by specifying the **TOC** parameter.

Creating a single set of backup volumes with data from multiple nodes

On the `GENERATE BACKUPSET` command, you can specify multiple nodes or node groups, and you can use wildcards with node names.

A separate backup set is generated for each specified node, but all of the backup sets will be stored together on the same set of output volumes. The backup set for each node has its own entry in the database. The `QUERY BACKUPSET` command will display information about all backup sets, whether they are on their own tape or stacked together with other backup sets onto one tape.

On the `DEFINE BACKUPSET` command, you can also specify multiple nodes or node groups, and you can use wildcards with node names. `DEFINE BACKUPSET` determines what backup sets are on the set of tapes and defines any that match the specified nodes. Specifying only a single wildcard character (*) for the node name has the effect of defining all the backup sets on the set of tapes. Conversely, you can define only those backup sets belonging to a particular node by specifying just the name of that node. Backup sets on tapes belonging to nodes that are not specified on the command are not defined. They will still exist on the tape, but cannot be accessed.

The `QUERY`, `UPDATE`, and `DELETE BACKUPSET` commands also allow the specification of node group names in addition to node names. When you delete backup sets, the volumes on which the backup sets are stored are not returned to scratch as long as any backup set on the volumes remain active.

Restoring backup sets from a backup-archive client

Backup-archive client nodes can restore their backup sets either directly from the server or by using a device attached to the client's computer that will read the media in which the backup set is stored. This second method only works for file-level backup sets.

Backup sets can only be used by a backup-archive client, and only if the files in the backup set originated from a backup-archive client.

For more information about restoring backup sets, see the *Backup-Archive Clients Installation and User's Guide* for your particular operating system.

Selecting individual files for restore

You can query a backup set table of contents to select individual files for restore. Table of contents are generated when a new backup set is created. They contain entries for each object stored in the backup set. Entries detail the position of the object within the backup set.

In order to query the contents of a backup set and choose files to restore, tables of contents need to be loaded into the server database. The backup-archive client can specify more than one backup set table of contents to be loaded to the server at the beginning of a restore session.

Restoring image data from backup sets

Backup sets containing image data can be used during a disaster recovery situation, for example, when a hard drive crashes and needs to be replaced. Individual file restore is not possible using an image backup, so backup sets containing normal file system data should be maintained for most other restores. A backup set may contain image data or file system data, including files and directories, but not both.

Image backups and restores require a table of contents when generating a backup set for image data. If the table of contents existed but was deleted for some reason then the image backup set cannot be restored until the table of contents is regenerated with the GENERATE BACKUPSETTOC command.

Moving backup sets to other servers

You can define (move) a backup set generated on one server to another Tivoli Storage Manager server. Any client backup set that you generate on one server can be defined to another server as long as the servers share ../common device type.

The level of the server defining the backup set must be equal to or greater than the level of the server that generated the backup set.

Using the example described in “Example: generating a client backup set” on page 505, you can make the backup set that was copied to the CD-ROM available to another server by issuing the following command:

```
define backupset johnson project devclass=cdrom volumes=BK1,BK2,BK3
description="backup set copied to CD-ROM"
```

Task	Required Privilege Class
Define a backup set	If the REQSYSAUTHOUTFILE server option is set to YES, system privilege is required. If the REQSYSAUTHOUTFILE server option is set to NO, system or restricted policy over the domain to which the node is assigned is required.

If you have multiple servers connecting to different clients, the DEFINE BACKUPSET command makes it possible for you to take a previously generated backup set and make it available to other servers. The purpose is to allow the user flexibility in moving backup sets to different servers, thus allowing the user the ability to restore their data from a server other than the one on which the backup set was created.

Attention:

1. Devclass=cdrom specifies a device class of type REMOVABLEFILE that points to your CD-ROM drive. CD-ROMs have a maximum capacity of 650MB.
2. Volumes=BK1,BK2,BK3 specifies the names of the volumes containing the backup set. The volume label of these CD-ROMs must match the name of the file on the volume exactly.

Managing client backup sets

You can update, query, and delete backup sets.

Task	Required Privilege Class
Update the retention period assigned to a backup set	System or restricted policy over the domain to which the node is assigned
Display information about backup sets	Any administrator
Display information about backup set contents	System or restricted policy over the domain to which the node is assigned
Delete backup set	If the REQSYSAUTHOUTFILE server option is set to YES, system privilege is required. If the REQSYSAUTHOUTFILE server option is set to NO, system or restricted policy over the domain to which the node is assigned is required.

Generating a table of contents for a backup set

Tables of contents for backup sets are created as part of backup set generation.

- By default, when generating a backup set containing files and directories, the server will attempt to create a table of contents for a new backup set, but will not fail the backup set generation process if it cannot do so. You can, however, choose to require a table of contents by setting the TOC parameter to YES.
- When generating a backup set that contains API data or images, the server will require that a table of contents is generated. You cannot override this default.

In either case, if a table of contents is required and the server cannot create it, the backup set generation process will fail.

Tables of contents:

- Reside on the server even if the backup set's media has been moved off-site.
- Can be generated for existing backup sets that do not contain a table of contents.
- Can be re-generated when a backup set is defined on a new server, or if using a user-generated copy on a different medium.

The GENERATE BACKUPSETTOC command allows a table of contents to be created for backup sets that do not have one. It may be used after a backup set is added to the server via the DEFINE BACKUPSET command or for backup sets that were generated by an earlier release of the Tivoli Storage Manager server.

Backup set tables of contents are stored in the storage pool identified by the TOCDESTINATION attribute of the backup copy group associated with the management class to which the backup set is bound. The management class to which the backup set is bound will either be the default management class in the policy domain in which the backup set's node is registered, or the management class specified by the TOCMGmtclass parameter of the GENERATE BACKUPSET, GENERATE BACKUPSETTOC, or DEFINE BACKUPSET command. Tables of contents for backup sets are retained until the backup set with which they are associated expires or is deleted. They are not subject to the policy associated with their management class. You can issue the QUERY BACKUPSET command to show whether a given backup set has a table of contents or not. Output from the QUERY BACKUPSET command can be filtered based on the existence of a table of contents. This allows you to determine which backup sets may need to have a new table of contents created, or conversely, which backup sets could be used with the

client's file-level restore.

Updating the retention period of a backup set

When you want to change the number of days the server retains a backup set, update the retention period that is associated with the backup set.

To update the retention period assigned to backup set named ENGDATA.3099, belonging to client node JANE, to 120 days, issue the following example command:
update backupset jane engdata.3099 retention=120

Displaying backup set information

To view information about backup sets, you can use the QUERY BACKUPSET command. The output that is displayed lists information such as the name of the client node whose data is contained in the backup set as well as the description of the backup set, assuming one has been used.

The following figure shows the report that is displayed after you enter:

```
query backupset f=d
```

```
Node Name: JANE
Backup Set Name: MYBACKUPSET.3099
Date/Time: 09/04/2002 16:17:47
Retention Period: 60
Device Class Name: DCFILE
Description:
Filespace names: \\jane\c$ \\jane\d$
Volume names: /tsspool/bksetvo101.ost /tsspool/bksetvo102.ost
```

The **FORMAT=DETAILED** parameter on the QUERY BACKUPSET provides the client file spaces contained in the backup set and the list of volumes of the backup set.

Displaying contents of backup sets

You can display information about the contents of backup sets by using the QUERY BACKUPSETCONTENTS command. When you issue the query, the server displays only one backup set at a time.

The server displays information about the files and directories that are contained in a backup set. After you issue the query backupsetcontents jane engdata.3099 command, the following output is displayed:

Node Name	Filespace Name	Client's Name for File
JANE	/srvr	/deblock
JANE	/srvr	/deblock.c
JANE	/srvr	/dsmerror.log
JANE	/srvr	/dsmxxxxx.log
JANE

Tip: To display the contents of an image backup set, specify DATATYPE=IMAGE on the QUERY BACKUPSETCONTENTS command.

Displaying file space and file names

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 Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (?), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

Deleting backup sets

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.

To delete a backup set named ENGDATA.3099, belonging to client node JANE, created before 11:59 p.m. on March 18, 1999, issue the following command:

```
delete backupset jane engdata.3099 begindate=03/18/1999 begintime=23:59
```

To delete all backup sets belonging to client node JANE, created before 11:59 p.m. on March 18, 1999, enter:

```
delete backupset jane * begindate=03/18/1999 begintime=23:59
```

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.

After a backup set is deleted, the volumes return to scratch status if Tivoli Storage Manager acquired them as scratch volumes. Scratch volumes associated with a device type of FILE are deleted. However, if multiple backup sets are stored on the volumes, all of the backup sets must expire or be deleted before the volumes are returned to scratch.

Enabling clients to use subfile backup

A basic problem that remote and mobile users face today is connecting to storage management services by using modems with limited bandwidth or poor line quality. This creates a need for users to minimize the amount of data they send over the network, as well as the time that they are connected to the network.

To help address this problem, you can use subfile backups. When a client's file has been previously backed up, any subsequent backups are typically made of the portion of the client's file that has changed (a subfile), rather than the entire file. A base file is represented by a backup of the entire file and is the file on which subfiles are dependent. If the changes to a file are extensive, a user can request a backup on the entire file. A new base file is established on which subsequent subfile backups are dependent.

This type of backup makes it possible for mobile users to reduce connection time, network traffic, and the time it takes to do a backup. To enable this type of backup, see “Setting up clients to use subfile backup.”

Subfile backups

Assume that on a Monday, a user requests an incremental backup of a file called CUST.TXT. The user makes daily updates to the CUST.TXT file and requests subsequent backups.

The following table describes how Tivoli Storage Manager manages backups of this file.

Version	Day of subsequent backup	What Tivoli Storage Manager backs up
One	Monday	The entire CUST.TXT file (the base file)
Two	Tuesday	A subfile of CUST.TXT. The server compares the file backed up on Monday with the file that needs to be backed up on Tuesday. A subfile containing the changes between the two files is sent to the server for the backup.
Three	Wednesday	A subfile of CUST.TXT. Tivoli Storage Manager compares the file backed up on Monday with the file that needs to be backed up on Wednesday. A subfile containing the changes between the two files is sent to the server for the backup.

Related reference

“Setting policy to enable point-in-time restore for clients” on page 488

“Policy for logical volume backups” on page 483

Setting up clients to use subfile backup

The subfile backup pertains to the sections of the files that have changed.

To enable subfile backup, perform the following tasks:

1. On the server: You must set up the server to allow clients to back up subfiles. Issue the SET SUBFILE command:

```
set subfile client
```
2. On the clients: The SUBFILEBACKUP, SUBFILECACHEDPATH, and SUBFILECACHESIZE options must be set in the client's options file (dsm.opt). You can control these options from the server by including them in client option sets. For example, you can disable subfile backup for individual client nodes by setting SUBFILEBACKUP=NO in the client option set associated with the client node. See “Creating client option sets on the server” on page 411 for how to set up and use client option sets.
See *Backup-Archive Clients Installation and User's Guide* for more information about the options.

Managing subfile backups

Tivoli Storage Manager manages subfiles that are restored, exported, imported, or added to a backup set.

Restoring subfiles

When a client issues a request to restore subfiles, Tivoli Storage Manager restores subfiles along with the corresponding base file back to the client. This process is transparent to the client. That is, the client does not have to determine whether all subfiles and corresponding base file were restored during the restore operation.

You can define (move) a backup set that contains subfiles to an earlier version of a server that is not enabled for subfile backup. That server can restore the backup set containing the subfiles to a client not able to restore subfiles. However, this process is not recommended as it could result in a data integrity problem.

Exporting and importing subfiles

When subfiles are exported during an export operation, Tivoli Storage Manager also exports the corresponding base file to volumes you specify.

When the base file and its dependent subfiles are imported from the volumes to a target server and import processing is canceled while the base file and subfiles are being imported, the server automatically deletes any incomplete base files and subfiles that were stored on the target server.

Expiration processing of base files and subfiles

Because subfiles are useless without the corresponding base file, the server processes base files eligible for expiration differently.

For example, when expiration processing runs, Tivoli Storage Manager recognizes a base file as eligible for expiration but does not delete the file until all its dependent subfiles have expired. For more information on how the server manages file expiration, see “Running expiration processing to delete expired files” on page 472.

Adding subfiles to backup sets

When a subfile is added to a backup set, Tivoli Storage Manager includes its corresponding base file with the backup set.

If the base file and dependent subfiles are stored on separate volumes when a backup set is created, additional volume mounts may be required to create the backup set.

Deleting base files

If a base file is deleted as a result of processing a DELETE VOLUME command, the server recognizes its dependent subfiles and deletes them from the server as well. Subfiles without the corresponding base file are incomplete and useless to the user.

Optimizing restore operations for clients

The progressive incremental backup that is the Tivoli Storage Manager standard results in operations that are optimized for the restore of individual files or small numbers of files.

Progressive incremental backup minimizes tape usage, reduces network traffic during backup operations, and eliminates the storage and tracking of multiple copies of the same data. Progressive incremental backup may reduce the impact to client applications during backup. For a level of performance that is balanced across both backup and restore operations, the best method is usually using progressive incremental backup with collocation set on in the storage pool.

If restore performance is more important than a balance between backup and restore operations, you can optimize based on your goals for restore performance. When you optimize for restore, there are often costs in tape usage and backup performance. To balance the costs against the need for optimized restore operations, you might perform the following tasks:

1. Identify systems that are most critical to your business. Consider where your most important data resides, what is most critical to restore, and what needs the fastest restore. Identify which systems and applications you need to focus on optimizing for restore.
2. Identify your goals and order the goals by priority. Some goals to consider are:
 - Disaster recovery or recovery from hardware crashes, requiring file system restores
 - Recovery from loss or deletion of individual files or groups of files
 - Recovery for database applications
 - Point-in-time recovery of groups of files

The importance of each goal can vary for the different client systems that you identified as being most critical.

For more background on restore operations for clients, see “Concepts for client restore operations” on page 517.

Environment considerations

The performance of Tivoli Storage Manager depends on the environment, including network characteristics, storage hardware (including the types of tape drives used and the availability of snapshot functions), and time constraints for backup and restore operations.

Consider using disk to store data that requires quick restoration. For data that is less critical, store the data to disk, then allow or force the data to migrate to tape later.

You can also use active-data pools to store active versions of client backup data. Archive and space-managed data is not permitted in active-data pools. Inactive files are deleted during reclamation processing. Active-data pools associated with a sequential-access disk (FILE) device class eliminate the need for tape mounts, and the server does not have to position past inactive files. In addition, FILE volumes can be accessed concurrently by multiple client sessions or server processes. Active-data pools associated with a SERVER device class offer similar advantages,

and you can save bandwidth during client restores. You can also create active-data pools that use tape or optical media, which can be moved off-site, but which require tape mounts.

If you do not use disk or active-data pools, you need to consider how restore performance is affected by the layout of data across single or multiple tape volumes. Major causes of performance problems are excessive tape mounts and the need to skip over expired or inactive data on a tape. After a long series of incremental backups, perhaps over years, the active data for a single file space can be spread across many tape volumes. A single tape volume can have active data mixed with inactive and expired data. See the following sections, which discuss ways to control the placement of data, such as:

- Use collocation in storage pools.
- Limit the number of inactive versions of data through policy.
- Use the `MOVE DATA` or `MOVE NODEDATA` commands.

Restoring entire file systems

Using a file system image backup optimizes restore operations when an entire file system needs to be restored, such as in disaster recovery or recovery from a hardware failure. Restoring from an image backup minimizes concurrent mounts of tapes and positioning within a tape during the restore operation.

Consider the following information as aids to file system restore operations:

- Combine image backups with progressive incremental backups for the file system. This allows for full restore to an arbitrary point-in-time.
- To minimize disruption to the client during backup, use either hardware-based or software-based snapshot techniques for the file system.
- Perform image backups infrequently. More frequent image backups give better point-in-time granularity, but will cost in terms of tape usage, interruption of the client system during backup, and greater network bandwidth needed.

A guideline is to perform an image backup when more than 20% of the data in the file system has changed since the last image backup.

The capability for image backup is not available for all clients at this time. If image backup is not available for your client, use file-level restore as an alternative.

Restoring parts of file systems

Progressive incremental backups optimize restore operations for small numbers of files or groups of files. These backups also make optimal use of network bandwidth for backup operations, and may minimize elapsed backup time and tape usage.

If you want to optimize for restoring a file or a group of files, or for a system on which an image backup cannot be made, consider the following additional methods:

- Use collocation by group, by a single client node, or by client file space for primary sequential pools that clients back up to. For large file spaces for which restore performance is critical, consider creating mount points on the client system. This would allow collocation of data below the file space level.
- Specify the client option `COLLOCATEBYFILESPEC`. This can help limit the number of tapes written to by objects from one file specification. See the *Administrator's Reference* for more information about client options.

- Use the `MOVE NODEDATA` command to consolidate critical data in tape storage pools, even in storage pools that have collocation set on. It may be important to consolidate data for certain nodes, file spaces, and data types more often than for others. If you do not use collocation or are limited by tape quantity, you may want to do this more often. The rate of data turnover is also something to consider.

Use the **RECONSTRUCT** parameter on the command to remove unused space in file aggregates when the aggregates are moved. Reconstruction is responsible for removing inactive backup files in active-data pools. Specifying `RECONSTRUCT=NO` when moving data from volumes in an active-data pool will prevent the inactive backup files from being removed.

Use the command for staging data to disk when the lead time for a restore request allows it.

The effectiveness of the command in optimizing for restore might be reduced if a large number of versions are kept.

- Create backup sets that can be taken to the client system and used to restore from directly. This is effective if there is sufficient lead time prior to the restore, and can save network bandwidth.

Creation of backup sets can also be done periodically when resources are available, for example on weekends.

- Use progressive incremental backups, but periodically force a full backup. Some users have found it effective to define multiple Tivoli Storage Manager client nodes on a system. One client node performs the incremental backups and uses policies which retain multiple versions. Another client node performs either full backups or incremental backups with collocation, but uses policies that retain a single version. One node can be used for restoring older versions of individual files, and the other client node can be used for restoring a complete file system or directory tree to the latest version.
- Create multiple storage pool hierarchies for clients with different priorities. For the most critical data, use of only disk storage might be the best choice. Using different storage hierarchies also allows you to set collocation differently in the different hierarchies.
- Minimize the number of versions you keep. This reduces the amount of time spent positioning a tape during a restore operation. An alternative would be to perform full backups.
- Consider storage media characteristics, for example, the type of tape drive you use. Use full file system backups if the tape drives you use are relatively slow at positioning operations.

See “Keeping client files together using collocation” on page 310 for more information about collocation.

Restoring databases for applications

Doing more frequent full backups leads to faster restores for databases. For some database products, you can use multiple sessions to restore, you can restore just the database, or restore just the logs for the database.

Optimal techniques for specific Tivoli Storage Manager application clients are documented in <http://www.ibm.com/shop/publications/order/>.

Restoring files to a point-in-time

If you need the ability to restore files to a point in time, consider setting policy to keep a large number of versions (by setting VEREXISTS=NOLIMIT and VERDELETED=NOLIMIT in the backup copy group).

Keeping a large number of versions is not essential for restoring to a point-in-time, but by increasing the number of versions that are kept, it may be possible to restore from an earlier point-in-time and still find the versions corresponding to that time.

If you also schedule incremental backups regularly, you will have greater granularity in restoring to a discrete point-in-time. However, keeping a large number of versions can degrade the performance of restore operations. Setting policy to keep a large number of versions also has costs in terms of database space and storage pool space. It may have overall performance implications.

If you cannot afford the resource costs of keeping the large numbers of file versions and need the ability to restore to a point-in-time, consider using backup sets, exporting the client data, or using archive. Using backup sets, exporting client data, or archiving files gives you the capability to restore to the point in time when the backup set was generated, the export was performed, or the archive was created. Keep in mind that when you need to restore the data, your selection is limited to the time at which you created the backup set, export, or archive.

Note: If you use the archive function, you should create an archive monthly or yearly. Archive should not be used as a primary backup method because frequent archives with large amounts of data can affect server performance.

See “Restoring parts of file systems” on page 515.

Concepts for client restore operations

The client restore operations include the no query restore, backup and restore using multiple commands, restore using multiple sessions on clients, and the control of resource utilization by a client.

“No-query restore processes”

“Using multiple commands with backup and restore” on page 518

“Using multiple sessions on clients for a restore” on page 519

“Controlling resource utilization by a client” on page 519

No-query restore processes

The client uses two different methods for restore operations: Standard restore (also called classic restore), and no-query restore.

Standard restore requires more interaction between the client and the server, and multiple processes cannot be used for the restore. The no-query restore requires less interaction between the client and the server, and the client can use multiple sessions for the restore process. The no-query restore process is useful when restoring large file systems on a client with limited memory because it avoids some processing that can affect the performance of other client applications. In addition, it can achieve a high degree of parallelism by restoring with multiple sessions from the server and storage agent simultaneously.

The method is called no-query restore because the client sends a single restore request to the server instead of querying the server for each object to be restored.

The server returns the files and directories to the client without further action by the client. The client accepts the data coming from the server and restores it to the destination named on the restore command.

The no-query restore process is used by the client only when the restore request meets both of the following criteria:

- You enter the restore command with a source file specification that has an unrestricted wildcard.

An example of a source file specification with an unrestricted wildcard is:

```
/home/mydocs/2002/*
```

An example of a source file specification with a restricted wildcard is:

```
/home/mydocs/2002/sales.*
```

- You do not specify any of the following client options:
 - inactive
 - latest
 - pick
 - fromdate
 - todate

To force the use of classic restore, use `?*` in the source file specification rather than `*`. For example:

```
/home/mydocs/2002/?*
```

For more information about restore processes, see *Backup-Archive Clients Installation and User's Guide*.

Using multiple commands with backup and restore

Another method which can aid in both the backup and restore of client nodes with critical data is to manage the backup process through multiple commands instead of multiple sessions. For example, when using multi-session backup, multiple backup sessions may be contending for the same underlying hard disk, thus causing delays.

An alternative is to manage this process externally by starting multiple `dsmc` commands. Each command backs up a pre-determined number of file systems. Using this method in conjunction with collocation at the file space level can improve backup throughput and allow for parallel restore processes across the same hard drives.

You must issue multiple commands when you are restoring more than one file space. For example, when you are restoring a c: drive and a d: drive on a Windows system you must issue multiple commands.

Consider using multiple commands when you are restoring a single, large file space, and all of the following conditions are true:

- The data was backed up to a storage pool that had collocation set to FILESPACE. Files will be on multiple volumes, and the volumes can be mounted by multiple processes.
- The files are approximately evenly distributed across the different top-level directories in the file space.
- The number of top-level directories in the file space is not large.
- You can issue commands for the different top-level directories, and the commands do not overlap (so that the same file is not restored multiple times by different commands).

Issue multiple commands either by issuing the commands one after another in a single session or window, or by issuing commands at the same time from different command windows.

When you enter multiple commands to restore files from a single file space, you must specify a unique part of the file space in each restore command. Be sure that you do not use any overlapping file specifications in the commands. To display a list of the directories in a file space, use the query backup command on the client. For example:

```
dsmc query backup -dirsonly -subdir=no /usr/
```

For more information, see *Backup-Archive Clients Installation and User's Guide*.

Using multiple sessions on clients for a restore

To use multiple sessions, data for the client must be on multiple, sequential access volumes or a combination of sequential access volumes and disk. The data for a client usually becomes spread out over some number of volumes over time. This occurs deliberately when collocation is not used for the storage pool where the client data is stored.

To potentially benefit from multi-session restores, consider collocating client data by group. Collocation by group can cause data for a node to be distributed on more than one volume (while keeping the group's total data on as few volumes as possible).

Restore operations themselves are not restricted on mount points. The **MAXNUMMP** parameter in the REGISTER NODE or UPDATE NODE command applies to restore in that the client can restrict the number of sessions based on the combination of the **MAXNUMMP** value and the client's RESOURCEUTILIZATION value.

Set the client option for resource utilization to one greater than the number of desired sessions (use the number of drives that you want that single client to use). The client option can be included in a client option set.

Issue the restore command so that it results in a no query restore process.

See "Controlling resource utilization by a client."

See "No-query restore processes" on page 517 for details.

Controlling resource utilization by a client

You can control the number of mount points (equivalent to drives) allowed to a client by setting the **MAXNUMMP** parameter on either the UPDATE NODE or REGISTER NODE command.

At the client, the option for resource utilization also has an effect on how many drives (how many sessions) the client can use. The client option, resource utilization, can be included in a client option set. If the number specified in the **MAXNUMMP** parameter is too low and there are not enough mount points for each of the sessions, it may not be possible to achieve the benefits of the multiple sessions specified in the resource utilization client option.

- For backup operations, you might want to prevent multiple sessions if the client is backing up directly to tape, so that data is not spread among multiple volumes. Multiple sessions can be prevented at the client by using a value of 2 for the resource utilization option on the client.

- For restore operations, set the resource utilization option to one greater than the number of desired sessions. Use the number of drives that you want that single client to use.

Managing archive data

Managing archive data on the server becomes important when you have client nodes that archive large numbers (hundreds or thousands) of files every day.

If you archive files using automated tools that invoke the command line client or API, there is a good chance you will encounter such extremely large numbers. If you have noticed that performance is degrading over time while performing an archive operation (adding, querying, retrieving or deleting archived files), or have a large amount of storage consumed by archives, consider using some advanced techniques that can help.

“Archive operations overview”

“Managing storage usage for archives” on page 521

Archive operations overview

IBM Tivoli Storage Manager allows clients to organize archived files into packages. The description field of an archive request serves as the way to identify the package.

All files that are archived with the same description become members of the same archive package. If the user does not specify a description when archiving, the client program provides a default description with each archive request. The default description includes the date.

When files are archived, the client program archives the paths (directories) to those files to preserve access permissions which are specific to the operating system. Directories are also included in archive packages. If the same directory is archived with different descriptions, the directory is stored once with each package. If a command line user issues a QUERY ARCHIVE command, multiple entries for the same directory may appear. Closer inspection shows that each entry has a different description.

The GUI and Web client programs allow a user to navigate through a client node's archives by first displaying all descriptions (the package identifiers), then the directories, and finally the files. Users can retrieve or delete individual files or all files in a directory. Command line client and API users can specify a description when they archive files, or when they send requests to query, retrieve or delete archived files.

When retrieving files, the server searches for the most current file versions. It will search in an active-data storage pool associated with a FILE device class, if such a pool exists.

Managing storage usage for archives

The amount of server database storage needed for archive operations is affected by the use of descriptions for archiving and by the archiving of directories.

Consider the following two actions that you can take to minimize the storage usage:

Minimize the number of unique descriptions

You can reduce storage usage by archiving more files into fewer packages (by reducing the number of unique descriptions). The amount of storage used for directories is also affected by the number of packages. If you archive a file three different times using three different descriptions, the server stores both the file and the directory three times, once in each package. If you archives the same file three different times using just one description, the server stores the file three times, but the directory is stored just one time.

Archive directories only if needed

Archiving directories might be necessary if the directories are needed to group files for query or retrieve, or if the directory-level access permission information needs to be archived.

The users of the GUI and Web client programs need descriptions to aid in navigation, to find archived files. You can minimize storage usage for archives by reducing the number of packages. For client nodes that are always accessed via the command-line interface you can also use some other techniques.

See the following topics for more information:

“Minimizing archive storage use”

“Reducing archive storage use for client nodes with many archives” on page 522

“Preventing archive of directory-level access permissions” on page 523

Minimizing archive storage use

You can minimize the storage used for new archives for a client node that is always accessed through the command-line interface by ensuring that each file archived by the client node has a unique name and that you use the same description or a limited number of unique descriptions to group and identify archived files for the client node.

If the user follows these guidelines, the client node will have one or a limited number of archive packages. Because of the small number of packages, there are only small numbers of copies of each directory entry. The savings in storage space that result are noticeable when files with the same path specification are archived multiple times over multiple days.

Having a unique description for an archive is not necessary if archived files themselves have unique names. For example, files might already have a date or time stamp encoded in their names. The same description can be used each time a client node archives files. The user must specify the description to avoid getting the default for the description, which is `Archive Date: yyyy/mm/dd`. The user can specify a description as short as one character, or as simple as a blank space (but not a null entry). For example, the user enters this command:

```
dsmc archive c:\10sept2003ch1.doc -description=" "
```

If the user follows these guidelines, you do not need to convert the node because converting the node affects archive descriptions. A client node that is used with these guidelines has only one or a small number of archive descriptions, so there is no improvement in performance to be gained by converting the node.

See the *Backup-Archive Clients Installation and User's Guide* for details about archive operations and client options.

Reducing archive storage use for client nodes with many archives

If a client node already has a large number of archive packages, you can reduce the storage use by updating that node's archives.

Do not run the UPDATE ARCHIVE command while any other processing for the node is running. If this command is issued for a node with any other object insertion or deletion activity occurring at the same time, locking contention may occur. This may result in processes and sessions hanging until the resource timeout is reached and the processes and sessions terminate.

When you update archives for a node, you have two choices for the action to take:

Delete directory entries in all archive packages

This action preserves the archive packages, but removes directory entries for all packages, reducing the amount of storage used for archives. Do this only when directory entries that include access permissions are not needed in the archive packages, and the paths are not needed to query or retrieve a group of files. The amount of reduction depends on the number of packages and the number of directory entries. For example, to remove directory entries for the client node SNOOPY, enter this command:

```
update archive snoopy deletedirs
```

Attention: After you delete the directory entries, the directory entries cannot be recreated in the archive packages. Do not use this option if users of the client node need to archive access permissions for directories.

Reduce the number of archive packages to a single package for the node

This action removes all unique descriptions, thereby reducing the number of archive packages to one for the client node. Do this only when the descriptions are not needed and are causing large use of storage. This action also removes directory entries in the archive packages. Because there is now one package, there is one entry for each directory. For example, to reduce the archive packages to one for the client node SNOOPY, enter this command:

```
update archive snoopy resetdescriptions
```

After updating the archives for a node in this way, keep the archive package count to a minimum.

Attention: You cannot recreate the packages after the descriptions have been deleted. Do not use this option if users of the client node manage archives by packages, or if the client node is accessed via the GUI or Web client interface.

Preventing archive of directory-level access permissions

Command-line users who do not need to archive the directory-level access permissions can reduce storage requirements by using the `v2archive` option with the archive request. This option prevents the creation of directory entries in archive packages.

See *Backup-Archive Clients Installation and User's Guide* for details about the option.

Tip: The GUI and Web client programs use the directories to allow users to navigate to the archived files. This option is not recommended for GUI or Web client interface users.

Chapter 16. Scheduling operations for client nodes

Scheduling client operations can provide better protection for data, because operations can occur consistently without user intervention. Scheduling can also maximize utilization of resources. When client backups are scheduled at times of lower usage, the impact on the network is minimized.

You can schedule operations such as the following:

- Archiving and retrieving client data.
- Running operating system commands.
- Running macro or command files that contain operating system commands, commands, or both. You can schedule a command file to run on clients or application clients.
- Backing up and restoring client data and application client data.

Administrators can perform the following activities to schedule client operations:

Tasks:
"Scheduling a client operation" on page 526
"Defining client schedules" on page 526
"Associating client nodes with schedules" on page 527
"Starting the scheduler on the clients" on page 527
"Displaying information about schedules" on page 532
"Starting the scheduler on the clients" on page 527
"Displaying information about schedules" on page 532
"Creating schedules for running command files" on page 529
"Updating the client options file to automatically generate a new password" on page 530

Prerequisites to scheduling operations

To interact with Tivoli Storage Manager for scheduling operations, client machines must meet certain prerequisites.

- The client node must be registered with the server. For information, see Chapter 11, "Adding client nodes," on page 367.
- The client options file (dsm.opt) must contain the network address of the server that the client will contact for services. See "Connecting nodes with the server" on page 372 for more information.
- The scheduler must be started on the client machine. Refer to the *Backup-Archive Clients Installation and User's Guide* for details.

Scheduling a client operation

To automate client operations, you can define new schedules.

When you define a schedule, you assign it to a specific policy domain. You can define more than one schedule for each policy domain. To set up a client schedule on the server, perform these steps:

1. Define a schedule (DEFINE SCHEDULE command). (“Defining client schedules”)
2. Associate client nodes with the schedule (DEFINE ASSOCIATION command). (“Associating client nodes with schedules” on page 527)
3. Ensure that the clients start the client scheduler. (“Starting the scheduler on the clients” on page 527)
4. Display the schedule information and check that the schedule completed successfully (QUERY SCHEDULE and QUERY EVENT commands). (“Displaying schedule information” on page 528)

You can modify, copy, and delete any schedule you create. See Chapter 17, “Managing schedules for client nodes,” on page 531 for more information.

Defining client schedules

When scheduling client operations, you need to know what operation you want to run and the time, day, and frequency to run your operation.

Task	Required Privilege Class
Define client schedules for any policy domain	System or unrestricted policy
Define client schedules for specific policy domains	System, unrestricted policy, or restricted policy for those domains

To define a schedule for daily incremental backups, use the DEFINE SCHEDULE command. You must specify the policy domain to which the schedule belongs and the name of the schedule (the policy domain must already be defined). For example:

```
define schedule engpoldom daily_backup starttime=21:00
duration=2 durunits=hours
```

This command results in the following:

- Schedule *DAILY_BACKUP* is defined for policy domain *ENGPOLDOM*.
- The scheduled action is an incremental backup; this is the default.
- The priority for the operation is 5; this is the default. If schedules conflict, the schedule with the highest priority (lowest number) runs first.
- The schedule window begins at 9:00 p.m., and the schedule itself has 2 hours to start.
- The start window is scheduled every day; this is the default.
- The schedule never expires; this is the default.
- The schedule style is classic; this is the default.

As a best practice, 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.

To change the defaults, see the DEFINE SCHEDULE command in the *Administrator's Reference*.

Associating client nodes with schedules

Client nodes process operations according to the schedules associated with the nodes. A client node can be associated with more than one schedule. However, a node must be assigned to the policy domain to which a schedule belongs.

Task	Required Privilege Class
Associate client nodes with schedules	System, unrestricted policy, or restricted policy for the policy domain to which the schedule belongs

To associate client nodes with a schedule, use the DEFINE ASSOCIATION command. After a client schedule is defined, you can associate client nodes with it by identifying the following information:

- Policy domain to which the schedule belongs
- List of client nodes to associate with the schedule

To associate the ENGNODE client node with the WEEKLY_BACKUP schedule, both of which belong to the ENGPOLDOM policy domain, enter:

```
define association engpoldom weekly_backup engnode
```

Starting the scheduler on the clients

The client scheduler must be started before work scheduled by an administrator can be initiated.

Administrators must ensure that users start the Tivoli Storage Manager scheduler on the client or application client directory, and that the scheduler is running at the schedule start time. After the client scheduler starts, it continues to run and initiates scheduled events until it is stopped.

The way that users start the Tivoli Storage Manager scheduler varies, depending on the operating system that the machine is running. The user can choose to start the client scheduler automatically when the operating system is started, or can start it manually at any time. The user can also have the client acceptor manage the scheduler, starting the scheduler only when needed. For instructions on these tasks, see the *Backup-Archive Clients Installation and User's Guide*.

The client and the Tivoli Storage Manager server can be set up to allow all sessions to be initiated by the server. See "Server-initiated sessions" on page 379 for instructions.

Note: Tivoli Storage Manager does not recognize changes that you made to the client options file while the scheduler is running. For Tivoli Storage Manager to use the new values immediately, you must stop the scheduler and restart it.

Displaying schedule information

When you request information about schedules, the server displays details.

Task	Required Privilege Class
Display information about scheduled operations	Any administrator

- Schedule name
- Policy domain name
- Type of operation to perform
- Start date and time for the initial startup window
- Duration of the startup window
- Time period between startup windows (if using a classic schedule)
- Day (classic schedule) or days (enhanced schedule) of the week on which scheduled operations can begin

The following output shows an example of a report for a classic schedule that is displayed after you enter:

```
query schedule engpoldom
```

Domain	* Schedule Name	Action	Start Date/Time	Duration	Period	Day
ENGPOLDOM	MONTHLY_BACKUP	Inc Bk	09/04/2002 12:45:14	2 H	2 Mo	Sat
ENGPOLDOM	WEEKLY_BACKUP	Inc Bk	09/04/2002 12:46:21	4 H	1 W	Sat

For enhanced schedules, the standard schedule format displays a blank period column and an asterisk in the day of week column. Issue `FORMAT=DETAILED` to display complete information about an enhanced schedule. Refer to the *Administrator's Reference* for command details. The following output shows an example of a report for an enhanced schedule that is displayed after you enter:

```
query schedule engpoldom
```

Domain	* Schedule Name	Action	Start Date/Time	Duration	Period	Day
ENGPOLDOM	MONTHLY_BACKUP	Inc Bk	09/04/2002 12:45:14	2 H	2 Mo	Sat
ENGPOLDOM	WEEKLY_BACKUP	Inc Bk	09/04/2002 12:46:21	4 H		(*)

Checking the status of scheduled operations

A schedule completes successfully if the command associated with the schedule is successfully issued. The success of the issued command is independent on the success of the schedule.

You need to ask these two questions:

- Did the schedule run successfully?

To determine the success of a scheduled operation, query the server. Each scheduled client operation is called an *event*, and is tracked by the server. You can get information about projected and actual scheduled processes by using the `QUERY EVENT` command. You can get information about scheduled processes that did not complete successfully by using exception reporting with this command.

For example, you can issue the following command to find out which events were missed (did not start) in the `ENGPOLDOM` policy domain for the `WEEKLY_BACKUP` schedule in the previous week:

```
query event engpoldom weekly_backup begindate=-7 begintime=now
enddate=today endtime=now exceptionsonly=yes
```

For more information about managing event records, see “Managing event records” on page 534.

- Did the operation or commands run as a result of the schedule run successfully?

To determine the success of the commands issued as the result of a successful schedule, you can:

- Check the client's schedule log.

The schedule log is a file that contains information such as the statistics about the backed-up objects, the name of the server backing up the objects, and the time and date of the next scheduled operation. By default, Tivoli Storage Manager stores the schedule log as a file called *dsmsched.log* and places the file in the directory where the Tivoli Storage Manager backup-archive client is installed. Refer to *Backup-Archive Clients Installation and User's Guide* for more information.

- Check the server's activity log.

Search or query the activity log for related messages. For example, search for messages that mention the client node name, within the time period that the schedule ran. For example:

```
query actlog begindate=02/23/2001 enddate=02/26/2001 originator=client
nodename=hermione
```

- Issue the QUERY EVENT command with FORMAT=DETAILED, and view the Result field of the output screen. For example:

```
query event nodes=joe domain2 standard begindate=02/26/2002 enddate=02/27/2002
format=detailed
```

Refer to *Backup-Archive Clients Installation and User's Guide* for an explanation of the Result field.

Creating schedules for running command files

For some clients, you may want to run a command for a different application before running a Tivoli Storage Manager backup. For example, you may want to stop a database application, back up files with Tivoli Storage Manager, and then restart the application. To do this, you can schedule the running of a command file. Application clients *require* schedules that run command files.

A command file (also known as a macro or batch file on different operating systems) is stored on the client. This file contains a sequence of commands that are intended to be run during a scheduled start date and time window. Commands can include operating system commands, the Tivoli Storage Manager client's DSMC command, and commands for other applications.

To use command files, administrators must create schedules with the ACTION=MACRO parameter. For example, you can define a schedule called DAILY_INCR that will process a command file called *c:\incr.cmd* on the client:

```
define schedule standard daily_incr description="daily incremental file"
action=macro objects="c:\incr.cmd" starttime=18:00 duration=5
durunits=minutes period=1 perunits=day dayofweek=any
```

Associate the client with the schedule and ensure that the scheduler is started on the client or application client directory. The schedule runs the file called *c:\incr.cmd* once a day between 6:00 p.m. and 6:05 p.m., every day of the week.

Updating the client options file to automatically generate a new password

If the server uses password authentication, clients must use passwords. Passwords are then also required for the server to process scheduled operations for client nodes.

If a password expires and is not updated, scheduled operations fail. You can prevent failed operations by allowing Tivoli Storage Manager to generate a new password when the current password expires. If you set the `PASSWORDACCESS` option to `GENERATE` in the Tivoli Storage Manager client options file, `dsm.opt`, Tivoli Storage Manager automatically generates a new password for your client node each time it expires, encrypts and stores the password in a file, and retrieves the password from that file during scheduled operations. You are not prompted for the password.

The `PASSWORDACCESS GENERATE` option is also required in other situations, such as when you want to use the Web backup-archive client to access a client node. See the *Backup-Archive Clients Installation and User's Guide* for more information.

Chapter 17. Managing schedules for client nodes

You can manage and coordinate Tivoli Storage Manager schedules for registered client nodes.

Administrators can perform the following tasks:

Tasks:
"Managing node associations with schedules" on page 533
"Specifying one-time actions for client nodes" on page 543
"Managing event records" on page 534
"Managing the throughput of scheduled operations" on page 537
"Managing IBM Tivoli Storage Manager schedules"

For a description of what Tivoli Storage Manager views as client nodes, see Chapter 11, "Adding client nodes," on page 367. For information about the scheduler and creating schedules, see Chapter 16, "Scheduling operations for client nodes," on page 525

Managing IBM Tivoli Storage Manager schedules

You can perform the following activities to manage schedules.

Task	Required Privilege Class
Verify that the schedule ran	Any administrator
Add, copy, modify, or delete client schedules in any policy domain	System or unrestricted policy
Add, copy, modify, or delete client schedules for specific policy domains	System, unrestricted policy, or restricted policy for those domains
Display information about scheduled operations	Any administrator

Adding new schedules

You can add Tivoli Storage Manager schedules by using the Schedule Configuration wizard in the Tivoli Storage Manager Console.

You can add new Tivoli Storage Manager schedules by using the DEFINE SCHEDULE command.

After you add a new schedule, associate the node with the schedule. For more information, see "Defining client schedules" on page 526.

Copying existing schedules

You can create new schedules by copying existing schedules to the same policy domain or a different policy domain. The schedule description and all schedule parameter values are copied to the new schedule. You can then modify the new schedule to meet site-specific requirements.

Client node associations are not copied to the new schedule. You must associate client nodes with the new schedule before it can be used. The associations for the old schedule are not changed.

To copy the WINTER schedule from policy domain DOMAIN1 to DOMAIN2 and name the new schedule WINTERCOPY, enter:

```
copy schedule domain1 winter domain2 wintercopy
```

For information, see “Associating client nodes with schedules” on page 527.

Modifying schedules

You can modify existing schedules by issuing the UPDATE SCHEDULE command.

For example, to modify the ENGWEEKLY client schedule in the ENGPOLDOM policy domain, enter:

```
update schedule engpoldom engweekly period=5 perunits=days
```

The ENGWEEKLY schedule is updated so that the incremental backup period is now every five days.

Deleting schedules

When you delete a schedule, Tivoli Storage Manager deletes all client node associations for that schedule.

To delete the schedule WINTER in the ENGPOLDOM policy domain, enter:

```
delete schedule engpoldom winter
```

Rather than delete a schedule, you may want to remove all nodes from the schedule and save the schedule for future use. For information, see “Removing nodes from schedules” on page 534.

See “Associating client nodes with schedules” on page 527 for more information.

Displaying information about schedules

You can display information about schedules.

The following information is displayed:

- Schedule name
- Policy domain name
- Type of operation to be performed
- Start date and time for the initial startup window
- Duration of the startup window
- Time period between startup windows (if using a classic schedule)
- Day (classic schedule) or days (enhanced schedule) of the week on which scheduled operations can begin

The following output shows an example of a report for a classic schedule that is displayed after you enter:

```
query schedule engpoldom
```

Domain	* Schedule Name	Action	Start Date/Time	Duration	Period	Day
ENGPOLDOM	MONTHLY_BACKUP	Inc Bk	09/04/2002 12:45:14	2 H	2 Mo	Sat
ENGPOLDOM	WEEKLY_BACKUP	Inc Bk	09/04/2002 12:46:21	4 H	1 W	Sat

For enhanced schedules, the standard schedule format displays a blank period column and an asterisk in the day of week column. Issue `FORMAT=DETAILED` to display complete information about an enhanced schedule. Refer to the *Administrator's Reference* for command details. The following output shows an example of a report for an enhanced schedule that is displayed after you enter:

```
query schedule engpoldom
```

Domain	* Schedule Name	Action	Start Date/Time	Duration	Period	Day
ENGPOLDOM	MONTHLY_BACKUP	Inc Bk	09/04/2002 12:45:14	2 H	2 Mo	Sat
ENGPOLDOM	WEEKLY_BACKUP	Inc Bk	09/04/2002 12:46:21	4 H		(*)

Managing node associations with schedules

You can add and delete node associations from schedules. Nodes can be associated with more than one schedule.

You can perform the following activities to manage associations of client nodes with schedules.

Task	Required Privilege Class
Add new nodes to existing schedules	System or restricted policy over the domain to which the node is assigned
Move nodes to existing schedules	System or restricted policy over the domain to which the node is assigned
Delete nodes associated with a schedule	System or restricted policy over the domain to which the node is assigned
Display nodes associated with a specific schedule	Any administrator

Adding new nodes to existing schedules

You can add new nodes to existing schedules by associating the node with the schedule.

To associate client nodes with a schedule, you can use one of the following methods:

- Issue the `DEFINE ASSOCIATION` command from the command-line interface.

- Use the Administration Center to associate a node with a schedule.

For more information, see “Associating client nodes with schedules” on page 527.

Moving nodes from one schedule to another

You can move a node from one schedule to another schedule.

1. Associate the node to the new schedule. For information, see “Adding new nodes to existing schedules” on page 533.
2. Delete the association of that node from the original schedule.

Related tasks

“Associating client nodes with schedules” on page 527

“Removing nodes from schedules”

Displaying nodes associated with schedules

You can display information about the nodes that are associated with a specific schedule.

For example, you should query an association before deleting a client schedule.

Figure 72 shows the report that is displayed after you enter:

```
query association engpoldom
```

```
Policy Domain Name: ENGPOLDOM
Schedule Name: MONTHLY_BACKUP
Associated Nodes: MAB SSTEINER

Policy Domain Name: ENGPOLDOM
Schedule Name: WEEKLY_BACKUP
Associated Nodes: MAB SSTEINER
```

Figure 72. Query association output

Removing nodes from schedules

When you remove the association of a node to a client schedule, the client no longer runs operations specified by the schedule. However, the remaining client nodes still use the schedule.

To delete the association of the ENGNOD client with the ENGWEEKLY schedule, in the policy domain named ENGPOLDOM, enter:

```
delete association engpoldom engweekly engrnod
```

Instead of deleting a schedule, you may want to delete all associations to it and save the schedule for possible reuse in the future.

Managing event records

Each scheduled client operation is called an *event*. All scheduled events, including their status, are tracked by the server. An *event record* is created in the server database whenever a scheduled event is completed or missed.

You can perform the following activities to manage event records:

Task	Required Privilege Class
Display information about scheduled events	Any administrator
Set the retention period for event records	System
Delete event records	System or unrestricted policy

Displaying information about scheduled events

To help manage schedules for client operations, you can request information about scheduled and completed events by using the QUERY EVENT command.

- To get information about past and projected scheduled processes, use a simple query for events. If the time range you specify includes the future, the results show which events should occur in the future based on current schedules.
- To get information about scheduled processes that did not complete successfully, use the exceptions-only option with the query.

To minimize the processing time when querying events:

- Minimize the time range
- For client schedules, restrict the query to those policy domains, schedules, and client node names for which information is required

You can also find information about scheduled events by checking the log file described in “Checking the schedule log” on page 536.

Displaying all client schedule events

You can display information about all client events by issuing the QUERY EVENT command. The information includes events for both successful and failed schedules. If the administrator specifies a time range that includes the future, Tivoli Storage Manager displays future events with a status of *future*.

Figure 73 shows an example of a report for client node GOODELL that is displayed after you enter:

```
query event standard weekly_backup node=goodell enddate=today+7
```

Scheduled Start	Actual Start	Schedule Name	Node Name	Status
09/04/2002 06:40:00	09/04/2002 07:38:09	WEEKLY_BACKUP	GOODELL	Started
09/16/2002 06:40:00		WEEKLY_BACKUP	GOODELL	Future

Figure 73. Events for a node

Displaying events that ended unsuccessfully

You can display information about scheduled events that ended unsuccessfully by using exception reporting.

For example, you can issue the following command to find out which events were missed in the previous 24 hours, for the DAILY_BACKUP schedule in the STANDARD policy domain:

```
query event standard daily_backup begindate=-1 begintime=now  
enddate=today endtime=now exceptionsonly=yes
```

Figure 74 on page 536 shows an example of the results of this query. To find out why a schedule was missed or failed, you may need to check the schedule log on the client node itself. For example, a schedule can be missed because the scheduler was not started on the client node.

Scheduled Start	Actual Start	Schedule Name	Node Name	Status
09/04/2002 20:30:00	-----	DAILY_BACKUP	ANDREA	Missed
09/04/2002 20:30:00		DAILY_BACKUP	EMILY	Missed

Figure 74. Exception report of events

Displaying past events

If you query the server for events, the server may display past events even if the event records have been deleted.

Such events are displayed with a status of *Uncertain*, indicating that complete information is not available because the event records have been deleted. To determine if event records have been deleted, check the message that is issued after the DELETE EVENT command is processed.

Checking the schedule log

The Tivoli Storage Manager client stores detailed information about each scheduled event in a file. This file contains information such as the statistics about the backed-up objects, the name of the server to which the objects are backed up, and the time and date of the next scheduled operation.

The default name for this file is *dsmsched.log*. The file is located in the directory where the Tivoli Storage Manager backup-archive client is installed. You can override this file name and location by specifying the SCHEDLOGNAME option in the client options file. See the client user's guide for more information.

Managing event records in the server database

By default, the server retains event records for 10 days before automatically removing them from the database. The server automatically deletes event records from the database after the event retention period has passed and after the startup window for the event has elapsed.

You can specify how long event records stay in the database before the server automatically deletes them by using the SET EVENTRETENTION command. You can also manually delete event records from the database, if database space is required.

Setting the event retention period

You can modify the retention period for event records in the database.

To change the retention period to 15 days, enter:

```
set eventretention 15
```

Manually deleting event records

You may want to manually delete event records to increase available database space.

For example, to delete all event records written prior to 11:59 p.m. on June 30, 2002, enter:

```
delete event 06/30/2002 23:59
```

Managing the throughput of scheduled operations

In the Tivoli Storage Manager environment where many nodes attempt to initiate scheduled operations simultaneously, you may have to manage scheduling throughput. You can choose a scheduling mode, and you can control how often client nodes contact the server to perform a scheduled operation.

Administrators can perform the following activities to manage the throughput of scheduled operations.

Task	Required Privilege Class
Modify the default scheduling mode	System
Modify the scheduling period for incremental backup operations	System
Balance the scheduled workload for the server	System
Set the frequency at which client nodes contact the server	System

Modifying the default scheduling mode

Tivoli Storage Manager provides two scheduling modes: *client-polling* and *server-prompted*. The mode indicates how client nodes interact with the server for scheduling operations.

With client-polling mode, client nodes poll the server for the next scheduled event. With server-prompted mode, the server contacts the nodes at the scheduled start time. By default, the server permits both scheduling modes. The default (ANY) allows nodes to specify either scheduling mode in their client options files. You can modify this scheduling mode.

If you modify the default server setting to permit only one scheduling mode, *all* client nodes must specify the same scheduling mode in their client options file. Clients that do not have a matching scheduling mode will not process the scheduled operations. The default mode for client nodes is client-polling.

The scheduler must be started on the client node's machine before a schedule can run in either scheduling mode.

For more information about modes, see “Overview of scheduling modes” on page 538.

By default, clients contact the server (client-polling scheduling mode and SESSIONINITIATION=CLIENTORSERVER). If SESSIONINITIATION is set to the default CLIENTORSERVER value, you can use either client-polling or server-prompted scheduling modes. The client might start sessions with the server by communicating on the TCP/IP port that was defined with a server option. Server-prompted scheduling also can be used to prompt the client to connect to the server.

You can instead prevent clients from starting sessions, and allow only the server to start sessions with clients. To limit the start of backup-archive client sessions to the server only, do the following steps for each node:

1. Use the REGISTER NODE command or the UPDATE NODE command to change the value of the SESSIONINITIATION parameter to SERVERONLY. Specify the high-level address and low-level address options. These options must match what the client is using, otherwise the server will not know how to contact the client.
2. Set the scheduling mode to server-prompted. All sessions must be started by server-prompted scheduling on the port that was defined for the client with the REGISTER NODE or the UPDATE NODE commands.
3. Ensure that the scheduler on the client is started. You cannot use the client acceptor (dsmcad) to start the scheduler when SESSIONINITIATION is set to SERVERONLY.

Overview of scheduling modes

With client-polling mode, client nodes poll the server for the next scheduled event. With server-prompted mode, the server contacts the nodes at the scheduled start time.

See Table 53 and Table 52 for the advantages and disadvantages of client-polling and server-prompted modes.

Table 52. Client-Polling mode

How the mode works	Advantages and disadvantages
<ol style="list-style-type: none"> 1. A client node queries the server at prescribed time intervals to obtain a schedule. This interval is set with a client option, QUERYSCHEDPERIOD. For information about client options, refer to the appropriate <i>Backup-Archive Clients Installation and User's Guide</i>. 2. At the scheduled start time, the client node performs the scheduled operation. 3. When the operation completes, the client sends the results to the server. 4. The client node queries the server for its next scheduled operation. 	<ul style="list-style-type: none"> • Useful when a high percentage of clients start the scheduler manually on a daily basis, for example when their workstations are powered off nightly. • Supports <i>randomization</i>, which is the random distribution of scheduled start times. The administrator can control randomization. By randomizing the start times, Tivoli Storage Manager prevents all clients from attempting to start the schedule at the same time, which could overwhelm server resources. • Valid with all communication methods.

Table 53. Server-Prompted mode

How the mode works	Advantages and disadvantages
<ol style="list-style-type: none"> 1. The server contacts the client node when scheduled operations need to be performed and a server session is available. 2. When contacted, the client node queries the server for the operation, performs the operation, and sends the results to the server. 	<ul style="list-style-type: none"> • Useful if you change the schedule start time frequently. The new start time is implemented without any action required from the client node. • Useful when a high percentage of clients are running the scheduler and are waiting for work. • Useful if you want to restrict sessions to server-initiated. • Does not allow for randomization of scheduled start times. • Valid only with client nodes that use TCP/IP to communicate with the server.

Modifying the scheduling mode on the server

If you modify the default so that the server permits only one scheduling mode for the server, all clients must specify the same scheduling mode in their client options file. Clients that do not have a matching scheduling mode do not process scheduled operations.

Client-Polling Scheduling Mode: To have clients poll the server for scheduled operations, enter:

```
set schedmodes polling
```

Ensure that client nodes specify the same mode in their client options files.

Server-Prompted Scheduling Mode: To have the server prompt clients for scheduled operations, enter:

```
set schedmodes prompted
```

Ensure that client nodes specify the same mode in their client options files.

Any Scheduling Mode: To return to the default scheduling mode so that the server supports both client-polling and server-prompted scheduling modes, enter:

```
set schedmodes any
```

Client nodes can then specify either polling or prompted mode.

Modifying the default scheduling mode on client nodes

Users can set the scheduling mode on client nodes.

They specify either the client-polling or the server-prompted scheduling mode on the command line or in the client user options file. (On UNIX and Linux systems, root users set the scheduling mode in the client system options file.)

For more information, refer to the appropriate *Backup-Archive Clients Installation and User's Guide*.

Specifying the schedule period for incremental backup operations

When you define a backup copy group, you specify the copy frequency, which is the minimum interval between successive backups of a file.

When you define a schedule, you specify the length of time between processing of the schedule. Consider how these interact to ensure that the clients get the backup coverage that you intend.

See “Defining and updating a backup copy group” on page 462.

Balancing the scheduled workload for the server

You can control the server’s workload and ensure that the server can perform all scheduled operations within the specified window.

To enable the server to complete all schedules for clients, you may need to use trial and error to control the workload. To estimate how long client operations take, test schedules on several representative client nodes. Keep in mind, for example, that the first incremental backup for a client node takes longer than subsequent incremental backups.

You can balance the server's scheduled workload by:

- Adjusting the number of sessions that the server allocates to scheduled operations
- Randomizing scheduled start time for client operations (if clients use client-polling scheduling mode)
- Increasing the length of the startup window

Setting the number of sessions the server allocates to scheduled operations

The maximum number of concurrent client/server sessions is defined by the MAXSESSIONS server option.

Of these sessions, you can set a maximum percentage to be available for processing scheduled operations. Limiting the number of sessions available for scheduled operations ensures that sessions are available when users initiate any unscheduled operations, such as restoring file or retrieving files.

If the number of sessions for scheduled operations is insufficient, you can increase either the total number of sessions or the maximum percentage of scheduled sessions. However, increasing the total number of sessions can adversely affect server performance. Increasing the maximum percentage of scheduled sessions can reduce the server availability to process unscheduled operations.

For example, assume that the maximum number of sessions between client nodes and the server is 80. If you want 25% of these sessions to be used by for scheduled operations, enter:

```
set maxschedsessions 25
```

The server then allows a maximum of 20 sessions to be used for scheduled operations.

The following table shows the trade-offs of using either the SET MAXSCHEDESESSIONS command or the MAXSESSIONS server option.

An administrator can...	Using...	With the result
Increase the total number of sessions	MAXSESSIONS server option	May adversely affect the server's performance
Increase the total number of sessions allocated to scheduled operations	SET MAXSCHEDESESSIONS command	May reduce the server's ability to process unscheduled operations

For information about the MAXSESSIONS option and the SET MAXSCHEDESESSIONS command, refer to *Administrator's Reference*.

Randomizing schedule start times

To randomize start times for schedules means to scatter each schedule's start time across its startup window. A startup window is defined by the start time and duration during which a schedule must be initiated.

For example, if the start time is 1:00 a.m. and the duration is 4 hours, the startup window is 1:00 a.m. to 5:00 a.m. For the client-polling scheduling mode, you can specify the percentage of the startup window that the server can use to randomize start times for different client nodes associated with a schedule.

If you set randomization to 0, no randomization occurs. This process can result in communication errors if many client nodes try to contact the server at the same instant.

The settings for randomization and the maximum percentage of scheduled sessions can affect whether schedules are successfully completed for client nodes. Users receive a message if all sessions are in use when they attempt to process a schedule. If this happens, you can increase randomization and the percentage of scheduled sessions allowed to make sure that the server can handle the workload. The maximum percentage of randomization allowed is 50%. This limit ensures that half of the startup window is available for retrying scheduled commands that have failed.

To set randomization to 50%, enter:

```
set randomize 50
```

It is possible, especially after a client node or the server has been restarted, that a client node may not poll the server until *after* the beginning of the startup window in which the next scheduled event is to start. In this case, the starting time is randomized over the specified percentage of the *remaining* duration of the startup window.

Consider the following situation:

- The schedule start time is 8:00 a.m. and its duration is 1 hour. Therefore the startup window for the event is from 8:00 to 9:00 a.m.
- Ten client nodes are associated with the schedule.
- Randomization is set to 50%.
- Nine client nodes poll the server before 8:00 a.m.
- One client node does not poll the server until 8:30 a.m.

The result is that the nine client nodes that polled the server *before* the beginning of the startup window are assigned randomly selected starting times between 8:00 and 8:30. The client node that polled at 8:30 receives a randomly selected starting time that is between 8:30 and 8:45.

Increasing the length of the schedule startup window

Increasing the size of the startup window (by increasing the schedule's duration) can also affect whether a schedule completes successfully.

A larger startup window gives the client node more time to attempt initiation of a session with the server.

Controlling how often client nodes contact the server

To control how often client nodes contact the server to perform a scheduled operation, an administrator can set the frequency for certain events.

- How often nodes query the server
- The number of command retry attempts
- The amount of time between retry attempts

Users can also set these values in their client user options files. (Root users on UNIX and Linux systems set the values in client system options files.) However, user values are overridden by the values that the administrator specifies on the server.

The communication paths from client node to server can vary widely with regard to response time or the number of gateways. In such cases, you can choose *not* to set these values so that users can tailor them for their own needs.

Related tasks

“Setting how often clients query the server”

“Setting the number of command retry attempts”

“Setting the amount of time between retry attempts” on page 543

Setting how often clients query the server

When scheduling client nodes with client-polling scheduling, you can specify how often the nodes query the server for a schedule. If nodes poll frequently for schedules, changes to scheduling information (through administrator commands) are propagated more quickly to the nodes. However, increased polling by client nodes also increases network traffic.

For the client-polling scheduling mode, you can specify the maximum number of hours that the scheduler on a client node waits between attempts to contact the server to obtain a schedule. You can set this period to correspond to the frequency with which the schedule changes are being made. If client nodes poll more frequently for schedules, changes to scheduling information (through administrator commands) are propagated more quickly to client nodes.

If you want to have all clients using polling mode contact the server every 24 hours, enter:

```
set querieschedperiod 24
```

This setting has no effect on clients that use the server-prompted scheduling mode.

The clients also have a QUERYSCHEDPERIOD option that can be set on each client. The server value overrides the client value once the client successfully contacts the server.

Setting the number of command retry attempts

You can specify the maximum number of times the scheduler on a client node can retry a scheduled command that fails.

The maximum number of command retry attempts does not limit the number of times that the client node can contact the server to obtain a schedule. The client node never gives up when trying to query the server for the next schedule.

Be sure not to specify so many retry attempts that the total retry time is longer than the average startup window.

If you want to have all client schedulers retry a failed attempt to process a scheduled command up to two times, enter:

```
set maxcmdretries 2
```

Maximum command retries can also be set on each client with a client option, MAXCMDRETRIES. The server value overrides the client value once the client successfully contacts the server.

Setting the amount of time between retry attempts

You can specify the length of time that the scheduler waits between command retry attempts. Command retry attempts occur when a client node is unsuccessful in establishing a session with the server or when a scheduled command fails to process.

Typically, this setting is effective when set to half of the estimated time it takes to process an average schedule. If you want to have the client scheduler retry every 15 minutes any failed attempts to either contact the server or process scheduled commands, enter:

```
set retryperiod 15
```

You can use this setting in conjunction with the SET MAXCMDRETRIES command (number of command retry attempts) to control when a client node contacts the server to process a failed command. See “Setting the number of command retry attempts” on page 542.

The retry period can also be set on each client with a client option, RETRYPERIOD. The server value overrides the client value once the client successfully contacts the server.

Specifying one-time actions for client nodes

You can use the DEFINE CLIENTACTION command to specify that one or more client nodes perform a one-time action if the client schedulers are active.

If the scheduling mode is set to prompted, the client performs the action within 3 to 10 minutes. If the scheduling mode is set to polling, the client processes the command at its prescribed time interval. The time interval is set by the QUERYSCHEDPERIOD client option. The DEFINE CLIENTACTION command causes Tivoli Storage Manager to automatically define a schedule and associate client nodes with that schedule. With the schedule name provided, you can later query or delete the schedule and associated nodes. The names of one-time client action schedules can be identified by a special character followed by numerals, for example @1.

The schedule name and association information is returned to the server console or the administrative client with messages ANR2500I and ANR2510I.

For example, you can issue a DEFINE CLIENTACTION command that specifies an incremental backup command for client node HERMIONE in domain ENGPOLDOM:

```
define clientaction hermione domain=engpoldom action=incremental
```

Tivoli Storage Manager defines a schedule and associates client node HERMIONE with the schedule. The server assigns the schedule priority 1, sets the period units (PERUNITS) to ONETIME, and determines the number of days to keep the schedule active based on the value set with SET CLIENTACTDURATION command.

For a list of valid actions, see the DEFINE CLIENTACTION command in the *Administrator's Reference*. You can optionally include the OPTIONS and OBJECTS parameters.

Determining how long the one-time schedule remains active

You can determine how long schedules that were defined via `DEFINE CLIENTACTION` commands remain active by using the `SET CLIENTACTDURATION` command.

This `SET CLIENTACTDURATION` command allows you to specify the number of days that schedules that were created with the `DEFINE CLIENTACTION` command are active. These schedules are automatically removed from the database whether the associated nodes have processed the schedule or not, after the specified number of days. The following example specifies that schedules for client actions be active for 3 days:

```
set clientactduration 3
```

If the duration of client actions is set to zero, the server sets the `DURUNITS` parameter (duration units) as indefinite for schedules defined with `DEFINE CLIENTACTION` command. The indefinite setting for `DURUNITS` means that the schedules are not deleted from the database.

Part 4. Maintaining the server

To help you maintain server operations, Tivoli Storage Manager allows you to automate tasks that should occur regularly, monitor processes, and ensure the availability and integrity of the database. In addition , Tivoli Storage Manager provides the tools needed to set up and manage a network of servers, and to move data from one server to another.

Chapter 18. Managing servers with the Administration Center

The Administration Center is a Web-based interface for centrally configuring and managing IBM Tivoli Storage Manager servers. It provides wizards to help guide you through common configuration tasks. Properties notebooks allow you to modify settings and perform advanced management tasks.

The Administration Center is installed as an IBM Tivoli Integrated Portal component. The Tivoli Integrated Portal allows you to install components provided by multiple IBM applications, and access them from a single interface.

In Tivoli Storage Manager Version 6.2, the Administration Center cannot be installed on HP-UX, but it can be used to manage HP-UX servers.

For Administration Center system requirements, see the following Web site:
<http://www.ibm.com/support/docview.wss?uid=swg21410467>

Using the Administration Center

The Administration Center is installed as a component of the Tivoli Integrated Portal. You can use the Administration Center to centrally configure and manage your IBM Tivoli Storage Manager environment.

Basic items (for example, server maintenance, storage devices, and so on) are listed in the navigation tree on the left side of the Tivoli Integrated Portal. When you click on an item, a work page containing one or more portlets (for example, the Servers portlet) is displayed in the work area on the right side of the interface. You use portlets to perform individual tasks, such as creating storage pools.

Each time you click an item in the navigation tree, a new work page is opened. This allows you to open the same item for more than one server. To navigate among open items, use the page bar at the top of the work area.

Many portlets contain tables. These tables display objects like servers, policy domains, or reports. There are two ways to work with table objects. For any table object, you can do the following:

1. Click its radio button or check box in the **Select** column.
2. Click **Select Action** to display the table action list.
3. Select an action to perform that action.

For some table objects, you can also click the object name to open a portlet or work page pertaining to it. In most cases, a properties notebook portlet is opened. This provides a fast way to work with table objects.

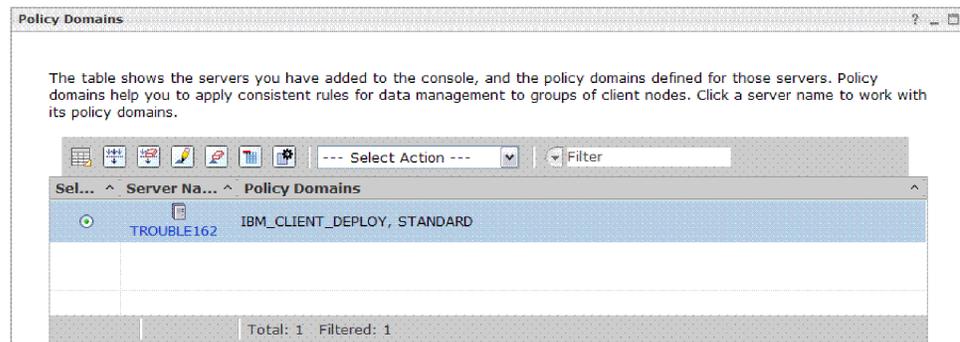
Fields marked with an asterisk and highlighted in yellow require an entry or selection. However, if you have the Google search bar installed in your browser, some fields can display bright yellow, whether they are required or not. To get help at any time, click the context sensitive help button  in the title bar of a portlet, properties notebook, and so on.

If you want more space in the work area, you can hide the navigation tree by clicking 

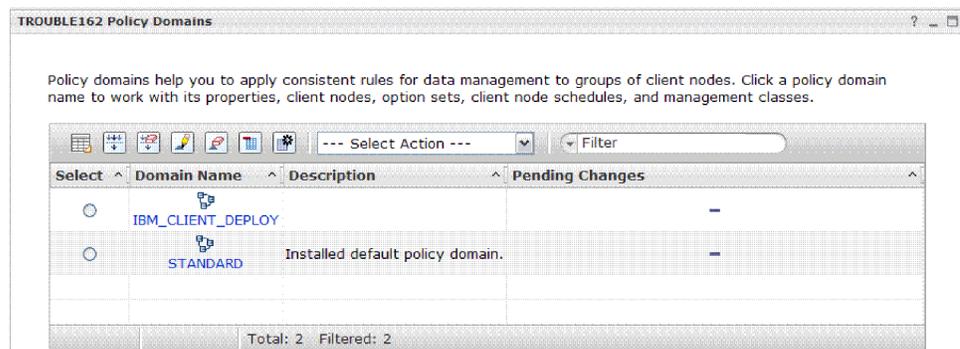
Do not use the **Back**, **Forward** and **Refresh** buttons in your browser. Doing so can cause unexpected results. Using your keyboard's **Enter** key can also cause unexpected results. Use the controls in the Administration Center interface instead.

This following simple task will help familiarize you with Administration Center controls. Suppose you want to create a new client node and add it to the STANDARD policy domain associated with a particular server.

1. If you have not already done so, access the Administration Center by entering the following address in a supported Web browser: `https://workstation_name:9043/ibm/console`. The *workstation_name* is the network name or IP address of the workstation on which you installed the Administration Center. The default Web administration port (HTTPS) is 9043. To get started, log in using the Tivoli Integrated Portal user ID and password that you created during the installation. Save this password in a safe location because you need it not only to log in but also to uninstall the Administration Center.
2. Click **Tivoli Storage Manager**, and then click **Policy Domains** in the navigation tree. The Policy Domains work page is displayed with a table that lists the servers that are accessible from the Administration Center. The table also lists the policy domains defined for each server:



3. In the **Server Name** column of the **Policy Domains** table, click the name of the server with the STANDARD domain to which you want to add a client node. A portlet is displayed with a table that lists the policy domains created for that server:



4. In the **Domain Name** column of the server's **Policy Domains** table, click the STANDARD policy domain. The STANDARD Properties portlet is displayed:



- In the domain's properties portlet, click **Client Nodes**. A table is displayed listing all the nodes assigned to the STANDARD policy domain:

Select ^	Name ^	Type ^	Platform ^
<input type="radio"/>	MSEG	Client	WinNT
<input type="radio"/>	NODE1	Client	(?)
<input type="radio"/>	NODE2	Client	(?)

Page 1 of 1 Total: 3 Filtered: 3 Displayed: 3 Selected: 0

- In the client nodes table, click **Select Action**, and then select **Create a Client Node**. The Create Client Node wizard is displayed:

Create a client node by accepting the default settings or by entering new information. You must enter a client node name and a password. Click OK to create a node and return to Client Nodes and Backup sets or click Add Another to create a node and save all entries to a new form. To edit the default settings, click the pencil icon in the upper right-hand corner of this portlet.

Server: Policy domain:

*Name:

*Password:

*Confirm password:

Contact:

Web address:

▶ Policy Settings

▶ Security Settings

▶ Memberships

Add the following parameters to the generated command:

OK Add Another Cancel

- Follow the instructions in the wizard. After you complete the wizard, the name of the new client node is displayed in the **Client Nodes** table for the STANDARD policy domain.

Starting and stopping the Administration Center

You can start and stop the Tivoli Storage Manager Administration Center server by using the supplied commands.

In the following task descriptions, *<TIP_HOME>* is the root directory for your Tivoli Integrated Portal installation and *<tip_admin>* and *<tip_pw>* are a valid Tivoli Integrated Portal user ID and password.

The default *<TIP_HOME>* location is */opt/ibm/ac*. To start the Administration Center from a command line, go to the *<TIP_HOME>/bin* directory or a subdirectory of the Tivoli Storage Manager installation directory and issue the following command: `startServer.sh server1`.

To stop the Administration Center from a command line, go to the *<TIP_HOME>/bin* directory or a subdirectory of the Tivoli Storage Manager installation directory and issue the following command: `stopServer.sh server1 -username tip_admin -password tip_pw`. Alternatively, you can issue the `stopServer.sh server1` command and you are prompted for your username and password.

Functions not in the Administration Center

The Administration Center offers the functions of most administrative commands, as well as unique functions such as the health monitor and wizards to help you perform complex tasks. However, some Tivoli Storage Manager functions are limited or not supported in the Administration Center.

The following table shows commands that are supported with some restrictions or not yet supported in the Administration Center. Use the command line if the command or command parameter that you need is not available in the Administration Center.

Command	Supported in the Administration Center
ACCEPT DATE	No
AUDIT LICENSES	No
BEGIN EVENTLOGGING	No
CANCEL EXPIRATION	No
CANCEL MOUNT	No
CANCEL RESTORE	No
CONVERT ARCHIVE	No
COPY DOMAIN	No
COPY MGMTCLASS	No
COPY POLICYSET	No
COPY PROFILE	No
COPY SCHEDULE	No
COPY SCRIPT	No
COPY SERVERGROUP	No

Command	Supported in the Administration Center
DEFINE COPYGROUP TYPE=ARCHIVE	Supported except for these parameters: <ul style="list-style-type: none"> • RETINIT • RETMIN These parameters are needed only to support IBM Total Storage Archive Manager.
DEFINE EVENTSERVER	No
DEFINE STGPOOL	Supported except for the RECLAMATIONTYPE parameter This parameter is needed only for support of EMC Centera devices.
DELETE DATAMOVER	No
DELETE DISK	No
DELETE EVENT	No
DELETE EVENTSERVER	No
DELETE SUBSCRIBER	No
DISABLE EVENTS	No
DISMOUNT DEVICE	No
DISPLAY OBJNAME	No
ENABLE EVENTS	No
Event logging commands (BEGIN EVENTLOGGING, END EVENTLOGGING, ENABLE EVENTS, DISABLE EVENTS)	No Some SNMP options can be viewed in the interface, in a server's properties notebook.
MOVE GRPMEMBER	No
QUERY AUDITOCCUPANCY	No
QUERY ENABLED	No
QUERY EVENTRULES	No
QUERY EVENTSERVER	No
QUERY LICENSE	No
QUERY NASBACKUP	No
QUERY RESTORE	No
QUERY SSLKEYRINGPW	No
QUERY SYSTEM	No
QUERY TAPEALERTMSG	No
RECONCILE VOLUMES	No
REGISTER LICENSE	No
RENAME FILESPACE	No
RESTORE STGPOOL	No
RESTORE VOLUME	Yes, except use the command line to restore random-access storage pool volumes.
SET ACCOUNTING	No
SET ACTLOGRETENTION	No

Command	Supported in the Administration Center
SET ARCHIVERETENTIONPROTECTION	No
SET CLIENTACTDURATION	No
SET CONTEXTMESSAGING	No
SET DBREPORTMODE	No
SET DEDUPVERIFICATIONLEVEL	No
SET EVENTRETENTION	No
SET LICENSEAUDITPERIOD	No
SET MAXCMDRETRIES	No
SET MAXSCHEDESESSIONS	No
SET QUERYSCHEDPERIOD	No
SET RANDOMIZE	No
SET RETRYPERIOD	No
SET SCHEDMODES	No
SET SERVERNAME	No
SET SSLKEYRINGPW	No
SET SUBFILE	No
SET SUMMARYRETENTION	No
SET TAPEALERTMSG	No
SET TOCLOADRETENTION	No
SETOPT	Only the following server options can be modified using the Administration Center: <ul style="list-style-type: none"> • EXPINTERVAL • RESTOREINTERVAL
UPDATE DISK	No
UPDATE DRIVE (<i>FILE type</i>)	No
UPDATE LIBRARY (<i>FILE type</i>)	No
UPDATE POLICYSET	No
VALIDATE LANFREE	Use the Enable LAN-free Data Movement wizard to get this function.

Protecting the Administration Center

The Administration Center is installed as a Tivoli Integrated Portal plug-in. To protect your Administration Center configuration settings, use the Tivoli Storage Manager backup-archive client to back up the Tivoli Integrated Portal.

Backing up the Administration Center

To back up the Administration Center, the Tivoli Storage Manager backup-archive client must be installed on the Tivoli Integrated Portal system. It must then be configured to back up to a Tivoli Storage Manager server.

For more information about backup operations, see the *Backup-Archive Client Installation and User's Guide*.

In the following task description, <TIP_HOME> is the root directory for your Tivoli Integrated Portal installation.

The <TIP_HOME> default location is /opt/ibm/ac.

To back up the Administration Center:

1. Stop the Tivoli Integrated Portal. See "Starting and stopping the Administration Center" on page 550 for the command syntax.
2. Using the backup-archive client, back up the Tivoli Integrated Portal installation directory.
For example: back up <TIP_HOME>/bin.
3. Start the Tivoli Integrated Portal. See "Starting and stopping the Administration Center" on page 550 for the command syntax.

Restoring the Administration Center

To restore the Tivoli Integrated Portal the Tivoli Storage Manager backup-archive client must be installed on the Tivoli Integrated Portal system. The backup-archive client must then be configured to restore from the Tivoli Storage Manager server that was used to back up the Tivoli Integrated Portal.

To restore the Administration Center, perform the following steps:

1. If necessary, restore the operating system and reinstall the Tivoli Storage Manager backup-archive client.
2. Reinstall the Tivoli Integrated Portal and the Administration Center. For more information, see the *Installation Guide*.

Note: In Tivoli Storage Manager Version 6.2, the Administration Center cannot be installed on HP-UX, but it can be used to manage HP-UX servers. For Administration Center system requirements, see the following Web site: <http://www.ibm.com/support/docview.wss?uid=swg21410467>.

3. Stop the Tivoli Integrated Portal. See "Starting and stopping the Administration Center" on page 550 for the command syntax.
4. Use the Tivoli Storage Manager backup-archive client to restore the Tivoli Integrated Portal to the same location where it was originally installed.
5. Start the Tivoli Integrated Portal. See "Starting and stopping the Administration Center" on page 550 for the command syntax.

Chapter 19. Managing server operations

Administrators can perform such server operations as licensing purchased features, starting and halting the server, and monitoring server information.

See the following topics:

Tasks:
"Licensing IBM Tivoli Storage Manager"
"Starting the Tivoli Storage Manager server" on page 557
"Moving the Tivoli Storage Manager server to another system" on page 562
"Date and time on the server" on page 563
"Managing server processes" on page 564
"Preemption of client or server operations" on page 566
"Setting the server name" on page 567
"Adding or updating server options" on page 568
"Getting help on commands and error messages" on page 569
"Setting the server name" on page 567

Licensing IBM Tivoli Storage Manager

There are tasks involved when licensing an IBM Tivoli Storage Manager system, including registering, saving and auditing.

Task	Required Privilege Class
Register licenses Audit licenses	System
Display license information	Any administrator

For current information about supported clients and devices, visit the IBM Tivoli Storage Manager home page at http://www.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Storage_Manager.

The base IBM Tivoli Storage Manager feature includes the following support:

- An unlimited number of administrative clients.
- Enterprise Administration, which includes: command routing, enterprise configuration, and enterprise logging (server-to-server).
- Server-to-server Virtual Volume capabilities (does not include database and storage pool backup).
- Network Enabler (network connections for clients).
- AFS/DFS Support, (the S/390[®] platform includes the S/390 UNIX client as part of Managed System for SAN).

Registering licensed features

You must register a new license if you want to add support for any of the features that are not already in your existing license agreement. Tivoli Storage Manager uses a license file and the REGISTER LICENSE command to complete this task.

Licenses are stored in enrollment certificate files, which contain licensing information for the server product. The enrollment certificate files are on the installation CD-ROM. When registered, the licenses are stored in a `nodelock` file within the current directory.

To register a license, you must issue the REGISTER LICENSE command. The command will register new licenses for server components, including Tivoli Storage Manager (base), Tivoli Storage Manager Extended Edition, and System Storage Archive Manager. You must specify the name of the enrollment certificate file containing the license to be registered when you issue the REGISTER LICENSE command. The file specification can contain a wildcard character (*). The following are possible certificate file names:

tsmbasic.lic

Registers IBM Tivoli Storage Manager base edition.

tsmee.lic

Registers IBM Tivoli Storage Manager Extended Edition. This includes the disaster recovery manager, large libraries, and NDMP.

dataret.lic

Registers the System Storage Archive Manager. This is required to enable Data Retention Protection as well as Expiration and Deletion Suspension (Deletion Hold).

***.lic** registers all IBM Tivoli Storage Manager licenses for server components.

Note:

1. To unregister licenses, you must erase the `nodelock` file found in the server directory of your installation. This will also require you to reregister any previously registered licenses.
2. You cannot register licenses for components that are licensed on the basis of processors (for example, Tivoli Storage Manager for Mail, Tivoli Storage Manager for databases, Tivoli Storage Manager for Enterprise Resource Planning, Tivoli Storage Manager for Hardware, and Tivoli Storage Manager for Space Management.)

Attention:

- Save the installation media that contains your enrollment certificate files. You may need to register your licenses again for any of the following reasons:
 - The server is corrupted.
 - The server is moved to a different machine.
 - The **nodelock** file is destroyed or corrupted. IBM Tivoli Storage Manager stores license information in the `nodelock` file, which is located in the directory from which the server is started.
- IBM Tivoli Storage Manager licenses are associated with the CPU chip of the machine on which IBM Tivoli Storage Manager is installed. If you change that CPU chip, you must first erase the existing `nodelock` files and then reregister all your licenses.

Monitoring licenses

When license terms change (for example, a new license is specified for the server), the server conducts an audit to determine if the current server configuration conforms to the license terms. The server also periodically audits compliance with license terms. The results of an audit are used to check and enforce license terms.

If 30 days have elapsed since the previous license audit, the administrator cannot cancel the audit. If an IBM Tivoli Storage Manager system exceeds the terms of its license agreement, one of the following occurs:

- The server issues a warning message indicating that it is not in compliance with the licensing terms.
- If you are running in Try Buy mode, operations fail because the server is not licensed for specific features.

You must contact your IBM Tivoli Storage Manager account representative to modify your agreement.

An administrator can monitor license compliance by:

Auditing licenses

Use the `AUDIT LICENSES` command to compare the current configuration with the current licenses.

Note: During a license audit, the server calculates, by node, the amount of backup, archive, and space management storage in use. This calculation can take a great deal of CPU time and can stall other server activity. Use the `AUDITSTORAGE` server option to specify that storage is not to be calculated as part of a license audit.

Displaying license information

Use the `QUERY LICENSE` command to display details of your current licenses and determine licensing compliance.

Scheduling automatic license audits

Use the `SET LICENSEAUDITPERIOD` command to specify the number of days between automatic audits.

Starting the Tivoli Storage Manager server

There are several methods for starting the server.

Task	Required Privilege Class
Start, halt, and restart the server	System or operator

The following events occur when you start or restart the IBM Tivoli Storage Manager server:

- The server invokes the communication methods specified in the server options file.
- The server starts an IBM Tivoli Storage Manager server console session that is used to operate and administer the server until administrative clients are registered to the server.
- The server uses the `STANDARD` policy that is shipped with IBM Tivoli Storage Manager.

Starting the server on AIX, Linux, and UNIX

When the server is started, Tivoli Storage Manager displays system information.

Upon startup, the following information is displayed:

- Product licensing and copyright information
- Processing information about the server options file
- Communication protocol information
- Database and recovery log information
- Storage pool volume information
- Server generation date
- Progress messages and any errors encountered during server initialization

The date and time check occur when the server is started and once each hour thereafter. An invalid date is one that is:

- Earlier than the server installation date and time
- More than one hour earlier than the last time the date was checked
- More than 30 days later than the last time the date was checked

You can start the server from the instance ID or from the root user ID.

See the *Installation Guide* for more information.

Automating server startup

You can configure the server to start automatically at startup. To configure the Tivoli Storage Manager server, use the `rc.dsmserv` script.

The `rc.dsmserv` script is located in `/opt/tivoli/tsm/server/bin`.

Tip: If you used either the upgrade wizard or the configuration wizard, you had the choice of starting the upgraded server automatically when the system is restarted. If you selected that choice, the startup of the server was added to the `/etc/inittab` file.

If you did not use a wizard to configure the IBM Tivoli Storage Manager server, add an entry to the `/etc/inittab` file for each server that you want to automatically start.

- Set the run level to the value that corresponds to multiuser mode, with networking enabled. Typically, the run level to use is 2, 3, or 5, depending on the operating system and its configuration. Consult the documentation for your operating system for details on run level.
- On the `rc.dsmserv` command, specify the instance owner name with the `-u` option and the location of the server instance directory with the `-i` option.

Verify that the correct syntax is correct by consulting the documentation of your operating system.

For example, if the instance owner is `tsminst1` and the server instance directory is `/home/tsminst1/inst1`, add the following entry to `/etc/inittab` on one line:

```
tsm1:2: once:/opt/tivoli/tsm/server/bin/rc.dsmserv -u tsminst1 -i  
/home/tsminst1/tsminst1 -q >/dev/console 2>&1
```

If you have two servers that you want to run (`tsminst1` and `tsminst2`), add the following to `/etc/inittab`, on one line:

```
tsm1:2: once:/opt/tivoli/tsm/server/bin/rc.dsmserv -u tsminst1 -i
/home/tsminst1/tsminst1 -q >/dev/console 2>&1
tsm2:2: once:/opt/tivoli/tsm/server/bin/rc.dsmserv -u tsminst2 -i
/home/tsminst2/tsminst2 -q >/dev/console 2>&1
```

For more information on the `rc.dsmserv` script, see the *Administrator's Reference*.

Stand-alone mode for server startup

Some Tivoli Storage Manager server operations require that you start the server in single user, stand-alone mode. Starting the server in this way is typically done when performing maintenance and performing a major reconfiguration of the Tivoli Storage Manager server.

Here are some examples of operations that require starting the server in stand-alone mode:

- Verifying the Tivoli Storage Manager server operations after completing a server upgrade.
- Verifying the Tivoli Storage Manager server operations after performing one of the following operations:
 - Restoring the server database by using the `DSMSERV RESTORE DB` command.
 - Dumping, reinitializing, and reloading the server database if a catastrophic error occurs (recovery log corruption, for example), and if the `DSMSERV RESTORE DB` command cannot be used.
- Running Tivoli Storage Manager recovery utilities when asked by IBM Customer Support.

To perform these tasks, you should disable the following server activities:

- All administrative sessions
- All client sessions
- All scheduled operations
- HSM client migration
- Storage pool migration
- Storage pool reclamation
- Client file expiration

Starting the server in stand-alone mode

To start a Tivoli Storage Manager server in stand-alone mode, follow this procedure.

1. Edit the `dsmserv.opt` file and add the following server options:
 - `NOMIGRRECL`
This option disables reclamation and migration for the server.
 - `DISABLESCHEDS YES`
This option specifies that administrative and client schedules are disabled during Tivoli Storage Manager server recovery.
 - `EXPINTERVAL 0`
This option prevents automatic inventory expiration of client files.
2. Start the server as described in “Starting the Tivoli Storage Manager server” on page 557.
3. Prevent new client sessions, administrative sessions, and server-to-server sessions by issuing the following command:

```
disable sessions all
```

Note: You can continue to access the server at this point. Any current client activities complete unless a user logs off or you cancel a client session.

4. At this point you can perform the maintenance, reconfiguration, or recovery operations, and the halt the server.

To restart the server after completing the operations, follow this procedure:

1. Edit the *dsmserv.opt* file to return the server options to their original settings.
2. Start the server as described in “Starting the Tivoli Storage Manager server” on page 557.
3. Enable client sessions, administrative sessions, and server-to-server sessions by issuing the following command:

```
enable sessions all
```

Starting the server in other modes

You can use IBM Tivoli Storage Manager command options to specify how to start the server in other modes as part of the `dsmserv` command.

For example:

```
dsmserv option
```

Where *option* can be any one of the following:

- q Starts the server as a daemon program. The server runs as a background process, and does not read commands from the server console. Output messages are directed to the `SERVER_CONSOLE`.

Note: Before issuing this command, you must have an administrative client registered and authorized with system authority. The administrative client must be started. Otherwise, the server will run in the quiet mode and you will not be able to access the server.

- o *filename*

Specifies an explicit options file name when running more than one server.

Running multiple server instances on a single system

A server instance runs the server program using its unique database, recovery log, and server options. To run multiple server instances on a single system, set up separate database and recovery log directories, and an instance directory for each server to contain the server options file and other files that are needed to run each server instance.

Each server instance requires a unique user ID that is the instance owner.

The files for one instance of the server are stored separately from those used by another server instance on the same system, and separately from the server program files.

As part of server configuration, you create a directory to store the files for the server instance. The following files are stored in the instance directory:

- The server options file, `dsmserv.opt`
- The device configuration file, if the `DEVCONFIG` server option does not specify a fully qualified name

- The volume history file, if the VOLUMEHISTORY server option does not specify a fully qualified name
- Volumes for **DEVTYPE=FILE** storage pools, if the directory for the device class is not fully specified, or not fully qualified
- The dsmserv.v6lock file
- User exits
- Trace output (if not fully qualified)

Database and recovery log files are stored in separate directories, not in the instance directory.

To manage the system memory that is used by each server on a system, use the DBMEMPERCENT server option to limit the percentage of system memory that can be used by the database manager of each server. If all servers are equally important, use the same value for each server. If one server is a production server and other servers are test servers, set the value for the production server to a higher value than the test servers.

For example, to run two server instances, tsminst1 and tsminst2, create instance directories such as /home/tsminst1 and /home/tsminst2. In each directory, place the dsmserv.opt file for that server. Each dsmserv.opt file must specify a different port for the server to use.

To automatically start the two server instances, you can use the script, rc.dsmserv.

For more information about configuring server instances, see the following information in the applicable *Installation Guide*:

- Configuring Tivoli Storage Manager using the configuration wizard or the Administration Center

Halting the server

You can halt the server without warning if an unplanned operating system problem requires the server to be stopped.

Task	Required Privilege Class
Start, halt, and restart the server	System or operator

When you halt the server, all processes are abruptly stopped and client sessions are canceled, even if they are not complete. Any in-progress transactions are rolled back when the server is restarted. Administrator activity is not possible.

If possible, halt the server only after current administrative and client node sessions have completed or canceled. To shut down the server without severely impacting administrative and client node activity with the server, you must:

1. Disable the server to prevent new client node sessions from starting by issuing the DISABLE SESSIONS command. This command does not cancel sessions currently in progress or system processes like migration and reclamation.
2. Notify any existing administrative and client node sessions that you plan to shut down the server. The server does not provide a network notification facility; you must use external means to notify users.
3. Cancel any existing administrative or client node sessions by issuing the CANCEL SESSION command and the associated session number. To obtain

session numbers and determine if any sessions are running, use the QUERY SESSION command. If a session is running, a table will appear showing the session number on the far left side of the screen.

4. Find out if any other processes are running, such as server migration or inventory expiration, by using the QUERY PROCESS command. If a database backup process is running, allow it to complete before halting the server. If other types of processes are running, cancel them by using the CANCEL PROCESS command.

Note: If the process you want to cancel is currently waiting for a tape volume to be mounted (for example, a process initiated by EXPORT, IMPORT, or MOVE DATA commands), the mount request is automatically cancelled. If a volume associated with the process is currently being mounted by an *automated* library, the cancel may not take effect until the mount is complete.

5. Halt the server to shut down all server operations by using the HALT command.

Note:

1. The HALT command can be replicated using the ALIASHALT server option. The server option allows you to define a term other than HALT that will perform the same function. The HALT command will still function, however the server option provides an additional method for issuing the HALT command.
2. In order for the administrative client to recognize an alias for the HALT command, the client must be started with the CHECKALIASHALT option specified. See the *Administrator's Reference* for more information.

Stopping the server when running as a background process

If you started the server as a background process and want to stop the server, connect to the server as an administrative client and issue the HALT command.

If you cannot connect to the server with an administrative client and you want to stop the server, you must cancel the process by using the **kill** command with the process ID number (pid) that is displayed at initialization.

Note: Before you issue the **kill** command, ensure that you know the correct process ID for the IBM Tivoli Storage Manager server.

The dsmserv.lock file, in the directory from which the server is running, can be used to identify the process ID of the process to kill. To display the file enter:

```
cat /opt/tivoli/tsm/server/bin/dsmserv.lock
```

Moving the Tivoli Storage Manager server to another system

You can move your Tivoli Storage Manager server from one computer to another.

These are the prerequisites to back up the database from one server and restore it to another server:

- The same operating system must be running on both servers.
- The sequential storage pool that you use to back up the server database must be accessible from both servers. Only manual and SCSI library types are supported for the restore operation.

- The restore operation must be done by a Tivoli Storage Manager server at a code level that is the same as that on the server that was backed up.

To move the database to another system perform the following steps:

1. Install Tivoli Storage Manager on the target server. See the *Installation Guide* for details.
2. Back up the database to sequential media. For example, issue the following command:

```
backup db devclass=1to4 type=full
```

The sequential storage pool that you use to back up the server database must be accessible from both servers.

3. Halt the server.
4. Move any libraries and devices from the original server to the new server, or ensure that they are accessible through a storage area network.
5. Move copies of the volume history file, device configuration file, and server options file to the target server.
6. Restore the backed up database on the target server. For example:

- To maintain the current directory structure on the target server, issue this command:

```
dsmserv restore db
```

- To change the current directory structure on the target server, create a file (for example `dbdir.txt`), list the directories that are to be restored on separate lines, and issue this command:

```
dsmserv restore db on=dbdir.txt
```

7. Start the target server.

Related tasks

“Moving the database and recovery log on a server” on page 607

Date and time on the server

The date and time on the server must be correct. If the server detects an invalid date or time, server sessions become disabled.

Every time the server is started and for each hour thereafter, a date and time check occurs. An invalid date can be one of the following:

- Earlier than the server installation date and time.
- More than one hour earlier than the last time the date was checked.
- More than 30 days later than the last time the date was checked.

An error message (ANR0110E) is displayed and expiration, migration, reclamation, and volume history deletion operations are not allowed. You may either change the system date if it is in error, or issue the `ACCEPT DATE` command to force the server to accept the current system date as valid. Use the `ENABLE SESSIONS` command after you issue the `ACCEPT DATE` command to re-enable the server for client node activity.

Managing server processes

When a user or administrator issues a IBM Tivoli Storage Manager command or uses a graphical user interface to perform an operation, the server starts a process. Some examples of an operation are registering a client node, deleting a management class, or canceling a client session.

Task	Required Privilege Class
Display information about a server background process	Any administrator
Cancel a server process	System

Most processes occur quickly and are run in the foreground, but others that take longer to complete run as background processes.

The server runs the following operations as background processes:

- Auditing licenses
- Auditing a volume
- Backing up the database
- Backing up a storage pool
- Deleting a file space
- Deleting a storage volume
- Expiring the inventory
- Exporting or importing data
- Generating a backup set
- Migrating files from one storage pool to the next storage pool
- Moving data from a storage volume
- Reclaiming space from tape storage volumes
- Restoring a storage pool
- Restoring a volume

Note: To prevent contention for the same tapes, the server does not allow a reclamation process to start if a DELETE FILESPACE process is active. The server checks every hour for whether the DELETE FILESPACE process has completed so that the reclamation process can start. After the DELETE FILESPACE process has completed, reclamation begins within one hour.

The server assigns each background process an ID number and displays the process ID when the operation starts. This process ID number is used for tracking purposes. For example, if you issue an EXPORT NODE command, the server displays a message similar to the following:

```
EXPORT NODE started as Process 10
```

Some of these processes can also be run in the foreground by using the WAIT=YES parameter when you issue the command from an administrative client. See *Administrator's Reference* for details.

Requesting information about server processes

You can request information about server background processes. If you know the process ID number, you can use the number to limit the search.

If you do not know the process ID, you can display information about all background processes by entering:

```
query process
```

The following figure shows a server background process report after a DELETE FILESPACE command was issued. The report displays a process ID number, a description, and a completion status for each background process.

Process Number	Process Description	Status
2	DELETE FILESPACE	Deleting filespace DRIVE_D for node CLIENT1: 172 files deleted.

Canceling server processes

You can cancel a server background process by specifying its ID number and issuing the CANCEL PROCESS command.

To find the process number, issue the QUERY PROCESS command . For details, see “Requesting information about server processes.”

When canceling processes, the following conditions apply:

- If a process is currently waiting for a tape volume to be mounted (for example, a process initiated by EXPORT, IMPORT, or MOVE DATA commands), the mount request is automatically canceled.
- If a volume associated with a process is currently being mounted by an *automated* library, the cancel may not take effect until the mount is complete.
- If a process has a pending mount request, the process might not respond to a CANCEL PROCESS command until the mount request has been answered, cancelled, or timed out. (For example, reclamation automatically generates mount requests as part of the process.)

To answer a mount request, issue the REPLY REQUEST command. To cancel a mount request, issue the CANCEL REQUEST command.

Note:

1. To list open mount requests, issue the QUERY REQUEST command. You can also query the activity log to determine if a given process has a pending mount request.
2. A mount request indicates that a volume is needed for the current process. However, the volume might not be available in the library. If the volume is not available, the reason might be that you either issued the MOVE MEDIA command or CHECKOUT LIBVOLUME command, or that you manually removed the volume from the library.

Preemption of client or server operations

The server can preempt server or client operations for a higher priority operation when a mount point is in use and no others are available, or access to a specific volume is required. You can use the `QUERY MOUNT` command to see the status of the volume for the mount point.

Mount point preemption

Some high priority operations can preempt operations for a mount point.

- Backup database
- Restore
- Retrieve
- HSM recall
- Export
- Import

The following operations cannot preempt other operations nor can they be preempted:

- Audit Volume
- Restore from a copy storage pool or an active-data pool
- Prepare a recovery plan
- Store data using a remote data mover

The following operations can be preempted and are listed in order of priority. The server selects the lowest priority operation to preempt, for example reclamation.

1. Move data
2. Migration from disk to sequential media
3. Backup, archive, or HSM migration
4. Migration from sequential media to sequential media
5. Reclamation

You can disable preemption by specifying `NOPREEMPT` in the server options file. When this option is specified, the `BACKUP DB` command is the only operation that can preempt other operations.

Volume access preemption

A high priority operation that requires access to a specific volume currently in use by a low priority operation can automatically preempt the operation.

For example, if a restore request requires access to a volume in use by a reclamation process and a drive is available, the reclamation process is canceled and message `ANR0494I` or `ANR1441I` is issued.

The following high priority operations can preempt operations for access to a specific volume:

- Restore
- Retrieve
- HSM recall

The following operations cannot preempt other operations nor can they be preempted:

- Audit Volume

- Restore from a copy storage pool or an active-data pool
- Prepare a recovery plan
- Store data using a remote data mover

The following operations can be preempted, and are listed in order of priority. The server preempts the lowest priority operation, for example reclamation.

1. Move data
2. Migration from disk to sequential media
3. Backup, archive, or HSM migration
4. Migration from sequential media
5. Reclamation

You can disable preemption by specifying `NOPREEMPT` in the server options file. When this option is specified, no operation can preempt another operation for access to a volume.

Setting the server name

At installation, the server name is set to `SERVER1` or on Windows, the machine name. After installation, you can use the `SET SERVERNAME` command to change the server name.

You can use the `QUERY STATUS` command to see the name of the server.

Task	Required Privilege Class
Specify the server name	System

To specify the server name as `WELLS_DESIGN_DEPT.`, for example, enter the following:

```
set servername wells_design_dept.
```

You must set unique names on servers that communicate with each other. See “Setting up communications among servers” on page 676 for details. On a network where clients connect to multiple servers, it is recommended that all of the servers have unique names.

Attention:

- 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.

Changing the server name using the `SET SERVERNAME` command can have other implications varying by platform. Some examples to be aware of are:

- Passwords might be invalidated. For example, Windows clients use the server name to identify which passwords belong to which servers. Changing the server name after Windows backup-archive clients are connected forces clients to re-enter the passwords.
- Device information might be affected.
- Registry information on Windows platforms might change.

Changing the host name for a Tivoli Storage Manager server

The Tivoli Storage Manager server requires that a few additional steps be taken to ensure that the database continues to function properly. If you do not complete the additional steps, you will receive errors when starting the server.

To change the host name and reconfigure the Tivoli Storage Manager servers on the system, complete the following steps:

1. Stop any Tivoli Storage Manager servers that are running on the system.
2. Change the host name on the system using normal operating system tools and procedures.
3. From the root user ID on the system, issue the following command:

```
db2set -g DB2SYSTEM=newhostname
```

The `db2set` command can be found in the *instance directory*/sql/lib/adm directory. The *newhostname* is the host name that the server is being changed to.

4. To verify that the DB2SYSTEM value has been changed, issue the following command:

```
db2set -all
```

This command displays all the configuration settings that are used by the database. Look for the DB2SYSTEM setting, which should now have the *newhostname* value.

5. Edit the `db2nodes.cfg` file, which is located in the *instance directory*/sql/lib directory. The file has an existing entry with the previous host name that looks like the following entry:

```
0 tsmmon TSMON 0
```

Update the entry to use the new host name, that looks like the following entry:

```
0 tsmnew newhostname 0
```

6. Save and close the changed file.
7. After completing these steps, the server should now start and run successfully.

Adding or updating server options

You may want to add or update server options to the server options file.

Task	Required Privilege Class
Add or update a server option	System

You can add or update server options by editing the `dsmserv.opt` file, using the `SETOPT` command.

For information about editing the server options file, refer to *Administrator's Reference*.

Adding or updating a server option without restarting the server

A system administrator can add or update a limited number of server options without stopping and restarting the server. The added or updated server option is appended to the end of the server options file.

You can update existing server options by issuing the SETOPT command. For example, to update the existing server option value for MAXSESSIONS to 20, you would enter:

```
setopt maxsessions 20
```

These server options can be added or updated:

- COMMTIMEOUT
- EXPINTERVAL
- EXPQUIET
- IDLETIMEOUT
- MAXSESSIONS
- RESTOREINTERVAL
- THROUGHPUTDATATHRESHOLD
- THROUGHPUTTIMETHRESHOLD

Note: SETOPT commands in a macro cannot be rolled back.

Getting help on commands and error messages

Any administrator can issue the HELP command to display information about administrative commands and messages from the server and the administrative command-line client.

You can issue the HELP command with no operands to display a menu of help selections. You also can issue the HELP command with operands that specify help menu numbers, commands, or message numbers.

To display the help menu, enter:

```
help
```

To display help information on the REMOVE commands, enter:

```
help remove
```

To display help information on a specific message, such as ANR0992I for example, enter:

```
help 0992
```

Additional information is also available in the online documentation.

Chapter 20. Automating server operations

You can schedule administrative commands to tune server operations and to start functions that require significant server or system resources during times of low usage. Automating these operations allows the administrator to ensure that server resources are available when needed by clients.

An administrator can automate the process of issuing a sequence of commands by storing the commands in a server script. From the command line, the administrator can immediately process the script or schedule the script for processing.

Tivoli Storage Manager includes a central scheduling component that allows the automatic processing of administrative commands during a specific time period when the schedule is activated. Schedules that are started by the scheduler can run in parallel. You can process scheduled commands sequentially by using scripts that contain a sequence of commands with `WAIT=YES`. You can also use a scheduler external to invoke the administrative client to start one or more administrative commands.

Each scheduled administrative command is called an *event*. The server tracks and records each scheduled event in the database. You can delete event records as needed to recover database space.

Scripts can be scheduled. For example:

1. Define a schedule named `EXPPROC` that invokes expiration processing every night at 2:00 a.m. For example:

```
define schedule expproc type=administrative -  
  cmd='expire inventory' active=yes starttime=02:00
```

This schedule calls for a schedule window that:

- Begins on the date the schedule is defined (the default) at 2:00 a.m.
 - Lasts for 1 hour (the default)
 - Is repeated every day
 - Takes effect immediately
 - Stays in effect indefinitely (the default)
2. Because the `EXPPROC` schedule is to run daily, you can verify that the automation is working as it should on the day after you define the schedule. For example:

```
query event expproc type=administrative begindate=today-1
```

If the schedule ran successfully, the status is *Completed*.

See the following topics for more information:

Concepts:
“Automating a basic administrative command schedule” on page 572
“Tailoring schedules” on page 573
“Copying schedules” on page 576
“Deleting schedules” on page 576

Concepts:
“Managing scheduled event records” on page 576
“IBM Tivoli Storage Manager server scripts” on page 578
“Using macros” on page 587

Automating a basic administrative command schedule

You can set up a basic administrative command schedule using Tivoli Storage Manager defaults.

Note:

1. Scheduled administrative command output is directed to the activity log. This output cannot be redirected. For information about the length of time activity log information is retained in the database, see “Using the IBM Tivoli Storage Manager activity log” on page 623.
2. You cannot schedule MACRO or QUERY ACTLOG commands.

To later update or tailor your schedules, see “Tailoring schedules” on page 573.

Task	Required Privilege Class
Define, update, copy, or delete administrative schedules	System
Display information about scheduled operations	Any administrator

Defining the schedule

You can use the DEFINE SCHEDULE command to create a new schedule for processing an administrative command.

Include the following parameters when defining the schedule:

- Specify the administrative command to be issued (CMD=).
- Specify whether the schedule is activated (ACTIVE=).

For example, if you issued this command:

```
define schedule backup_archivepool type=administrative
cmd='backup stgpool archivepool recoverypool' active=yes
```

It would result in the following:

- The schedule created is *BACKUP_ARCHIVEPOOL*.
- The schedule is to process the administrative command:

```
backup stgpool archivepool recoverypool
```

This command specifies that primary storage pool ARCHIVEPOOL is backed up to the copy storage pool RECOVERYPOOL.
- The schedule is currently active.
- Administrative command output is redirected to the activity log.
- The following defaults are in effect:
 - The start date and time defaults to the current date and time.
 - The length of the startup window is 1 hour.

- The priority for the schedule is 5. If schedules conflict, the schedule with the highest priority (lowest number) is run first.
- The schedule never expires.

To change the defaults, see “Tailoring schedules.”

Verifying the schedule

You can verify the details of what you have scheduled by using the QUERY SCHEDULE command. When you use the QUERY SCHEDULE command, you must specify the TYPE=ADMINISTRATIVE parameter to view an administrative command schedule.

The following figure shows an example of a report that is displayed after you enter:

```
query schedule backup_archivepool type=administrative
```

* Schedule Name	Start Date/Time	Duration	Period	Day
BACKUP_ARCHIVE- POOL	09/04/2002 14:08:11	1 H	1 D	Any

Note: The asterisk (*) in the first column specifies whether the corresponding schedule has expired. If there is an asterisk in this column, the schedule has expired.

You can check when the schedule is projected to run and whether it ran successfully by using the QUERY EVENT command. For information about querying events, see “Querying events” on page 577.

Tailoring schedules

To control more precisely when and how your schedules run, you can specify values for schedule parameters instead of accepting the defaults when you define or update schedules.

Schedule name

All schedules must have a unique name, which can be up to 30 characters.

Schedule style

You can specify either classic or enhanced scheduling. With classic scheduling, you can define the interval between the startup windows for a schedule. With enhanced scheduling, you can choose the days of the week, days of the month, weeks of the month, and months the startup window can begin on.

Initial start date, initial start time, and start day

You can specify a past date, the current date, or a future date for the initial start date for a schedule with the STARTDATE parameter.

You can specify a start time, such as 6 p.m. with the STARTTIME parameter.

For classic scheduling, you can use the DAYOFWEEK parameter to specify that the startup window begins on a certain day of the week, over the weekend, during the week, or on any day. If the start date and time

specified fall on a day that does not correspond to your value for the day of the week, the start date and time are shifted forward in 24-hour increments until the day of the week is satisfied. If you select a value for the day of the week other than ANY, schedules may not process when you expect. This depends on the values for PERIOD and PERUNITS. Use the QUERY EVENT command to project when schedules will process to ensure that you achieve the desired result.

For enhanced scheduling, you can use the DAYOFWEEK parameter to specify that the startup window begin on one or more days of the week, over the weekend, during the week, or on any day. MONTH, DAYOFMONTH, and WEEKOFMONTH can also determine a start date. When used with the DAYOFMONTH parameter, DAYOFWEEK must have a value of ANY. If DAYOFWEEK=WEEKDAY or DAYOFWEEK=WEEKEND, you must specify a value of either FIRST or LAST for the parameter WEEKOFMONTH.

Duration of a startup window

You can specify the duration of a startup window, such as 12 hours, with the DURATION and DURUNITS parameters. The server must start the scheduled service within the specified duration, but does not necessarily complete it within that period of time. If the schedule needs to be retried for any reason, the retry attempt must begin before the startup window elapses or the operation does not restart.

If the schedule does not start during the startup window, the server records this as a *missed event* in the database. You can get an exception report from the server to identify schedules that did not run. For more information, see “Querying events” on page 577.

How often to run the scheduled service

With classic scheduling, you can set the schedule frequency based on a period of hours, days, weeks, months, or years with the PERIOD and PERUNITS parameters. To have weekly backups, for example, set the period to one week with PERIOD=1 and PERUNITS=WEEKS.

With enhanced scheduling specified, you can set your schedule to run on certain months, days of the month, days of the week, and weeks of the month with the MONTH, DAYOFMONTH, DAYOFWEEK, and WEEKOFMONTH parameters, respectively. For example, if you want your schedule to run on the first and last day of January and June, specify the months of January and June and choose the first and last days of the month with MONTH=JANUARY,JUNE and DAYOFMONTH=1,-1. If you want your schedule to run during the last week of November, for example, choose the last week of the month and November with MONTH=NOVEMBER and WEEKOFMONTH=LAST.

Expiration date

You can specify an expiration date for a schedule with the EXPIRATION parameter if the services it initiates are required for only a specific period of time. If you set an expiration date, the schedule is not used after that date, but it still exists. You must delete the schedule to remove it from the database.

Priority

You can assign a priority to schedules with the PRIORITY parameter. For example, if you define two schedules and they have the same startup

window or windows overlap, the server runs the schedule with the highest priority first. A schedule with a priority of 1 is started before a schedule with a priority of 3.

If two schedules try to use the same resources, the schedule that first initiated the process will be the one to continue processing. The second schedule will start but will not successfully complete. Be sure to check the activity log for details.

Administrative schedule name

If you are defining or updating an administrative command schedule, you must specify the schedule name.

Type of schedule

If you are updating an administrative command schedule, you must specify `TYPE=ADMINISTRATIVE` on the `UPDATE` command. If you are defining a new administrative command schedule, this parameter is assumed if the `CMD` parameter is specified.

Command

When you define an administrative command schedule, you must specify the complete command that is processed with the schedule with the `CMD` parameter. These commands are used to tune server operations or to start functions that require significant server or system resources. The functions include:

- Migration
- Reclamation
- Export and import
- Database backup

Whether or not the schedule is active

Administrative command schedules can be active or inactive when they are defined or updated. Active schedules are processed when the specified command window occurs. Inactive schedules are not processed until they are made active by an `UPDATE SCHEDULE` command with the `ACTIVE` parameter set to `YES`.

Using classic and enhanced command schedules

Depending on what type of event you want you schedule, and how often, you can schedule commands to run using classic or enhanced scheduling.

Classic Scheduling

To schedule the backup of the `ARCHIVEPOOL` primary storage pool periodically, use classic scheduling. Enter the following command:

```
define schedule backup_archivepool type=administrative
cmd='backup stgpool archivepool recoverypool'
active=yes starttime=20:00 period=2
```

This command specifies that, starting today, the `ARCHIVEPOOL` primary storage pool is to be backed up to the `RECOVERYPOOL` copy storage pool every two days at 8 p.m.

To update the `BACKUP_ARCHIVEPOOL` schedule, enter:

```
update schedule backup_archivepool type=administrative
starttime=20:00 period=3
```

Starting today, the `BACKUP_ARCHIVEPOOL` schedule begins the backup every three days at 10 p.m.

Enhanced Scheduling

To schedule the backup of the CENTRALPOOL primary storage pool on specific days of the month, use enhanced scheduling. Enter the following command:

```
define schedule backup_centralpool type=administrative
cmd='backup stgpool centralpool auxiliarypool'
active=yes starttime=22:00 schedstyle=enhanced dayofmonth=10,-1
```

This command specifies that the CENTRALPOOL primary storage pool is to be backed up to the AUXILARYPOOL copy storage pool on the tenth and last day of each month at 10 p.m.

To update the BACKUP_CENTRALPOOL schedule, enter:

```
update schedule backup_centralpool type=administrative
starttime=19:00 dayofmonth=-2
```

Starting today, the BACKUP_CENTRALPOOL schedule will begin the backup on the second-to-last day of the month at 7 p.m.

Copying schedules

You can create a new schedule by copying an existing administrative schedule. When you copy a schedule, Tivoli Storage Manager copies the following information:

- A description of the schedule
- All parameter values from the original schedule

You can then update the new schedule to meet your needs.

To copy the BACKUP_ARCHIVEPOOL administrative schedule and name the new schedule BCKSCHED, enter:

```
copy schedule backup_archivepool bcksched type=administrative
```

Deleting schedules

To delete the administrative schedule ENGBKUP, enter:

```
delete schedule engbkup type=administrative
```

Managing scheduled event records

Each scheduled administrative command operation is called an *event*.

Task	Required Privilege Class
Display information about events	Any administrator
Set the retention period for event records	System
Delete event records	System or unrestricted policy

All scheduled events, including their status, are tracked by the server. An *event record* is created in the server database whenever processing of a scheduled command is created or missed.

Querying events

To help manage schedules for administrative commands, you can request information about scheduled and completed events. You can request general or exception reporting queries.

- To get information about past and projected scheduled processes, use a general query. If the time range you specify includes the future, the query output shows which events should occur in the future based on current schedules.
- To get information about scheduled processes that did not complete successfully, use exception reporting.

To minimize the processing time when querying events, minimize the time range.

To query an event for an administrative command schedule, you must specify the `TYPE=ADMINISTRATIVE` parameter. Figure 75 shows an example of the results of the following command:

```
query event * type=administrative
```

Scheduled Start	Actual Start	Schedule Name	Status
09/04/2002 14:08:11	09/04/2002 14:08:14	BACKUP_ARCHI- VEPOOL	Completed

Figure 75. Query results for an administrative schedule

Removing event records from the database

You can specify how long event records stay in the database before the server deletes them. You can also manually remove event records from the database.

If you issue a query for events, past events may display even if the event records have been deleted. The events displayed with a status of *Uncertain* indicate that complete information is not available because the event records have been deleted. To determine if event records have been deleted, check the message that is issued after the `DELETE EVENT` command is processed.

Setting the event record retention period

You can specify the retention period for event records in the database. After the retention period passes, the server automatically removes the event records from the database. At installation, the retention period is set to 10 days.

Event records are automatically removed from the database after both of the following conditions are met:

- The specified retention period has passed
- The startup window for the event has elapsed

You can change the retention period from the default of 10 days by using the `SET EVENTRETENTION` command.

Deleting event records

Because event records are deleted automatically, you do not have to manually delete them from the database. However, you may want to manually delete event records to increase available database space.

Use the DELETE EVENT command manually remove event records. For example, to delete all event records written prior to 11:59 p.m. on June 30, 2002, enter:

```
delete event type=administrative 06/30/2002 23:59
```

IBM Tivoli Storage Manager server scripts

Tivoli Storage Manager provides for automation of common administrative tasks with server scripts that are stored in the database.

Tivoli Storage Manager provides sample scripts in:

- scripts.smp

The sample scripts have an example order of execution for scheduling administrative commands. If one of the specified commands in the script does not process successfully, the remaining commands are not processed. For more information, see “Using SELECT commands in IBM Tivoli Storage Manager scripts” on page 621.

The administrator can run the script from the Administration Center, or schedule the script for processing using the administrative command scheduler on the server.

Tivoli Storage Manager scripts can include the following:

- Command parameter substitution.
- SQL SELECT statements that you specify when the script is processed.
- Command execution control, such as PARALLEL and SERIAL processing options.
- Conditional logic flow statements. These logic flow statements include:
 - The IF clause; this clause determines how processing should proceed based on the current return code value.
 - The EXIT statement; this statement ends script processing.
 - The GOTO and LABEL statement; this statement directs logic flow to continue processing with the line that starts with the label specified.
 - Comment lines.

Defining a server script

You can define a server script line-by-line, create a file that contains the command lines, or copy an existing script.

Task	Required Privilege Class
Define a server script	System, policy, storage, and operator

You can define a script with the DEFINE SCRIPT command. You can initially define the first line of the script with this command. For example:

```
define script qaixc "select node_name from nodes where platform='aix'"  
desc='Display AIX clients'
```

This example defines the script as QAIXC. When you run the script, all AIX clients are displayed.

To define additional lines, use the UPDATE SCRIPT command. For example, you want to add a QUERY SESSION command, enter:

```
update script qaixc "query session *"
```

You can also easily define and update scripts using the Administration Center where you can also use local workstation cut and paste functions.

Note: The Administration Center only supports ASCII characters for input. If you need to enter characters that are not ASCII, do not use the Administration Center. Issue the DEFINE SCRIPT and UPDATE SCRIPT commands from the server console.

You can specify a WAIT parameter with the DEFINE CLIENTACTION command. This allows the client action to complete before processing the next step in a command script or macro. To determine where a problem is within a command in a script, use the ISSUE MESSAGE command.

Refer to *Administrator's Reference* for information on the DEFINE CLIENTACTION and ISSUE MESSAGE commands.

For additional information about updating server scripts, or updating a command line, see "Updating a script" on page 584.

Defining a server script using contents of another file

You can define a script whose command lines are read in from another file that contains statements for the script to be defined.

For example, to define a script whose command lines are read in from the file BKUP12.MAC, issue:

```
define script admin1 file=bkup12.mac
```

The script is defined as ADMIN1, and the contents of the script have been read in from the file BKUP12.MAC.

Note: The file must reside on the server, and be read by the server.

Creating a maintenance script

You can create a predefined or a custom maintenance script. Either kind can help you protect your data by running maintenance commands on a schedule.

Custom maintenance script

The custom maintenance script is either created using the maintenance script editor or is converted using a predefined script. The script editor is designed for experienced Tivoli Storage Manager users who require more flexibility when constructing maintenance scripts.

Predefined maintenance script

The predefined maintenance script is created using a wizard.

You must schedule the maintenance script to run. The script typically includes commands to back up, copy, and delete data. You can automate your server maintenance by creating a maintenance script, and running it when your server is not in heavy use.

Creating a custom maintenance script:

A custom maintenance script can be created using the maintenance script editor or by converting a predefined maintenance script.

When you click **Server Maintenance** in the navigation tree, a list of servers is displayed in the **Maintenance Script** table with either None, Custom, or Predefined noted in the **Maintenance Script** column.

If you want to convert a predefined maintenance script into a custom maintenance script, select a server that has a predefined script and click **Select Action** → **Convert to Custom Maintenance Script**. Your predefined maintenance script converts into a custom script and opens in the maintenance script editor. You cannot convert a custom script into a predefined maintenance script.

Perform the following steps to create a custom maintenance script using the maintenance script editor:

1. Select a server.
2. Click **Select Action** → **Create Custom Maintenance Script**.
3. Click **Select an Action** and construct your maintenance script by adding a command to the script. The following actions are available:
 - Back Up Server Database
 - Back Up Storage Pool
 - Copy Active Data to Active-data Pool
 - Create Recovery Plan File
 - Insert Comment
 - Delete Volume History
 - Delete Expired Data
 - Migrate Stored Data
 - Move Disaster Recovery Media
 - Run Script Commands in Parallel
 - Run Script Commands Serially
 - Reclaim Primary Storage Pool
 - Reclaim Copy Storage Pool

You can change the order of commands by selecting a command and using the up or down arrow to position it.

4. When you are satisfied with your script, define the schedule and click **OK**.

To edit your custom script after it is created and saved, click **Server Maintenance** in the navigation tree, select the server with the custom script and click **Select Action** → **Modify Maintenance Script**. Your custom maintenance script opens in the script editor where you can add, remove, or change the order of the commands.

Creating a predefined maintenance script:

You can produce a predefined maintenance script using the maintenance script wizard.

When you click **Server Maintenance** in the navigation tree, a list of servers is displayed in the **Maintenance Script** table with either None, Custom, or Predefined noted in the **Maintenance Script** column.

Perform the following steps to create a maintenance script using the maintenance script wizard:

1. Select a server that requires a maintenance script to be defined (None is specified in the **Maintenance Script** column).
2. Click **Select Action** → **Create Maintenance Script**.
3. Follow the steps in the wizard.

After completing the steps in the wizard, you can convert your predefined maintenance script into a custom maintenance script. If you choose to convert your script into a custom script, select the server and click **Select Action** → **Convert to Custom Maintenance Script**. Your predefined maintenance script is converted and opened in the maintenance script editor where you can modify the schedule and the maintenance actions.

Running commands in parallel or serially

You have the options of running commands serially, in parallel, or serially and in parallel. 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.

Refer to the *Administrator's Reference* for more information on the PARALLEL and SERIAL script commands.

The following example illustrates how the parallel command is used to backup, migrate and reclaim storage pools.

```
/*run multiple commands in parallel and wait for
them to complete before proceeding*/
PARALLEL
/*back up four storage pools simultaneously*/
BACKUP STGPOOL PRIMPOOL1 COPYPOOL1 WAIT=YES
BACKUP STGPOOL PRIMPOOL2 COPYPOOL2 WAIT=YES
BACKUP STGPOOL PRIMPOOL3 COPYPOOL3 WAIT=YES
BACKUP STGPOOL PRIMPOOL4 COPYPOOL4 WAIT=YES
/*wait for all previous commands to finish*/
SERIAL
/*after the backups complete, migrate stgpools
simultaneously*/
PARALLEL
MIGRATE STGPOOL PRIMPOOL1 DURATION=90 WAIT=YES
MIGRATE STGPOOL PRIMPOOL2 DURATION=90 WAIT=YES
MIGRATE STGPOOL PRIMPOOL3 DURATION=90 WAIT=YES
MIGRATE STGPOOL PRIMPOOL4 DURATION=90 WAIT=YES
/*wait for all previous commands to finish*/
SERIAL
/*after migration completes, reclaim storage
pools simultaneously*/
PARALLEL
RECLAIM STGPOOL PRIMPOOL1 DURATION=120 WAIT=YES
```

```
RECLAIM STGPOOL PRIMPOOL2 DURATION=120 WAIT=YES
RECLAIM STGPOOL PRIMPOOL3 DURATION=120 WAIT=YES
RECLAIM STGPOOL PRIMPOOL4 DURATION=120 WAIT=YES
```

Using continuation characters for long commands

You can continue long commands across multiple command lines by specifying the continuation character (-) as the last character for a command that is continued.

The following example continues an SQL statement across multiple command lines:

```
/*-----*/
/* Sample continuation example */
SELECT-
* FROM-
NODE WHERE-
PLATFORM='win32'
```

When this command is processed, it runs the following:

```
select * from nodes where platform='win32'
```

Using substitution variables

You can include substitution variables in a script. Substitution variables are specified with a \$ character followed by a number that represents the position of the parameter when the script is processed.

The following example SQLSAMPLE script specifies substitution variables \$1 and \$2:

```
/*-----*/
/* Sample substitution example */
/* -----*/
SELECT-
$1 FROM-
NODES WHERE-
PLATFORM='$2'
```

When you run the script you must specify two values, one for \$1 and one for \$2. For example:

```
run sqlsample node_name aix
```

The command that is processed when the SQLSAMPLE script is run is:

```
select node_name from nodes where platform='aix'
```

Using logic flow statements in a script

You can use conditional logic flow statements based on return codes issued from previous command processing. These logic statements allow you to process your scripts based on the outcome of certain commands. You can use IF, EXIT, or GOTO (label) statements.

As each command is processed in a script, the return code is saved for possible evaluation before the next command is processed. The return code can be one of three severities: OK, WARNING, or ERROR. Refer to *Administrator's Reference* for a list of valid return codes and severity levels.

Specifying the IF clause:

You can use the IF clause at the beginning of a command line to determine how processing of the script should proceed based on the current return code value. In the IF clause you specify a return code symbolic value or severity.

The server initially sets the return code at the beginning of the script to RC_OK. The return code is updated by each processed command. If the current return code from the processed command is equal to any of the return codes or severities in the IF clause, the remainder of the line is processed. If the current return code is not equal to one of the listed values, the line is skipped.

The following script example backs up the BACKUPPOOL storage pool only if there are no sessions currently accessing the server. The backup proceeds only if a return code of RC_NOTFOUND is received:

```
/* Backup storage pools if clients are not accessing the server */
select * from sessions
/* There are no sessions if rc_notfound is received */
if(rc_notfound) backup stg backuppool copypool
```

The following script example backs up the BACKUPPOOL storage pool if a return code with a severity of warning is encountered:

```
/* Backup storage pools if clients are not accessing the server */
select * from sessions
/* There are no sessions if rc_notfound is received */
if(warning) backup stg backuppool copypool
```

Specifying the EXIT statement:

Use the EXIT statement to end script processing.

The following example uses the IF clause together with RC_OK to determine if clients are accessing the server. If a RC_OK return code is received, this indicates that client sessions are accessing the server. The script proceeds with the exit statement, and the backup does not start.

```
/* Back up storage pools if clients are not accessing the server */
select * from sessions
/* There are sessions if rc_ok is received */
if(rc_ok) exit
backup stg backuppool copypool
```

Specifying the GOTO statement:

The GOTO statement is used in conjunction with a label statement. The label statement is the target of the GOTO statement. The GOTO statement directs script processing to the line that contains the label statement to resume processing from that point.

The label statement always has a colon (:) after it and may be blank after the colon. The following example uses the GOTO statement to back up the storage pool only if there are no sessions currently accessing the server. In this example, the return code of RC_OK indicates that clients are accessing the server. The GOTO statement directs processing to the **done:** label which contains the EXIT statement that ends the script processing:

```

/* Back up storage pools if clients are not accessing the server */
select * from sessions
/* There are sessions if rc_ok is received */
if(rc_ok) goto done
backup stg backuppool copypool
done:exit

```

Managing server scripts

You can update, copy, rename, query, delete, and run server scripts.

Task	Required Privilege Class
Update, copy, rename, query, and delete a script	System, policy, storage, and operator
Run a script	System, policy, storage, and operator

Updating a script

You can update a script to change an existing command line or to add a new command line to a script.

Appending a new command:

To append a command line to an existing script issue the UPDATE SCRIPT command without the LINE= parameter. The appended command line is assigned a line number of five greater than the last command line number in the command line sequence. For example, if your script ends with line 010, the appended command line is assigned a line number of 015.

The following is an example of the QSTATUS script. The script has lines 001, 005, and 010 as follows:

```

001 /* This is the QSTATUS script */
005 QUERY STATUS
010 QUERY PROCESS

```

To append the QUERY SESSION command at the end of the script, issue the following:

```
update script qstatus "query session"
```

The QUERY SESSION command is assigned a command line number of 015 and the updated script is as follows:

```

001 /* This is the QSTATUS script */
005 QUERY STATUS
010 QUERY PROCESS
015 QUERY SESSION

```

Replacing an existing command:

You can change an existing command line by specifying the LINE= parameter.

Line number 010 in the QSTATUS script contains a QUERY PROCESS command. To replace the QUERY PROCESS command with the QUERY STGPOOL command, specify the LINE= parameter as follows:

```
update script qstatus "query stgpool" line=10
```

The QSTATUS script is updated to the following:

```
001 /* This is the QSTATUS script */
005 QUERY STATUS
010 QUERY STGPOOL
015 QUERY SESSION
```

Adding a new command and line number:

You can change an existing script by adding new lines.

To add the SET REGISTRATION OPEN command as the new line 007 in the QSTATUS script, issue the following:

```
update script qstatus "set registration open" line=7
```

The QSTATUS script is updated to the following:

```
001 /* This is the QSTATUS script */
005 QUERY STATUS
007 SET REGISTRATION OPEN
010 QUERY STGPOOL
015 QUERY SESSION
```

Copying a server script

You can copy an existing script to a new script with a different name.

For example, to copy the QSTATUS script to QUERY1 script, issue:

```
copy script qstatus query1
```

The QUERY1 command script now contains the same command lines as the QSTATUS command script.

Querying a server script

You can query a script to display information about the script. You can specify wildcard characters to display all scripts with names that match a particular pattern. When you query a script, you can direct the output to a file in a file system that the server can access.

The various formats you can use to query scripts are as follows:

Format	Description
Standard	Displays the script name and description. This is the default.
Detailed	Displays commands in the script and their line numbers, date of last update, and update administrator for each command line in the script.
Lines	Displays the name of the script, the line numbers of the commands, comment lines, and the commands.
File	Outputs only the commands contained in the script without all other attributes. You can use this format to direct the script to a file so that it can be loaded into another server with the DEFINE script command specifying the FILE= parameter.

To query a script in the standard format, issue the following:

```
query script *
```

The command gives results like the following:

Name	Description
QCOLS	Display columns for a specified SQL table
QSAMPLE	Sample SQL Query

For more information about querying a server script, refer to *Administrator's Reference*.

Querying a server script to create another server script:

You can create additional server scripts by querying a script and specifying the `FORMAT=FILE` and `OUTPUTFILE` parameters. You can use the resulting output as input into another script without having to create a script line by line.

The following is an example of querying the `SRTL2` script and directing the output to `newsript.script`:

```
query script srtl2 format=file outputfile=newscript.script
```

You can then edit the `newsript.script` with an editor that is available to you on your system. To create a new script using the edited output from your query, issue:

```
define script srtnew file=newscript.script
```

Renaming a server script

You can rename a script to a different name.

For example, to rename the `QUERY1` script to `QUERY5`, issue:

```
rename script query1 query5
```

The `QUERY1` script is now named `QUERY5`.

Deleting a command from a server script

You can delete an individual command line from a script. When you specify a line number, only the corresponding command line is deleted from the script.

For example, to delete the `007` command line from the `QSTATUS` script, issue:

```
delete script qstatus line=7
```

Deleting a server script

To delete an entire script, issue the `DELETE SCRIPT` command.

For example, to delete the `QSTATUS` script, issue:

```
delete script qstatus
```

Running a server script

To process a script, issue the `RUN` command. You can run a script that contains substitution variables by specifying them along with the `RUN` command.

Note: There is no Tivoli Storage Manager command that can cancel a script after it starts. To stop a script, an administrator must halt the server.

You can preview the command lines of a script without actually executing the commands by using the `PREVIEW=YES` parameter with the `RUN` command. If the script contains substitution variables, the command lines are displayed with the substituted variables. This is useful for evaluating a script before you run it.

For example, to process the QAIXC script previously defined, issue:

```
run qaixc
```

To process the following script that contains substitution variables:

```
/*-----*/  
/* Sample continuation and substitution example */  
/* -----*/  
SELECT-  
$1 FROM-  
NODES WHERE-  
PLATFORM='$2'
```

Enter:

```
run qaixc node_name aix
```

Where \$1 is node_name and \$2 is aix.

Using macros

Tivoli Storage Manager supports macros on the administrative client. A macro is a file that contains one or more administrative client commands. You can only run a macro from the administrative client in batch or interactive modes. Macros are stored as a file on the administrative client. Macros are not distributed across servers and cannot be scheduled on the server.

Macros can include the following:

- Administrative commands
For more information on administrative commands, see “Writing commands in a macro” on page 588.
- Comments
For more information on comments, see “Writing comments in a macro” on page 588.
- Continuation characters
For more information on continuation characters, see “Using continuation characters” on page 589.
- Variables
For more information on variables, see “Using substitution variables in a macro” on page 589.

The name for a macro must follow the naming conventions of the administrative client running on your operating system. For more information about file naming conventions, refer to the *Administrator's Reference*.

In macros that contain several commands, use the COMMIT and ROLLBACK commands to control command processing within the macro. For more information about using these commands, see “Command processing in a macro” on page 590.

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

Writing commands in a macro

You can add commands to a macro.

The administrative client ignores any blank lines included in your macro. However, a completely blank line terminates a command that is continued (with a continuation character).

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

```
register admin pease mypasswd -  
  contact='david pease, x1234'  
grant authority pease -  
  classes=policy,storage -  
  domains=domain1,domain2 -  
  stgpools=stgpool1,stgpool2
```

This example uses continuation characters in the macro file. For more information on continuation characters, see “Using continuation characters” on page 589.

After you create a macro file, you can update the information that it contains and use it again. You can also copy the macro file, make changes to the copy, and then run the copy. Refer to the *Administrator's Reference* for more information on how commands are entered and the general rules for entering administrative commands.

Writing comments in a macro

You can add comments to your macro file.

To write a comment:

- Write a slash and an asterisk (/*) to indicate the beginning of the comment.
- Write the comment.
- Write an asterisk and a slash (*/) to indicate the end of the comment.

You can put a comment on a line by itself, or you can put it on a line that contains a command or part of a command.

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

```
/* auth.mac-register new nodes */
```

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

```
domain=domain1          /*assign node to domain1 */
```

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

Using continuation characters

You can use continuation characters in a macro file. Continuation characters are useful when you want to execute a command that is longer than your screen or window width.

Attention: Without continuation characters, you can enter up to 256 characters. With continuation characters, you can enter up to 1500 characters. In the MACRO command, these maximums are *after* any substitution variables have been applied (see “Using substitution variables in a macro”).

To use a continuation character, enter a dash or a back slash at the end of the line that you want to continue. With continuation characters, you can do the following:

- Continue a command. For example:

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

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

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

- Continue a string of values enclosed in quotation marks by entering the first part of the string enclosed in quotation marks, followed by a dash or a back slash at the end of the line. Then, enter the remainder of the string on the next line enclosed in the *same* type of quotation marks. For example:

```
contact="david pease, bldg. 100, room 2b, san jose,"-  
"ext. 1234, alternate contact-norm pass,ext 2345"
```

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

Using substitution variables in a macro

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

A substitution variable consists of a percent sign (%), followed by a number that indicates the number of the substitution variable. When you run the file with the MACRO command, you must specify values for the variables.

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

```
/* register new nodes */  
register node %1 %2 - /* userid password */  
contact=%3 - /* 'name, phone number' */  
domain=%4 /* policy domain */
```

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

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

```
macro auth.mac david davidpw "david pease, x1234" domain1
```

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

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

Running a macro

Use the `MACRO` command when you want to run a macro. You can enter the `MACRO` command in batch or interactive mode.

If the macro does not contain substitution variables (such as the `REG.MAC` macro described in the “Writing commands in a macro” on page 588), run the macro by entering the `MACRO` command with the name of the macro file. For example:

```
macro reg.mac
```

If the macro contains substitution variables (such as the `AUTH.MAC` macro described in “Using substitution variables in a macro” on page 589), include the values that you want to supply after the name of the macro. Each value is delimited by a space. For example:

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

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

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

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

Command processing in a macro

When you issue a `MACRO` command, the server processes all commands in the macro file in order, including commands contained in any nested macros. The server commits all commands in a macro after successfully completing processing for the highest-level macro.

If an error occurs in any command in the macro or in any nested macro, the server terminates processing and rolls back any changes caused by all previous commands.

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

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

Note: Commands that start background processes cannot be rolled back. For a list of commands that can generate background processes, see “Managing server processes” on page 564.

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

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

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

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

Chapter 21. Managing the database and recovery log

The IBM Tivoli Storage Manager database contains information that is needed for server operations and information about client data that has been backed up, archived, and space-managed. The recovery log contains information about database updates that have not yet been committed.

The following sections provide detailed concept and task information about the database and recovery log.

Concepts:
“Database and recovery log overview”

Tasks:
“Estimating database space requirements” on page 599
“Estimating recovery log space requirements” on page 601
“Monitoring the database and recovery log” on page 603
“Increasing the size of the database” on page 604
“Reducing the size of the database” on page 604
“Increasing the size of the active log” on page 605
“Backing up the database” on page 605
“Restoring the database” on page 607
“Moving the database and recovery log on a server” on page 607
“Adding optional logs after server initialization” on page 610
“Transaction processing” on page 610

Database and recovery log overview

The Tivoli Storage Manager administrative interfaces work with the database and recovery log. The skills of a database administrator are not required to manage them.

Experienced DB2 administrators can issue advanced SQL queries and use DB2 tools to monitor the database. However, do not use DB2 tools to change DB2 configuration settings from those that are preset by Tivoli Storage Manager, or alter the DB2 environment for Tivoli Storage Manager in other ways. The Tivoli Storage Manager server has been built and tested extensively using the data definition language (DDL) and database configuration that Tivoli Storage Manager deploys.

Attention: Making changes to the DDL or database configuration without using Tivoli Storage Manager interfaces can adversely affect performance, damage or destroy the server database, or cause data to become permanently lost. Do not use database tools or interfaces other than those provided or documented by Tivoli Storage Manager to change configuration settings from those that are set by Tivoli Storage Manager at installation. Do not alter the DB2 environment in other ways. If you use database tools or interfaces other than those provided or documented by Tivoli Storage Manager, you must treat the server database as read-only. Do not use other interfaces to make changes to the server database.

Database

The database does not store client data; it points to the locations of the client files in the storage pools.

The database includes information about:

- Client nodes and administrators
- Policies and schedules
- Server settings
- Locations of client files on server storage
- Server operations (for example, activity logs and event records)

The database manager manages database volumes, and there is no need to format them.

Attention: If the database is unusable, the entire Tivoli Storage Manager server is unavailable. If a database is lost and cannot be recovered, it might be difficult or impossible to recover data managed by that server. Therefore, it is critically important to back up the database. However, even without the database, fragments of data or complete files might easily be read from storage pool volumes that are not encrypted. Even if data is not completely recovered, security can be compromised. For this reason, sensitive data should always be encrypted by the Tivoli Storage Manager client or the storage device, unless the storage media is physically secured. See Part 5, “Protecting the server,” on page 753 for steps that you can take to protect your database.

The database can be distributed across up to 128 directories. The maximum supported size of the database is 2 TB. It is important that the database is placed on fast, reliable disks that are configured for random access I/O. Locating each directory on a different file system provides the best performance because the data is striped across the directories. Enable read cache for the database, and enable write cache if the disk subsystem supports it.

The database cannot be mirrored through Tivoli Storage Manager; but it can be mirrored using hardware mirroring, such as is provided by RAID 5.

Some advantages of the database manager are:

Automatic backups

When the server is started for the first time, a full backup begins automatically. When the server is next started, the database manager automatically backs up the database according to the following values set by Tivoli Storage Manager:

- The active log space consumed since the last backup, which triggers a full database backup

- The active log utilization ratio, which triggers an incremental database backup

Automatic statistics collection

Automatic statistics collection helps to improve database performance by collecting up-to-date table statistics. The database manager determines which statistics need to be updated.

Automatic database reorganization

Based on activity, the database manager program analyzes selected database tables to determine when reorganization is needed for the tables. The database manager then runs a reorganization while server operations continue. If the database reorganization puts too heavy a workload on the CPU, processing is reduced or halts.

SQL queries

The database makes more sophisticated SQL queries on the data possible.. To take advantage of these capabilities, SQL skills might be required to develop new tools and SQL statements.

Database audits

Database audits are run automatically, as needed, to ensure consistency. As data is added to the server database, the database manager checks data constraints and data types. Online integrity checks can prevent problems for which offline audits had been needed in earlier releases.

Database buffer size

The database manager automatically adjusts the values for several memory configuration parameters based on the requirements of the workload of the system.

Recovery log

The recovery log helps to ensure that a failure (such as a system power outage or application error) does not leave the database in an inconsistent state. The recovery log is essential if you need to restore the database.

If a failure occurs, the changes that were made but not committed are rolled back. Then all committed transactions, which might not have been physically written to disk, are redone.

The recovery log consists of these logs:

- Active log
- Log mirror (optional)
- Archive log
- Archive failover log (optional)

| During the installation process, you specify the directory location, the size of the
| active log, and the location of the archive logs. You can also specify the directory
| location of a log mirror if you want the additional protection of mirroring the
| active log. The amount of space for the archive logs is not limited, which improves
| the capacity of the server for concurrent operations compared to previous versions.

| The space that you designate for the recovery log is managed automatically by the
| database manager program. Space is used as needed, up to the capacity of the
| defined log directories. You do not need to create and format volumes for the
| recovery log.

Ensuring that the recovery log has enough space is as important for a V6.2 server as for earlier versions of the server. Monitor the space usage for the recovery log to prevent problems.

Attention: To protect your data, locate the database directories and all the log directories on separate physical disks.

Recovery log mode

The Tivoli Storage Manager server always runs in a mode that is equivalent to the roll-forward mode.

Changes to the database are recorded in the recovery log to maintain a consistent database image. Active and archive log files, which are included in database backups, let you restore the server to the latest time possible. You can also restore the database to a specific point in time.

To help ensure that the required log information is available for restoring the database, you can specify that the active log is mirrored to another file system location. For the best availability, locate the active log mirror on a different physical device.

Active log

The active log files record transactions that are in progress on the server.

The active log stores all the transactions that have not yet been committed. The active log always contains the most recent log records. If a failure occurs, the changes that were made but not committed are rolled back, and all committed transactions, which might not have been physically written to disk, are reapplied and committed again.

The location and size of the active log are set during initial configuration of a new or upgraded server (or by specifying the `ACTIVELOGDIR` and the `ACTIVELOGSIZE` parameters of the `DSMSERV FORMAT` or `DSMSERV LOADFORMAT` utility). Both the location and size can be changed later. To change the size of the active log, see “Increasing the size of the active log” on page 605. To change the location of the active log directory, see “Moving only the active log” on page 609.

For information about the space required for the active log directory, see “Active log space” on page 601.

Active log mirror

The active log mirror is a copy of the active log that can be used if the active log files cannot be read. All changes made to the active log are also written to the log mirror. There can be only one active log mirror.

Mirroring the active log can protect the database when a hardware failure occurs on the device where the active log is stored. Mirroring the active log provides another level of protection in addition to placing the active log on hardware that has high-availability features. Creating a log mirror is optional but recommended. Place the active log directory and the log mirror directory on different physical devices. If you increase the size of the active log, the log mirror size is increased automatically.

Mirroring the log can affect performance, because of the doubled I/O activity that is required to maintain the mirror. The additional space that the log mirror requires is another factor to consider.

You can create the log mirror during initial configuration of a new or upgraded server. If you use the DSMSEV LOADFORMAT utility instead of the wizard to configure the server, you specify the MIRRORLOGDIR parameter. If the log mirror directory is not created at that time, you can create it later by specifying the MIRRORLOGDIR option in the server options file, dsmserv.opt.

Archive log

The archive log contains copies of closed log files that had been in the active log. The archive log is not needed for normal processing, but it is typically needed for recovery of the database.

To provide roll-forward recovery of the database to the current point in time, all logs since the last database backup must be available for the restore operation. The archive log files are included in database backups and are used for roll-forward recovery of the database to the current point-in-time. All logs since the last full database backup must be available to the restore function. These log files are stored in the archive log. The pruning of the archive log files is based on full database backups. The archive log files that are included in a database backup are automatically pruned after a full database backup cycle has been completed.

The archive log is not needed during normal processing, but it is typically needed for recovery of the database. Archived log files are saved until they are included in a full database backup. The amount of space for the archive log is not limited.

Archive log files are automatically deleted as part of the full backup processes. Archive log files must not be deleted manually. Monitor both the active and archive logs. If the active log is close to filling, check the archive log. If the archive log is full or close to full, run one or more full database backups.

If the file systems or drives where the archive log directory and the archive failover log directory are located become full, the archived logs are stored in the active log directory. Those archived logs are returned to the archive log directory when the space problem is resolved, or when a full database backup is run.

You initially set the location of the archive log directory during initial configuration of a new or upgraded server (or by specifying the ARCHLOGDIR parameter of the DSMSEV FORMAT or DSMSEV LOADFORMAT utility). The location of the log can be changed later. To change the location of the archive log, see “Moving only the archive log” on page 609.

For information about the space required for the archive log, see “Archive log space” on page 602.

Archive failover log

The archive failover log, also called a secondary archive log, is the directory that the server uses to store archive log files when the archive log directory is full. Its use is optional but highly recommended.

Specifying an archive failover log directory can prevent problems that occur if the archive log runs out of space. Place the archive log directory and the archive failover log directory on different physical drives.

You can specify the location of the failover log directory during initial configuration of a new or upgraded server (or with the ARCHFAILOVERLOGDIR parameter of the DSMSEV FORMAT or DSMSEV LOADFORMAT utility). If it is not created through the utilities, it can be created later by specifying the ARCHFAILOVERLOGDIR in the server options file, dsmserv.opt. See “Adding optional logs after server initialization” on page 610 for details.

For information about the space required for the log, see “Archive failover log space” on page 602.

The role of the recovery log

When the logs that make up the recovery log are set up carefully, they work together to ensure that data is not lost.

The active log files contain information about in-progress transactions. This information is needed to restart the server and database after a disaster. Transactions are stored in the log files of the active log, and a transaction can span multiple log files.

When all transactions that are part of an active log file complete, that log file is copied from the active log to the archive log. Transactions continue to be written to the active log files while the completed active log files are copied to the archive log. If a transaction spans all the active log files, and the files are filled before the transaction is committed, the Tivoli Storage Manager server halts.

When an active log file is full, and there are no active transactions referring to it, the file is copied to the archive log directory. An active log file cannot be deleted until all transactions in the log file are either committed or discontinued.

If the archive log is full and there is no failover archive log, the log files remain in the active log. If the active log then becomes full and there are in-progress transactions, the Tivoli Storage Manager server halts. If there is an archive failover log, it is used only if the archive log fills. The database manager can move active log files to the failover archive log. It is important to monitor the archive log directory to ensure that there is space in the active log.

When the database is backed up, the database manager deletes the archive log files that are no longer needed for future database backups or restores.

The archive log is included in database backups and is used for roll-forward recovery of the database. The archive log files that are included in a database backup are automatically pruned after a full database backup cycle has completed. Therefore, ensure that the archive log has enough space to store the log files for the database backups.

Disk space requirements for the server database and recovery log

The drives or file systems on which you locate the database and log directories are important to the proper operation of your IBM Tivoli Storage Manager server. Placing each database and recovery log directory on a separate disk provides the best performance and the best disaster protection.

For the optimal database performance, choose the fastest and most reliable disks that are configured for random access I/O, such as Redundant Array of Independent Disks (RAID) hardware. The internal disks included by default in most servers and consumer grade Parallel Advanced Technology Attachment (PATA) disks and Serial Advanced Technology Attachment (SATA) disks are too slow.

To maintain database integrity, ensure that the storage hardware can withstand failures such as power outages and controller failure. You can improve database performance by using hardware that provides a fast, nonvolatile write cache for both the database and logs. Put the database directories on fault tolerant storage with high-availability features.

It is best to use multiple directories for the database, with four to eight directories for a large Tivoli Storage Manager database. Locate each database directory on a disk volume that uses separate physical disks from other database directories. The Tivoli Storage Manager server database I/O workload is spread over all directories, thus increasing the read and write I/O performance. Having many small capacity physical disks are better than having a few large capacity physical disks with the same rotation speed.

Locate the active log, mirror log, and archive log directories also on high-speed, reliable disks. The failover archive log can be on slower disks, assuming that the archive log is sufficiently large and that the failover log is used infrequently.

The access pattern for the active log is always sequential. Physical placement on the disk is important. It is best to isolate the active log from the database and from the disk storage pools. If they cannot be isolate, then place the active log with storage pools and not with the database.

Enable read cache for the database and recovery log, and enable write cache if the disk subsystems support it.

Restriction: You cannot use raw logical volumes for the database. To reuse space on the disk where raw logical volumes were located for an earlier version of the server, create file systems on the disk first.

Estimating database space requirements

The size of the database depends on the number of client files to be stored and the method by which the server manages them.

If you can estimate the maximum number of files that might be in server storage at any time, you can estimate the database size from the following information:

- Each stored version of a file requires about 600 - 1000 bytes of database space.
- Each cached file, copy storage pool file, active-data pool file, and deduplicated file requires about an additional 100 - 200 bytes of database space.
- Overhead can require up to 50% in additional space.

In the following example for a single client, the computations are probable maximums. In addition, the numbers are not based on using file aggregation. In general, the aggregation of small files reduces the required database space. For details about aggregation, see "How the server groups files before storing" on page 243. Assume the following numbers for a Tivoli Storage Manager system:

Versions of files

Backed up files

Up to 500,000 client files might be backed up. Storage policies call for keeping up to three copies of backed up files:

$$500,000 \text{ files} \times 3 \text{ copies} = 1,500,000 \text{ files}$$

Archived files

Up to 100,000 files might be archived copies of client files.

Space-managed files

Up to 200,000 files migrated from client workstations might be in server storage. File aggregation does not affect space-managed files.

At 1000 bytes per file, the space required for these files is:

$$(1,500,000 + 100,000 + 200,000) \times 1000 = 1.8 \text{ GB}$$

Cached, copy storage pool, active-data pool, and deduplicated files

Cached copies

Caching is enabled in a 5 GB disk storage pool. The high and low migration thresholds of the pool are 90% and 70%. Thus, 20% of the disk pool, or 1 GB, is occupied by cached files.

If the average file size is about 10 KB, about 100,000 files are in cache at any one time.

$$100,000 \text{ files} \times 200 \text{ bytes} = 19 \text{ MB}$$

Copy storage pool files

All primary storage pools are backed up to the copy storage pool:

$$(1,500,000 + 100,000 + 200,000) \times 200 \text{ bytes} = 343 \text{ MB}$$

Active-data pool files

All the active client-backup data in primary storage pools is copied to the active-data pool. Assume that 500,000 versions of the 1,500,000 backup files in the primary storage pool are active.

$$500,000 \times 200 \text{ bytes} = 95 \text{ MB}$$

Deduplicated files

Assume that a deduplicated storage pool contains 50,000 files.

$$50,000 \times 200 \text{ bytes} = 10 \text{ MB}$$

Therefore, these cached files, copy storage pool files, active-data pool files, and deduplicated storage pool files require about an additional 0.5 GB of database space.

Overhead

About 2.3 GB is required for file versions, cached copies, copy storage pool files, and active-data pool files. Allow up to 50% additional space (or 1.2 GB) for overhead.

The database should then have at least 3.5 GB per client.

During SQL queries of the server, intermediate results are stored in temporary tables that require space in the free portion of the database. Therefore, using SQL queries requires additional database space. The more complicated the queries, the greater the space that is required.

Tip:

- In the preceding examples, the results are estimates. The actual size of the database might differ from the estimate because of factors such as the number of directories and the length of the path and file names. As a best practice, periodically monitor your database and adjust its size as necessary.
- If you cannot estimate the numbers of files, you can roughly estimate the database size as from 1% to 5% of the required server storage space. For example, if you need 100 GB of server storage, your database should be 1 - 5 GB. See “Estimating space needs for storage pools” on page 330 for details.

Estimating recovery log space requirements

The recovery log space that you require depends on the amount of client activity with the server.

Active log space

Ensuring that the recovery log has enough space is essential for the server.

The default size of the active log is 16,384 MB (16 GB). Under normal server operations, you are likely to need an active log that is larger than the default. The maximum size of the active log is 131,072 MB (128 GB). The minimum size of the active log is 2048 MB (2 GB).

When estimating the size of the active log, ensure that the active log is large enough to handle not only the amount of concurrent activity that the server typically handles, but also higher workloads that can occur occasionally or under unusual conditions. Try to anticipate the greatest amount of workload that the server might need to handle.

For simple backup and archive activity with no data deduplication, 20 GB for the active log is adequate. If you use data deduplication, and if you deduplicate large objects (for example, image backups), use an active log size that is 20% of the database size.

Monitor the space usage and adjust the size of the active log as needed. To change the size of the active log, see “Increasing the size of the active log” on page 605

Active log mirror space

The active log mirror is a copy of the active log that can be used if the active log files cannot be read. There can be only one active log mirror.

Creating a log mirror is optional. If you increase the active log size, the log mirror size is increased automatically. Be aware that mirroring the log can affect performance because of the doubled I/O activity that is required to maintain the mirror. The additional space that the log mirror requires is another factor to consider when deciding whether to create a log mirror.

Archive log space

The size of the archive log depends on the number of objects stored by client nodes between full backups of the database.

To recover space, a full backup of the database causes obsolete archive log files to be pruned. The archive log files that are included in a backup are automatically pruned on a full database backup cycle. Therefore, the archive log must be large enough to contain the logs generated during a full database backup cycle.

If you perform a full backup of the database every day, the archive log must be large enough to hold the log files for client activity that occurs over two days. Typically, 600 - 4000 bytes of log space are used when an object is stored in the server. Therefore you can estimate a starting size for the archive log using the following calculation:

objects stored per day x 3000 bytes per object x 2 days

For example:

5,000,000 objects/day x 3000 bytes/object x 2 days = 30,000,000,000 bytes,
or 30 GB

It is important to maintain adequate space for the archive log directory. If the drive or file system where the archive log directory is located becomes full and there is no archive failover log directory, the data remains in the active log directory. This condition can cause the active log to fill up, which causes the server to stop.

Archive failover log space

The archive failover log is used by the server if the archive log directory runs out of space.

Specifying an archive failover log directory is optional, but it can prevent problems that occur if the archive log runs out of space. If both the archive log directory and the drive or file system where the archive failover log directory is located become full, the data remains in the active log directory. This condition can cause the active log to fill up, which causes the server to halt. If you use an archive failover log directory, place the archive log directory and the archive failover log directory on different physical drives.

Important: Maintain adequate space for the archive log directory, and consider using an archive failover log directory. If the drive or file system where the archive log directory is located becomes full and either there is no archive failover log directory or it also is full, the log files that are ready to be moved to the archive log instead remain in the active log directory. If the active log becomes full, the server stops.

By monitoring the usage of the archive failover log, you can determine whether additional space is needed for the archive log. The goal is to minimize the need to use the archive failover log by ensuring that the archive log has adequate space.

The locations of the archive log and the archive failover log are set during initial configuration. If you use the DSMSEV LOADFORMAT utility instead of the wizard to configure the server, you specify the ARCHLOGDIR parameter for the archive log directory, and the ARCHFAILOVERLOGDIR parameter for the archive failover log directory. If the archive failover log is not created at initial configuration, you can create it later by specifying the ARCHFAILOVERLOGDIR option in the server options file.

Monitoring the database and recovery log

Monitor the database, log space, and file systems where the directories are located to ensure that space is always available.

You can monitor the database and recovery log space whether the server is online or offline.

- When the Tivoli Storage Manager server is online, you can issue the `QUERY DBSPACE` command to view the total space, used space, and free space for the file systems or drives where your database is located. To view the same information when the server is offline, issue the `DSMSERV DISPLAY DBSPACE` command. The following example shows the output of this command:

```
Location: /tsmdb001
Total Space (MB): 46,080.00
Used Space (MB): 20,993.12
Free Space (MB): 25,086.88

Location: /tsmdb002
Total Space (MB): 46,080.00
Used Space (MB): 20,992.15
Free Space (MB): 25,087.85

Location: /tsmdb003
Total Space (MB): 46,080.00
Used Space (MB): 20,993.16
Free Space (MB): 25,086.84

Location: /tsmdb004
Total Space (MB): 46,080.00
Used Space (MB): 20,992.51
Free Space (MB): 25,087.49
```

- To view more detailed information about the database when the server is online, issue the `QUERY DB` command. The following example shows the output of this command if you specify `FORMAT=DETAILED`:

```
Database Name: TSMDB1
Total Size of File System (MB): 184,320
Space Used by Database (MB): 83,936
Free Space Available (MB): 100,349
Total Pages: 6,139,995
Usable Pages: 6,139,451
Used Pages: 6,135,323
Free Pages: 4,128
Buffer Pool Hit Ratio: 100.0
Total Buffer Requests: 97,694,823,985
Sort Overflows: 0
Lock Escalation: 0
Package Cache Hit Ratio: 100.0
Last Database Reorganization: 06/25/2009 01:33:11
Full Device Class Name: LT01_CLASS
Incrementals Since Last Full: 0
Last Complete Backup Date/Time: 06/06/2009 14:01:30
```

- When the Tivoli Storage Manager server is online, issue the `QUERY LOG FORMAT=DETAILED` command to display the total space, used space, and free space for the active log, and the locations of all the logs. To display the same information when the Tivoli Storage Manager server is offline, issue the `DSMSERV DISPLAY LOG` command. The following example shows the output of this command:

```
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
```

- You can view information about the database on the server console and in the activity log. You can set the level of that information by using the SET DBREPORTMODE command. Specify that no diagnostic information is displayed (NONE), that all diagnostic information is displayed (FULL), or that the only events that are displayed are those that are exceptions and might represent errors (PARTIAL). The default is PARTIAL.

Increasing the size of the database

You can increase the size of the database by creating directories and adding them to the database.

The server can use all the space that is available to the drives or file systems where the database directories are located. To ensure that database space is always available, monitor the space in use by the server and the file systems where the directories are located. The maximum supported size of the database is 2 TB.

The QUERY DB command, shown in “Monitoring the database and recovery log” on page 603, displays number of free pages in the table space and the free space available to the database. If the number of free pages are low and there is a lot of free space available, the database allocates additional space. However, if free space is low, it might not be possible to expand the database.

To increase the size of the database, take the following steps:

1. Create one or more database directories. Locate the directories on separate drives or file systems.
2. Issue the EXTEND DBSPACE to add one or more directories to the database. The directories must be accessible to the user ID of the database manager. Locate the directories on different drives or file systems.

For example, to add two directories to the storage space for the database, issue the following command:

```
extend dbspace /tsmdb005,/tsmdb006
```

Reducing the size of the database

If a significant amount of data has been deleted from the database, consider reducing the database size.

1. Create a file containing a list of directories that represent the new directories. For example, dbdirs.txt.
2. Run a full database backup. For example:

```
backup db devclass=tapeclass type=full
```
3. Halt the server.
4. Remove the database instance.

```
dsmserv removedb TSMDB1
```

5. Restore the database specifying the file containing the directories to be used.
For example:

```
dsmserv restore db todate=today on=dbdirs.txt
```
6. Restart the server.

Increasing the size of the active log

If the log is running out of space, the current transaction is rolled back, and the server issues an error message and halts. You cannot restart the server until the active log size is increased.

To increase the size of the active log while the server is halted, complete the following steps:

1. Issue the DSMSERV DISPLAY LOG offline utility to display the size of the active log.
2. Ensure that the location for the active log has enough space for the increased log size. If a log mirror exists, its location must also have enough space for the increased log size.
3. Halt the server.
4. In the dsmserv.opt file, update the ACTIVELOGSIZE server option to the new maximum size of the active log, in megabytes. For example, to change the active log to its maximum size of 128 GB, enter the following server option:

```
activelogsize 131072
```
5. If you will use a new active log directory, update the directory name specified in the ACTIVELOGDIR server option. The new directory must be empty and must be accessible to the user ID of the database manager.
6. Restart the server.

Log files of 512 MB are automatically defined until the size specified in the ACTIVELOGSIZE server option is reached. If a log mirror exists, log files are also defined automatically.

Backing up the database

Backing up the database at least daily is essential to protect the data that your server manages.

You can restore the database to the latest possible time or to a specific point in time.

Attention: To restore a damaged or lost database you must have a database backup, which includes backups of the logs, and copies of the volume history file and the device configuration file.

Database backups can be full, incremental, or snapshot. In this version of Tivoli Storage Manager, an incremental backup includes all changes since the last full backup. You can schedule the backups to occur automatically, or you can perform them manually.

Full and incremental backups can be used to restore a database.

- To restore a database to the most current time, you need the latest full backup, the latest incremental backup after that full backup, and the active and archive log files.

- To restore a database to point in time, you need the latest full backup before the point in time and the latest incremental backup after the last full backup before the point in time.

To set up your system for database backups specify a device class to be used for the backups. You can also set the percentage of the virtual address space that is dedicated to the database manager processes, and you can define a schedule for backing up the database automatically. For details about these procedures, see “Preparing the system for database backups” and “Scheduling database backups.”

Preparing the system for database backups

To prepare the system for automatic and manual database backups, you must specify the device class to be used. You can also set the percentage of the virtual address space that is dedicated to the database manager processes.

Perform the following setup procedure:

1. Specify the device class to be used for backups, by issuing the SET DBRECOVERY. For example, to specify that the DBBACK device class is to be used, issue this command:

```
set dbrecovery dback
```

If you do not specify a device class with the SET DBRECOVERY command, the backup fails. If you issue the BACKUP DB command with the TYPE=FULL parameter, 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. You can also change the device class to be used for database backups with the SET DBRECOVERY command.

2. By default, the percentage of the virtual address space that is dedicated to all database manager processes is set to 70 - 80 % of system RAM. To change this setting, modify the DBMEMPERCENT server option. If applications other than the Tivoli Storage Manager server are running on the system, ensure that the value allows adequate memory for the other applications.

Scheduling database backups

Set up full or incremental database backups to run on a regular schedule.

When the server is first started, a full backup is begun automatically. Then the database manager backs up the database automatically based on the following values set by Tivoli Storage Manager:

- The active log space consumed since the last backup, which triggers a full database backup
- The active log utilization ratio, which triggers an incremental database backup

In addition, you might want to schedule daily database backups.

Issue the DEFINE SCHEDULE command to schedule your database backups. For example, to set up a schedule to run a full backup to device class FILE every day at 1:00 a.m., enter the following command:

```
define schedule daily_backup type=administrative
cmd="backup db deviceclass=file type=full" starttime=01:00
```

A database backup schedule can also be part of a maintenance script created in the Administration Center.

Backing up the database manually

The database manager automatically backs up the database, and you can define a schedule to back up the database on a regular basis. However, there might be times when you want to run a backup manually.

Use the BACKUP DB command to back up the database manually. For example, to run a full database backup manually, enter the following command:

```
backup db devclass=ltotape type=full volumenames=vol1,vol2,vol3
```

Restoring the database

If a database is damaged or destroyed and a database backup and other files are available, the database can be restored.

Attention: To restore the database, the following files must be available:

- Database backup volumes (last full and incremental, or snapshot)
- Volume history file
- Device configuration file

The database backup include backups of the recovery log.

There are two types of database restore operations: restoring the database to its most current state, and restoring the database to a point in time.

To restore a database to its most current state, see “Restoring a server database to its most current state” on page 776. To restore a database to a point in time, see “Restoring a server database to a point in time” on page 774.

You can use the disaster recovery manager (DRM) function to prepare a plan that can help you to recover your applications if a disaster occurs. See Chapter 26, “Using disaster recovery manager,” on page 801 for details.

Moving the database and recovery log on a server

You can move the database and logs to different locations on the server.

You might want to move the database and logs to take advantage of a larger or faster disk. You have the following options:

- “Moving both the database and recovery log” on page 608
- “Moving only the database” on page 608
- “Moving only the active log” on page 609
- “Moving only the archive log” on page 609
- “Moving only the archive failover log” on page 609

For information about moving a Tivoli Storage Manager server to another machine, see “Moving the Tivoli Storage Manager server to another system” on page 562

Moving both the database and recovery log

You can move the database, active log, and archive logs that are on the same file system to various directories on different file systems for better protection.

1. Back up the database. For example:

```
backup db type=full devclass=files
```
2. Halt the server.
3. Create directories for the database, active logs, and archive logs. The directories must be accessible to the user ID of the database manager. For example:

```
mkdir /tsmdb005  
mkdir /tsmdb006  
mkdir /tsmdb007  
mkdir /tsmdb008  
mkdir /activeLog2  
mkdir /archiveLog2
```
4. Create a file that lists the locations of the database directories. This file will be used if the database must be restored. Enter each location on a separate line. For example, here are the contents of the dbdirs.txt file:

```
/tsmdb005  
/tsmdb006  
/tsmdb007  
/tsmdb008
```
5. Remove the database instance.

```
dsmserv removedb TSMDB1
```
6. Issue the DSMSEV RESTORE DB utility to move the database and create the new active log. For example:

```
dsmserv restore db todate=today on=dbdirs.txt  
activeLog=/activeLog2
```
7. Restart the server.
8. Move the archive logs from the old directory to the new directory. Ensure that you preserve any subdirectory structure. Use and verify a recursive copy directory command:

```
cp -r /archiveLog/* /archiveLog2
```

Moving only the database

You can move only the database to new directories.

To move the database from one location on the server to another location, follow this procedure:

1. Back up the database. For example:

```
backup db type=full devclass=files
```
2. Halt the server.
3. Create directories for the database. The directories must be accessible to the user ID of the database manager. For example:

```
mkdir /tsmdb005  
mkdir /tsmdb006  
mkdir /tsmdb007  
mkdir /tsmdb008
```
4. Create a file that lists the locations of the database directories. This file will be used if the database must be restored. Enter each location on a separate line. For example, here are the contents of the dbdirs.txt file:

```
/tsmdb005  
/tsmdb006  
/tsmdb007  
/tsmdb008
```

5. Remove the database instance.
`dsmserv removedb TSMDB1`
6. Issue the DSMSERV RESTORE DB utility to move the database to the new directories. For example:
`dsmserv restore db todate=today on=dbdir.file`
7. Start the server.

Moving only the active log

You can move only the active log from one directory to another.

1. Halt the server.
2. Create a new active log directory. For example:
`mkdir /activelog2`
3. Update the `dsmserv.opt` option file for the new active log directory. For example:
`activelogdir /activelog2`
4. Restart the server. The active logs are automatically moved by the database manager from the old directory to the new directory.
5. Optional: Remove the old directory.

Moving only the archive log

You can move only the archive log from one directory to another.

1. Halt the server.
2. Create a new archive log directory. For example:
`mkdir /archivelog2`
3. Update the `dsmserv.opt` option file for the new archive log directory. For example:
`archlogdir /archivelog2`
4. Restart the server.
5. Move the archive logs from the old directory to the new directory. Preserve any subdirectory structure. Use and verify a recursive copy directory command. For example:
`cp -r /archivelog/* /archivelog2`
6. Optional: Remove the old directory.

Moving only the archive failover log

Move only the archive failover log from one directory to another.

1. Halt the server.
2. Create a new archive failover log directory. For example:
`mkdir /tsmsserver1/archfaillog2`
3. Update the `dsmserv.opt` option file for the new archive failover log directory. For example:
`archfailoverlogdir /tsmsserver1/archfaillog2`
4. Restart the server.

5. Copy or move the archive log from the old directory to the new directory. Preserve any subdirectory structure. Use and verify a recursive copy directory command. For example:

```
cp -r /tmsserver1/archfaillog/* /tmsserver1/archfaillog2
```
6. Optional: Remove the old directory.

Adding optional logs after server initialization

You can specify an archive failover log and a mirror log if they were not created during server initialization.

Complete the following procedure to add one or both of the logs.

1. Create the directories for the logs. The directories must be accessible to the user ID of the database manager.
2. Halt the server.
3. Add the log server option or options to the server options file, `dsmserv.opt`. For example:

```
archfailoverlogdir /archfailoverlog  
mirrorlogdir /mirrorlog
```
4. Save the server options file.
5. Halt and restart the server.

Transaction processing

A *transaction* is the unit of work exchanged between the client and server.

The log records for a given transaction are moved into stable storage when the transaction is committed. The database information that is stored on disk remains consistent because the server ensures that the recovery log records, which represent the updates to these database pages, are written to disk.

During restart-recovery, the server uses the active and archive log information to maintain the consistency of the server by redoing and, if necessary, undoing ongoing transactions from the time that the server was halted. The transaction is then committed to the database.

Transaction commit is a function of all the log records for that transaction being written to the recovery log. This function ensures that the necessary redo and undo information is available to replay these transaction changes against the database information.

Files moved as a group between client and server

The client program can move multiple files or directories between the client and server before it commits the data to server storage.

A transaction that contains multiple files or directories is called a *transaction group*. Using the `TXNGROUPMAX` server option, you can specify the number of files or directories that are contained within a transaction group. A larger value for the `TXNGROUPMAX` option can affect the performance of client backup, archive, restore, and retrieve operations. You can use the `TXNGROUPMAX` option to increase performance when Tivoli Storage Manager writes to tape. This performance increase can be considerable when you transfer multiple small files.

If you increase the value of TXNGROUPMAX by a large amount, monitor the effects on the recovery log. A larger value can increase utilization of the recovery log, as well as increase the length of time for a transaction to commit. Also consider the number of concurrent sessions to be run. It might be possible to run with a higher TXNGROUPMAX value with a few clients running. However, if there are hundreds of clients running concurrently, you might need to reduce the TXNGROUPMAX to help manage the recovery log usage and support this number of concurrent clients. If the performance effects are severe, they might affect server operations. See “Monitoring the database and recovery log” on page 603 for more information.

The following examples show how the TXNGROUPMAX option can affect performance throughput for operations to tape and the recovery log.

- The TXNGROUPMAX option is set to 512. The MAXSESSIONS option, which specifies the maximum number of concurrent client/server sessions, is set to 5. Five concurrent sessions are processing, and each file in the transaction requires 10 logged database operations. This would be a concurrent load of:

$$20 \times 10 \times 5 = 1000$$

This represents 1000 log records in the recovery log. Each time a transaction commits the data, the server can free 200 log records. Over time and as transactions end, the recovery log can release the space that is used by the oldest transactions. These transactions complete, and the log space usage increases.

- The TXNGROUPMAX option is set to 2000. The MAXSESSIONS option is set to 5. Five concurrent sessions are processing, and each file in the transaction requires 10 logged database operations, resulting in a concurrent load of:

$$2000 \times 10 \times 5 = 20\ 000$$

This represents 100 000 log records in the recovery log. Each time a transaction commits the data, the server can free 20 000 log records. Over time and as transactions end, the recovery log can release the space that is used by the oldest transactions. These transactions complete, and the log space usage increases.

Based on the previous two examples, five concurrent transactions with a TXNGROUPMAX setting of 2000 consume much more space in the recovery log. This increase in log space usage also increases the risk of running out of recovery log space.

The following table shows a comparison of the examples of the preceding TXNGROUPMAX settings. This example becomes more significant if a given log record takes 100 bytes.

Table 54. Example of log bytes that are consumed by five concurrent sessions

TXNGROUPMAX Setting	Number of Log Bytes Consumed
TXNGROUPMAX=20	100,000
TXNGROUPMAX=2000	10,000,000

You can use several server options to tune server performance and reduce the risk of running out of recovery log space:

- Use the THROUGHPUTTIMETHRESHOLD and THROUGHPUTDATATHRESHOLD options with the TXNGROUPMAX option to prevent a slower performing node from holding a transaction open for extended periods.

- Increase the size of the recovery log when you increase the TXNGROUPMAX setting.

Evaluate the performance and characteristics of each node before increasing the TXNGROUPMAX setting. Nodes that have only a few larger objects to transfer do not benefit as much as nodes that have multiple, smaller objects to transfer. For example, a file server benefits more from a higher TXNGROUPMAX setting than does a database server that has one or two large objects. Other node operations can consume the recovery log at a faster rate. Be careful when increasing the TXNGROUPMAX settings for nodes that often perform high log-usage operations. The raw or physical performance of the disk drives that are holding the database and recovery log can become an issue with an increased TXNGROUPMAX setting. The drives must handle higher transfer rates to handle the increased load on the recovery log and database.

You can set the TXNGROUPMAX option as a global server option value, or you can set it for a single node. Refer to the REGISTER NODE command and the server options in the *Administrator's Reference*. For optimal performance, specify a lower TXNGROUPMAX value (between 4 and 512). Select higher values for individual nodes that can benefit from the increased transaction size.

Chapter 22. Monitoring the Tivoli Storage Manager server

Administrators can monitor the server to learn information about server processes.

- To find the status of operations
- To display information about objects
- To monitor the record of activity
- To select the types of events to save
- To select a location to save events
- To view historical reports and real-time monitoring information for the Tivoli Storage Manager servers and clients.

Tasks:
“Using IBM Tivoli Storage Manager queries to display information”
“Using SQL to query the IBM Tivoli Storage Manager database” on page 618
“Using the IBM Tivoli Storage Manager activity log” on page 623
“Logging IBM Tivoli Storage Manager events to receivers” on page 626
“Monitoring IBM Tivoli Storage Manager accounting records” on page 645
“Daily monitoring scenario” on page 646
“Reporting and monitoring servers and clients” on page 647

Using IBM Tivoli Storage Manager queries to display information

IBM Tivoli Storage Manager provides QUERY commands to display formatted information about definitions, settings, processes, and status.

For some commands, you can display the information in either a standard or detailed format. The standard format presents less information than the detailed format, and is useful in displaying an overview of many objects. For displaying more information about a particular object, use the detailed format when supported by a given command.

For information about creating customized queries of the database, see “Using SQL to query the IBM Tivoli Storage Manager database” on page 618.

Requesting information about IBM Tivoli Storage Manager definitions

During Tivoli Storage Manager system setup, an administrator can define many objects, for example, storage management policies, storage pools, and device classes. Tivoli Storage Manager provides queries that display information about these objects.

Most of these definition queries let you request standard format or detailed format. Standard format limits the information and usually displays it as one line per object. Use the standard format when you want to query many objects, for example, all registered client nodes. Detailed format displays the default and specific definition parameters. Use the detailed format when you want to see all the information about a limited number of objects.

Here is an example of the standard output for the QUERY NODE command:

Node Name	Platform	Policy Domain Name	Days Since Last Access	Days Since Password Set	Locked?
CLIENT1	AIX	STANDARD	6	6	No
GEORGE	Linux86	STANDARD	1	1	No
JANET	HPUX	STANDARD	1	1	No
JOE2	Mac	STANDARD	<1	<1	No
TOMC	WinNT	STANDARD	1	1	No

Here is an example of the detailed output for the QUERY NODE command:

```

Node Name: JOE
Platform: WinNT
Client OS Level: 5.00
Client Version: Version 5, Release 1, Level 5.0
Policy Domain Name: STANDARD
Last Access Date/Time: 05/19/2002 18:55:46
Days Since Last Access: 6
Password Set Date/Time: 05/19/2002 18:26:43
Days Since Password Set: 6
Invalid Sign-on Count: 0
Locked?: No
Contact:
Compression: Client's Choice
Archive Delete Allowed?: Yes
Backup Delete Allowed?: No
Registration Date/Time: 03/19/2002 18:26:43
Registering Administrator: SERVER_CONSOLE
Last Communication Method Used: Tcp/Ip
Bytes Received Last Session: 108,731
Bytes Sent Last Session: 698
Duration of Last Session (sec): 0.00
Pct. Idle Wait Last Session: 0.00
Pct. Comm. Wait Last Session: 0.00
Pct. Media Wait Last Session: 0.00
Optionset:
URL: http://client.host.name:1581
Node Type: Client
Password Expiration Period: 60
Keep Mount Point?: No
Maximum Mount Points Allowed: 1
Auto Filespace Rename: No
Validate Protocol: No
TCP/IP Name: JOE
TCP/IP Address: 9.11.153.39
Globally Unique ID: 11.9c.54.e0.8a.b5.11.d6.b3.c3.00.06.29.45.c1.5b
Transaction Group Max: 0
Session Initiation: ClientOrServer
HLADDRESS:
LLADDRESS:

```

Requesting information about client sessions

When administrators or users access Tivoli Storage Manager, an administrative or client node session is established with the server. The server assigns each client session a unique session number.

You can use the QUERY SESSION command to request information about client sessions. Figure 76 on page 615 shows a sample client session report.

Sess Number	Comm. Method	Sess State	Wait Time	Bytes Sent	Bytes Recvd	Sess Type	Platform	Client Name
3	Tcp/Ip	IdleW	9 S	7.8 K	706	Admin	WinNT	TOMC
5	Tcp/Ip	IdleW	0 S	1.2 K	222	Admin	AIX	GUEST
6	Tcp/Ip	Run	0 S	117	130	Admin	Mac2	MARIE

Figure 76. Information about client sessions

Check the *wait time* to determine the length of time (seconds, minutes, hours) the server has been in the current state. The *session state* reports status of the session and can be one of the following:

Start Connecting with a client session.

Run Running a client request.

End Ending a client session.

RecvW

Waiting to receive an expected message from the client while a database transaction is in progress. A session in this state is subject to the COMMTIMEOUT limit.

SendW

Waiting for acknowledgment that the client has received a message sent by the server.

MediaW

Waiting for removable media to become available.

IdleW Waiting for communication from the client, and a database transaction is *not* in progress. A session in this state is subject to the IDLETIMEOUT limit.

For example, Tivoli Storage Manager cancels the client session if the IDLETIMEOUT option is set to 30 minutes, and a user does not initiate any operations within those 30 minutes. The client session is automatically reconnected to the server when it starts to send data again.

Requesting information about server processes

When a command runs in the foreground (synchronous command execution), you are unable to issue any other commands until the process completes. When a command runs in the background (asynchronous command execution), you can issue other commands while the process completes.

Most commands run in the foreground, but others generate background processes. In some cases, you can specify that a process run in the foreground. Tivoli Storage Manager issues messages that provide information about the start and end of processes. In addition, you can request information about active background processes. If you know the process ID number, you can use the number to limit the search. However, if you do not know the process ID, you can display information about all background processes by issuing the QUERY PROCESS command.

Figure 77 on page 616 shows a server background process report after a DELETE FILESPACE command was issued. The report displays a process ID number, a description, and a completion status for each background process.

Process Number	Process Description	Status
2	DELETE FILESPACE	Deleting filespace DRIVE_D for node CLIENT1: 172 files deleted.

Figure 77. Information about background processes

Requesting information about server settings

Any administrator can request general server information, most of which is defined by SET commands, by issuing the QUERY STATUS command.

The displayed text includes a variety of information, such as:

- The server name and TCP/IP settings
- Server password and authentication settings
- Client node settings
- Activity log settings and status
- License audits and compliance status
- Client/server session limits
- Central scheduler settings and status
- Recovery log and backup trigger modes
- Refresh settings and status
- Table of contents retention period
- Machine globally unique ID at last startup
- Archive retention protection status
- Data encryption strength

This list is not all-inclusive. For a detailed explanation of the QUERY STATUS command, see the *Administrator's Reference*.

Querying server options

Use the QUERY OPTION command to display information about one or more server options.

Task	Required Privilege Class
Query server options	Any administrator

You can issue the QUERY OPTION command with no operands to display general information about all defined server options. You also can issue it with a specific option name or pattern-matching expression to display information on one or more server options. You can set options by editing the server options file.

See *Administrator's Reference* for more information.

Querying the system

The QUERY SYSTEM command combines multiple queries of your Tivoli Storage Manager system into a single command. This command can be used to collect statistics and provide information for problem analysis by IBM service.

When you enter the QUERY SYSTEM command, the server issues the following queries:

QUERY ASSOCIATION

Displays all client nodes that are associated with one or more client schedules

QUERY COPYGROUP

Displays all backup and archive copy groups (standard format)

QUERY DB

Displays information about the database (detailed format)

QUERY DBSPACE

Displays display information about the directories used by the database.

QUERY DEVCLASS

Displays all device classes (detailed format)

QUERY DOMAIN

Displays all policy domains (standard format)

QUERY LOG

Displays information about the recovery log (detailed format)

QUERY MGMTCLASS

Displays all management classes (standard format)

QUERY OPTION

Displays all server options

QUERY PROCESS

Displays information about all active background processes

QUERY SCHEDULE

Displays client schedules (standard format)

QUERY SESSION

Displays information about all administrative and client node sessions in standard format

QUERY STATUS

Displays general server parameters, such as those defined by SET commands

QUERY STGPOOL

Displays information about all storage pools (detailed format)

QUERY VOLUME

Displays information about all storage pool volumes (standard format)

SELECT

Displays the results of two SQL queries:

```
select platform_name,count(*) from nodes group by platform_name
select stgpool_name,devclass_name,count(*) from volumes
group by stgpool_name,devclass_name
```

The first command displays the number of client nodes by platform.

The second command displays the name and associated device class of all storage pools having one or more volumes assigned to them.

Using SQL to query the IBM Tivoli Storage Manager database

You can use a standard SQL SELECT statement to get information from the database.

IBM Tivoli Storage Manager Versions 6.1 and later use the DB2 open database connectivity (ODBC) driver to query the database and display the results.

DB2 provides its own ODBC driver which can also be used to access the Tivoli Storage Manager server DB2 database. For more information on the DB2 native ODBC driver, refer to DB2 documentation at: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>. Search on *Introduction to DB2 CLI and ODBC*

Using SELECT commands

SELECT commands allow you to create and format customized queries of the IBM Tivoli Storage Manager database.

For SELECT statement syntax and guidelines, refer to DB2 documentation: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>. Search on the term *Select-statement*.

You can issue the SELECT command from the command line of an administrative client. You cannot issue this command from the server console.

Learning what information is available: system catalog tables

System catalog tables provide information about information that is available in the database.

To help you find what information is available in the database, 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 enumerated type and the order of the values for each type.

You can issue the SELECT command to query these tables and determine the location of the information that you want. For example, to get a list of all tables available for querying in the database *TSMDB1* enter the following command:

```
select tabname from syscat.tables where tabschema='TSMDB1' and type='V'
```

The results are:

```

TABNAME: ACTLOG
TABNAME: AF_VOL_SEGMENTS
TABNAME: ARCHDESC_NAMEVIEW
TABNAME: ARCHIVES
TABNAME: ARCHIVE_NAMEVIEW
TABNAME: AR_COPYGROUPS
TABNAME: ASSOCIATIONS
TABNAME: AS_VOLUME_ASSIGNMENT
TABNAME: BACKUPS
TABNAME: BACKUPSETS
TABNAME: BACKUP_NAMEVIEW
TABNAME: BU_COPYGROUPS
TABNAME: CLIENT ADMINISTRATORS
TABNAME: CONTENTS
TABNAME: DB
TABNAME: DEVCLASSES
TABNAME: DF_VOL_CONTENTS
TABNAME: DRIVES
TABNAME: DRMSTATUS
TABNAME: EVENTS
TABNAME: FILESPACEVIEW
TABNAME: GROUPMEMBER
TABNAME: LIBRARIES

```

You can also issue the `SELECT` command to query columns. For example, to get a list of columns for querying in the database `TSMDB1` and the table name `ACTLOG`, enter the following command:

```
select colname from syscat.columns where tabschema='TSMDB1'and tabname='ACTLOG'
```

The results are:

```

COLNAME: DATE_TIME
COLNAME: DOMAINNAME
COLNAME: MESSAGE
COLNAME: MSGNO
COLNAME: NODENAME
COLNAME: ORIGINATOR
COLNAME: OWNERNAME
COLNAME: PROCESS
COLNAME: SCHEDNAME
COLNAME: SERVERNAME
COLNAME: SESSID
COLNAME: SESSION
COLNAME: SEVERITY

```

Customizing queries using the `SELECT` command

With the `SELECT` command, you can customize a wide variety of queries.

This section shows two examples.

For many more examples of the command, see the *Administrator's Reference*.

Example 1: Find the number of nodes by type of operating system by issuing the following command:

```
select platform_name,count(*) as "Number of Nodes" from nodes
group by platform_name
```

This command gives results like the following:

PLATFORM_NAME	Number of Nodes
OS/2	45
AIX	90
Windows	35

Example 2: For all active client sessions, determine how long they have 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
```

This command gives results like the following:

```

      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
```

Note: When issuing SELECT * FROM DB statements, the output column PHYSICAL_VOLUMES is included for legacy compatibility purposes only. While Tivoli Storage Manager Versions 6.1 and later no longer reference physical volumes, the number of physical volumes listed represents the DBSPACE locations defined to the server.

For example:

```

DATABASE_NAME: mgsA62
TOT_FILE_SYSTEM_MB: 511872
USED_DB_SPACE_MB: 448
FREE_SPACE_MB: 452802
PAGE_SIZE: 16384
TOTAL_PAGES: 32772
USABLE_PAGES: 32636
USED_PAGES: 24952
FREE_PAGES: 768
BUFF_HIT_RATIO: 99.7
TOTAL_BUFF_REQ: 385557
SORT_OVERFLOW: 0
LOCK_ESCALATION: 0
PKG_HIT_RATIO: 99.8
LAST_REORG:
FULL_DEV_CLASS:
NUM_BACKUP_INCR: 0
LAST_BACKUP_DATE:
PHYSICAL_VOLUMES: 1

```

Using SELECT commands in IBM Tivoli Storage Manager scripts

A Tivoli Storage Manager script is one or more commands that are stored as an object in the database. You can define a script that contains one or more SELECT commands.

A script can be run from an administrative client or the server console. You can also include it in an administrative command schedule to run automatically. See “IBM Tivoli Storage Manager server scripts” on page 578 for details.

You can also use the Administration Center to run scripts.

Tivoli Storage Manager is shipped with a file that contains a number of sample scripts. The file, `scripts.smp`, is in the server directory. To create and store the scripts as objects in your server's database, issue the `DSMSERV RUNFILE` command during installation:

```
> dsmserv runfile scripts.smp
```

You can also run the file as a macro from an administrative command line client:

```
macro scripts.smp
```

The sample scripts file contains Tivoli Storage Manager commands. These commands first delete any scripts with the same names as those to be defined, then define the scripts. The majority of the samples create SELECT commands, but others do such things as back up storage pools. You can also copy and change the sample scripts file to create your own scripts.

Here are a few examples from the sample scripts file:

```

def script q_inactive_days '/* -----*/'
upd script q_inactive_days '/* Script Name: Q_INACTIVE */'
upd script q_inactive_days '/* Description: Display nodes that have not */'
upd script q_inactive_days '/*   accessed Tivoli Storage Manager for a */'
upd script q_inactive_days '/*   specified number of days */'
upd script q_inactive_days '/* Parameter 1: days */'
upd script q_inactive_days '/* Example:   run q_inactive_days 5 */'
upd script q_inactive_days '/* -----*/'
upd script q_inactive_days "select node_name,lastacc_time from nodes where -"
upd script q_inactive_days " cast((current timestamp-lastacc_time)days as -"
upd script q_inactive_days " decimal) >= $1 "

```

```

/* Display messages in the activity log of severity X or Y */
def script q_msg_sev desc='Show msgs in the activity log of severity X or Y'
upd script q_msg_sev '/* -----*/'
upd script q_msg_sev '/* Script Name: Q_MSG_SEV */'
upd script q_msg_sev '/* Description: Display messages in the */'
upd script q_msg_sev '/* activity log that have either */'
upd script q_msg_sev '/* of two specified severities. */'
upd script q_msg_sev '/* Parameter 1: severity 1 */'
upd script q_msg_sev '/* Parameter 2: severity 2 */'
upd script q_msg_sev '/* where severity is I, W, E, S, or D */'
upd script q_msg_sev '/* Example: run q_msg_sev S E */'
upd script q_msg_sev '/* -----*/'
upd script q_msg_sev "select date_time,msgno,message from actlog -"
upd script q_msg_sev " where severity=upper('$1') or severity=upper('$2')"
```

Querying the SQL activity summary table

You can query the SQL activity summary table to view statistics about client operations and server processes.

Some of the client operations recorded to the table are BACKUP, RESTORE, ARCHIVE and RETRIEVE. Server processes include MIGRATION, RECLAMATION and EXPIRATION.

To list column names and their descriptions from the activity summary table, enter the following command:

```
select colname,remarks from columns where tablename='summary'
```

Here are a few example queries of the activity summary table.

- To display all events starting at 00:00 a.m. of the current day until the present time, enter:

```
select * from summary
```

The result might look like this:

```

START_TIME: 2008-10-10 10:48:52.000000
END_TIME: 2008-10-10 10:48:56.000000
ACTIVITY: BACKUP
NUMBER: 10
ENTITY: NODE1
COMMMETH: Tcp/Ip
ADDRESS: ibm-164391ac47a.tucson.ibm.com:2515
SCHEDULE_NAME:
EXAMINED: 3
AFFECTED: 3
FAILED: 0
BYTES: 36631067
IDLE: 0
MEDIWA: 0
PROCESSES: 2
SUCCESSFUL: YES
VOLUME_NAME:
DRIVE_NAME:
LIBRARY_NAME:
LAST_USE:
COMM_WAIT: 2
NUM_OFFSITE_VOLS:
```

ANS8002I Highest return code was 0.

- To display all events starting at or after 00:00 a.m. on October 10, 2008 until the present time, enter:

```
select * from summary where start_time>='2008-10-10 00:00:00'
```

You can determine how long to keep information in the summary table. For example, to keep the information for 5 days, enter the following command:
set summaryretention 5

To keep no information in the table, specify a value of 0.

Tivoli Storage Manager does not create records in the SQL activity summary table for manual backups or for successful scheduled backups of 0 bytes. Records are created in the summary table for successful scheduled backups only if data is backed up.

Creating output for use by another application

You can redirect the output of SELECT commands for use in another program (for example, a spreadsheet or database program). The use of command output redirection and one of the delimited output format options lets you create queries whose output can be further processed in other applications.

For example, based on the output of a SELECT command, a spreadsheet program could produce graphs of average file sizes and file counts summarized by type of client platform. When using another program for data formatting, the output to be used should be written in a format that is easily processed. Two standard formats for tabular data files are *comma-separated values* (CSV) and *tab-separated values* (TSV). Most modern applications that can import tabular data can read one or both of these formats.

Use the administrative client command line options `-COMMADELIMITED` or `-TABDELIMITED` to select one of these formats for tabular query output. All tabular output created during the administrative session will be formatted into either comma-separated or tab-separated values.

For details about using command line options and redirecting command output, see the *Administrator's Reference*

Using the IBM Tivoli Storage Manager activity log

The activity log contains all messages normally sent to the server console during server operation. The only exceptions are responses to commands entered at the console, such as responses to QUERY commands.

Task	Required Privilege Class
Request information from the activity log	Any administrator
Set the activity log retention period	System
Set the activity log size limit	system

Examples of messages sent to the activity log include:

- When client sessions start or end
- When migration starts and ends
- When backup versions expire
- What data is exported to tape
- When expiration processing is performed
- What export or import processing is performed

Any error messages sent to the server console are also stored in the activity log.

Use the following sections to adjust the size of the activity log, set an activity log retention period, and request information about the activity log.

Requesting information from the activity log

You can request information stored in the activity log.

To minimize processing time when querying the activity log, you can:

- Specify a time period in which messages have been generated. The default for the QUERY ACTLOG command shows all activities that have occurred in the previous hour.
- Specify the message number of a specific message or set of messages.
- Specify a string expression to search for specific text in messages.
- Specify the QUERY ACTLOG command from the command line for large queries instead of using the graphical user interface.
- Specify whether the originator is the server or client. If it is the client, you can specify the node, owner, schedule, domain, or session number. If you are doing client event logging to the activity log and are only interested in server events, then specifying the server as the originator will greatly reduce the size of the results.

For example, to review messages generated on May 30 between 8 a.m. and 5 p.m., enter:

```
query actlog begindate=05/30/2002 enddate=05/30/2002  
begintime=08:00 endtime=17:00
```

To request information about messages related to the expiration of files from the server storage inventory, enter:

```
query actlog msgno=0813
```

Refer to *Messages* for message numbers.

You can also request information only about messages logged by one or all clients. For example, to search the activity log for messages from the client for node JEE:

```
query actlog originator=client node=jee
```

Requesting session and process information

Often, the messages issued on the behalf of a session or process do not indicate (in the message itself) the session or process to which they are related. To help you correlate messages with the session or process for which they are issued, the server records additional information about messages that can be generated by querying the activity log.

For example, an error message issued because of a volume mount failure for a client session operation does not have any explicit information to tie it back to the session that was performing the action. Scenario 1: Client FRED has been having problems getting its scheduled backup to complete. During the nightly scheduled backup's, there are 1000 different sessions running for scheduled clients and TDPs, and data is being stored to various destinations of DISK and TAPE. For the scheduled backups when FRED has failed, it has not been clear from the activity log which error messages were issued that related to FRED. Using this improvement to the activity log information, an administrator would only need to query the activity log to determine what session or sessions FRED was using for

the nightly backup. And then for each session, they could issue a "QUERY ACTLOG SEARCH="(SESSION: sessNum)" or they could issue "SELECT * from ACTLOG where SESSION=sessNum". Either of these would report ALL the messages that were issued in relation to that session.

You can also request information about a client session. For example, to search the activity log for a message that were issued in relation to a session, enter:

```
query actlog search="(SESSION:4)"
```

This command gives results like the following:

```
EXAMPLE 2 (Client SESSION that performs a backup)
09/23/2003 10:26:38 ANR0406I Session 4 started for node FRED (WinNT) (Tcp/Ip
colind(2463)). (SESSION: 4)
09/23/2003 10:26:40 ANR8493I FILE volume C:\CODE\522\00000000.BFS mounted in
drive OUTFILE4 (FILE) in library OUTFILE. (SESSION: 4)
09/23/2003 10:26:40 ANR8340I FILE volume C:\CODE\522\00000000.BFS mounted.
(SESSION: 4)
09/23/2003 10:26:40 ANR8468I FILE volume C:\CODE\522\00000000.BFS dismantled
from drive OUTFILE4 (FILE) in library OUTFILE. (SESSION:
4)
09/23/2003 10:26:40 ANR0403I Session 4 ended for node FRED (WinNT). (SESSION:
4)
```

Setting a retention period for the activity log

You can use the SET ACTLOGRETENTION command to specify how long activity log information is kept in the database.

Activity log management is retention-based when the optional parameter MGMTSTYLE is set to its default value, DATE. The server automatically deletes messages from the activity log once the number of days that are specified pass. At installation, activity log management is retention-based, and the retention period is set to one day. To change the retention period to 10 days, for example, enter:

```
set actlogretention 10
```

To disable activity log retention, set the SET ACTLOGRETENTION command to zero. To display the current retention period and size of the activity log, query the server status.

Note: With retention-based management, you lose some control over the amount of space that the activity log occupies. For more information on size-based activity log management, see "Setting a size limit for the activity log."

Setting a size limit for the activity log

You can use size-based activity log management as an alternative to retention-based management. This allows greater control over the amount of space that the activity log occupies.

The server will periodically remove the oldest activity log records until the activity log size no longer exceeds the configured maximum size allowed. To manage the activity log by size, the parameter MGMTSTYLE must be set to the value SIZE. To change the maximum size of the activity log to 12 MB, for example, enter:

```
set actlogretention 12 mgmtstyle=size
```

To disable activity log retention, set the SET ACTLOGRETENTION command to zero. To display the current and maximum size of the activity log, query the server status.

Note: With size-based management, you lose some control over the length of time that activity log messages are kept. For more information on retention-based activity log management, see “Setting a retention period for the activity log” on page 625.

Logging IBM Tivoli Storage Manager events to receivers

The server and client messages provide a record of Tivoli Storage Manager activity that you can use to monitor the server. You can log server messages and most client messages as *events* to one or more repositories called *receivers*.

You can log the events to any combination of the following receivers:

Tivoli Storage Manager server console and activity log

See “Logging events to the IBM Tivoli Storage Manager server console and activity log” on page 628.

File and user exits

See “Logging events to a file exit and a user exit” on page 629.

Tivoli event console

See “Logging events to the Tivoli Enterprise Console” on page 630.

Event server receiver (Enterprise Event Logging)

Routes the events to an event server. See “Enterprise event logging: logging events to another server” on page 639.

Simple Network Management Protocol (SNMP)

See “Logging events to an SNMP manager” on page 634.

In addition, you can filter the types of events to be enabled for logging. For example, you might enable only severe messages to the event server receiver and one or more specific messages, by number, to another receiver. Figure 78 on page 627 shows a possible configuration in which both server and client messages are filtered by the event rules and logged to a set of specified receivers.

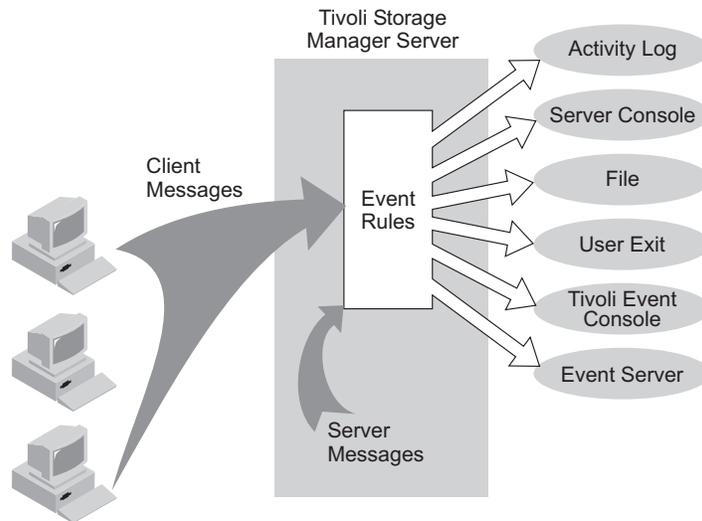


Figure 78. Event logging overview

Task	Required Privilege Class
Enable or disable events	System
Begin or end event logging	

You can control event logging through the following actions:

1. Enable or disable logging for one or more event types and for one or more receivers. See “Enabling and disabling events.”
2. Begin or end logging to one or more receivers. See “Beginning and ending event logging” on page 628.

Enabling and disabling events

You can enable and disable events using the ENABLE EVENTS and DISABLE EVENTS commands.

When you enable or disable events, you can specify the following:

- A message number or an event severity (ALL, INFO, WARNING, ERROR, or SEVERE).
- Events for one or more client nodes (NODENAME) or for one or more servers (SERVERNAME).

To enable or disable events, issue the ENABLE EVENTS and DISABLE EVENTS commands. For example,

- To enable event logging to a user exit for all error and severe server messages, enter:
enable events userexit error,severe
- To enable event logging to a user exit for severe client messages for all client nodes, enter:
enable events userexit severe nodename=*
- To disable event logging to a user exit for error server messages, enter
disable events userexit error

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. Certain events, such as messages that are issued during server startup and shutdown, automatically go to the console. They do not go to other receivers, even if they are enabled.

Note: Server messages in the SEVERE category and message ANR9999 can provide valuable diagnostic information if there is a serious problem. For this reason, you should not disable these messages. Use the SET CONTEXTMESSAGING ON command to get additional information that could help determine the cause of ANR9999D messages. The IBM 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.

Beginning and ending event logging

The BEGIN EVENTLOGGING and END EVENTLOGGING commands can be used to log events when event logging is not automatically started during server startup.

At server startup, event logging begins automatically to the server console and activity log and for any receivers that are started based on entries in the server options file. A receiver for which event logging has begun is an *active receiver*.

To begin logging events to receivers for which event logging is not started automatically, issue the BEGIN EVENTLOGGING command. You can also use this command after you have disabled event logging to one or more receivers. To end event logging for an active receiver issue the END EVENTLOGGING command.

For example,

- To begin logging events to the event server, enter:
`begin eventlogging eventserver`
- To end logging events to the event server, enter:
`end eventlogging eventserver`

Logging events to the IBM Tivoli Storage Manager server console and activity log

Logging events to the server console and activity log begins automatically at server startup.

Enabling client events to the activity log will increase the database utilization. You can set a retention period or size limit for the log records by using the SET ACTLOGRETENTION command (see “Setting a retention period for the activity log” on page 625 and “Setting a size limit for the activity log” on page 625). At server installation, activity log management is retention-based, and this value is set to one day. If you increase the retention period or the size limit, utilization is further increased. For more information about the activity log, see “Using the IBM Tivoli Storage Manager activity log” on page 623.

You can disable server and client events to the server console and client events to the activity log. However, you cannot disable server events to the activity log. Also, certain messages, such as those issued during server startup and shutdown and responses to administrative commands, will still be displayed at the console even if disabled.

To enable all error and severe client events to the console and activity log, you can issue the ENABLE EVENTS command. See the Administrator's Reference for more information.

Logging events to a file exit and a user exit

A file exit is a file that receives all the information related to its enabled events. You can log events to a file exit and a user exit.

Be aware that this file can rapidly grow in size depending on the events enabled for it. There are two versions of the file exit: binary and text. The binary file exit stores each logged event as a record, while the text file exit stores each logged event as a fixed-sized, readable line. For more information about the text file exit, see “Readable text file exit (FILETEXTEXIT) format” on page 644.

See “Adding a file exit or user exit option.”

Adding a file exit or user exit option

File and user exits receive event data in the same data block structure. Setting up logging for these receivers is similar.

1. Add an option for the exit to the server options file:
 - **For a file exit:** Add either the FILEEXIT option (for a binary file exit) or FILETEXTEXIT (for a text file exit) option.
 - a. Specify whether event logging to the file exit receiver begins automatically at server startup. The parameters are YES and NO. If you do not specify YES, you must begin event logging manually by issuing the BEGIN EVENTLOGGING command.
 - b. Specify the file where each logged event is to be stored.
 - c. Specify how files will be stored if the file being stored already exists. REPLACE will overwrite the existing file, APPEND will append data to the existing file, and PRESERVE will not overwrite the existing file.

For example,

```
fileexit yes /tsm/server/data replace
```

```
filetextexit yes /tsm/server/data replace
```

- **For a user exit:** Add the USEREXIT option.
 - Specify whether event logging to the user exit receiver begins automatically at server startup. The parameters for this option are YES and NO. If you do not specify YES, you must begin event logging manually by issuing the BEGIN EVENTLOGGING command.
 - Specify the name of the user-exit function in the service program.
 - Specify a module name of the user exit. This is the name of a shared library containing the exit.

For example,

```
userexit no fevent.exit
```

2. Enable events for the receiver. You must specify the name of the user exit in the USEREXIT server option and the name of the file in the FILEEXIT server option. Here are two examples:

```
enable events file error
```

```
enable events userexit error,severe
```

You can also enable events to one or more client nodes or servers by specify the NODENAME OR SERVERNAME parameter. See “Enabling and disabling events” on page 627 for more information.

- If you did not specify YES in the server option, begin event logging. For example, to begin event logging for a user-defined exit, enter:

```
begin eventlogging userexit
```

See “Beginning and ending event logging” on page 628 for more information.

Logging events to the Tivoli Enterprise Console

Tivoli Storage Manager includes the Tivoli receiver, a Tivoli Enterprise Console adapter for sending events to the Tivoli Enterprise Console. You can specify the events to be logged based on their source.

Application clients, Data Protection for IBM ESS for DB2, and Data Protection for IBM ESS for Oracle must have enhanced Tivoli Enterprise Console support enabled in order to route the events to the Tivoli Enterprise Console. Because of the number of messages, you should *not* enable all messages from a node to be logged to the Tivoli Enterprise Console.

The valid event names are:

Event Name	Source
TSM_SERVER_EVENT	Tivoli Storage Manager server
TSM_CLIENT_EVENT	Tivoli Storage Manager clients
TSM_APPL_EVENT	Tivoli Storage Manager application program interface
TSM_TDP_DOMINO_EVENT	Data Protection for Lotus Domino
TSM_TDP_EXCHANGE_EVENT	Data Protection for Microsoft Exchange Server
TSM_TDP_INFORMIX_EVENT	Data Protection for Informix®
TSM_TDP_ORACLE_EVENT	Data Protection for Oracle
TSM_TDP_SQL_EVENT	Data Protection for Microsoft SQL Server
TSM_TDP_SAP_R3_EVENT	Data Protection for mySAP.com Technology and Data Protection for IBM ESS for mySAP.com Technology
TSM_TDP_ESS_DB2_EVENT	Data Protection for IBM ESS for DB2
TSM_TDP_ESS_ORACLE_EVENT	Data Protection for IBM ESS for Oracle

Controlling the format of events

The server options UNIQUETECEVENTS and UNIQUETDPTECEVENTS are available to control the format of events sent from the Tivoli Storage Manager server to the Tivoli Enterprise Console.

Enabling either of these options not only changes the event class format, but also generates a unique event class for individual Tivoli Storage Manager messages for the client, the server, application clients, Data Protection for IBM ESS for DB2, Data Protection for IBM ESS for Oracle, and Data Protection for IBM ESS for R/3.

Option Name	Function
UNIQUETECEVENTS	Changes the event class format and generates a unique event class for client, server, and some Data Protection messages

Option Name	Function
UNIQUETDPTECEVENTS	Changes the event class format and generates a unique event class for all client, server, and all Data Protection messages

Setting UNIQUETDPTECEVENTS to YES will dynamically set UNIQUETECEVENTS to YES. However, the Tivoli Storage Manager will not update the server options file to reflect this change.

Depending on your particular environment, enabling one or both of these options can facilitate rule-evaluation performance on the Tivoli Enterprise Console server. Enabling one or both of these options can also incur a performance penalty during event reception on the Tivoli Enterprise Console server. Test the options in your own environment. It is possible that you might not benefit from enabling either of these two options.

If the UNIQUETECEVENTS option is enabled, unique events have the following event class format:

```
TSM_SERVER_ANR####
TSM_CLIENT_ANE####
TSM_APPL_ANE####
TSM_TDP_DOMINO_ACD####
TSM_TDP_EXCHANGE_ACN####
TSM_TDP_ORACLE_ANS####
TSM_TDP_INFORMIX_ANS####
TSM_TDP_SQL_ACO####
```

where ##### represents the message number.

If the UNIQUETDPTECEVENTS option is enabled, the messages logged to the Tivoli Storage Manager server for Data Protection for IBM ESS for DB2, Data Protection for IBM ESS for Oracle, and Data Protection for R/3 will have unique events with the following formats:

```
TSM_TDP_ESS_DB2_EEP####TSM_TDP_ESS_ORACLE_EEO####
TSM_TDP_SAP_R3_BKI#### (includes messages for Data Protection for IBM ESS for R/3)
```

where ##### represents the message number. For exact details of the event class format, look at the appropriate baroc file.

Application clients can issue unique events in the following ranges. All events follow the IBM 3.4 naming convention, which uses a three-character prefix followed by four digits.

Application client	Event range
Data Protection for Microsoft Exchange Server	ACN3500 to ACN3649
Data Protection for Lotus Domino	ACD5200 to ACD5299
Data Protection for Microsoft SQL Server	ACO3000 to ACO3999
Data Protection for Oracle	ANS0500 to ANS0599
Data Protection for Informix	ANS0600 to ANS0699

If UNIQUETDPTECEVENTS is enabled, Data Protection for IBM ESS for DB2, Data Protection for IBM ESS for Oracle, and Data Protection for R/3 can issue unique events in the following ranges:

Application client	Event range
Data Protection for IBM ESS for DB2	EEP0000 to EEP9999
Data Protection for IBM ESS for Oracle	EEO0000 to EEO9999
Data Protection for R/3 and Data Protection for IBM ESS for R/3	BKI0000 to BKI9999

As noted, enabling UNIQUETDPTECEVENTS also enables UNIQUETECEVENTS. This means that all recognized Data Protection messages will be sent as unique events from the Tivoli Storage Manager server.

Based upon the setting of the option or options on the Tivoli Storage Manager server, the Tivoli Enterprise Console administrator must create a rule base using one of the following baroc files:

UNIQUETDPTECEVENTS Setting	UNIQUETECEVENTS Setting	Baroc File
NO	NO	ibmtsm.baroc
NO	YES	itsmuniq.baroc
YES	Defaults to YES because UNIQUETDPTECEVENTS is set to YES.	itsmdpex.baroc

Each successive baroc file accepts the events of the previous baroc file. For example, itsmuniq.baroc accepts all events in ibmtsm.baroc, and itsmdpex.baroc accepts all events contained in itsmuniq.baroc.

Encoding events to UTF-8

Tivoli Storage Manager supports the following option to encode a Tivoli Enterprise Console event into UTF-8 before sending it to the Tivoli Enterprise Console server. Some Tivoli Enterprise Console patches (for example, Patch 0004 for Tivoli Enterprise Console Version 3.6 Modification 2) require UTF-8 encoded events for some locales.

Option Name	Function
TECUTF8EVENT	Encodes a Tivoli Enterprise Console event into UTF-8

To determine whether this option is enabled, issue the QUERY OPTION command.

Setting up a Tivoli Enterprise Console as a receiver

You can set up a Tivoli Enterprise Console as a receiver for event logging.

To set up Tivoli as a receiver for event logging, complete the following procedure:

1. Define the Tivoli Storage Manager event classes to the Tivoli Enterprise Console with the baroc file for your operating system:

ibmtsm.baroc

This file is distributed with the server.

Note: Please refer to Tivoli Enterprise Console documentation for instructions on removing an existing baroc file, if needed, and installing a new baroc file. Before the events are displayed on a Tivoli Enterprise Console, you must import the baroc file into an existing rule base or create a new rule base and activate it. To do this, complete the following steps:

- a. From the Tivoli desktop, click on the **Rule Base** icon to display the pop-up menu.
- b. Select **Import**, then specify the location of the baroc file.
- c. Select the **Compile** pop-up menu.
- d. Select the **Load** pop-up menu and **Load, but activate only when server restarts** from the resulting dialog.
- e. Shut down the event server and restart it.

To create a new rule base, complete the following steps:

- a. Click on the **Event Server** icon from the Tivoli desktop. The **Event Server Rules Bases** window will open.
 - b. Select **Rule Base** from the **Create** menu.
 - c. Optionally, copy the contents of an existing rule base into the new rule base by selecting the **Copy** pop-up menu from the rule base to be copied.
 - d. Click on the **RuleBase** icon to display the pop-up menu.
 - e. Select **Import** and specify the location of the baroc file.
 - f. Select the **Compile** pop-up menu.
 - g. Select the **Load** pop-up menu and **Load, but activate only when server restarts** from the resulting dialog.
 - h. Shut down the event server and restart it.
2. To define an event source and an event group:
 - a. From the Tivoli desktop, select **Source** from the **EventServer** pop-up menu. Define a new source whose name is Tivoli Storage Manager from the resulting dialog.
 - b. From the Tivoli desktop, select **Event Groups** from the **EventServer** pop-up menu. From the resulting dialog, define a new event group for Tivoli Storage Manager and a filter that includes event classes `IBMTSMSEVER_EVENT` and `IBMTSMCLIENT_EVENT`.
 - c. Select the **Assign Event Group** pop-up menu item from the **Event Console** icon and assign the new event group to the event console.
 - d. Double-click on the **Event Console** icon to start the configured event console.
 3. Enable events for logging to the Tivoli receiver. See “Enabling and disabling events” on page 627 for more information.
 4. In the server options file, specify the location of the host on which the Tivoli server is running. For example, to specify a Tivoli server at the IP address 9.114.22.345:1555, enter the following:

```
techost 9.114.22.345
tecport 1555
```
 5. Begin event logging for the Tivoli receiver. You do this in one of two ways:
 - To begin event logging automatically at server start up, specify the following server option:

```
tecbegineventlogging yes
```

Or
 - Enter the following command:

```
begin eventlogging tivoli
```

See “Beginning and ending event logging” on page 628 for more information.

Logging events to an SNMP manager

Tivoli Storage Manager supports the simple network management protocol (SNMP) together with event logging.

You can do the following:

- Set up an SNMP heartbeat monitor to regularly check that the Tivoli Storage Manager server is running.
- Send messages known as *traps* to an SNMP manager, such as NetView® or Tivoli Enterprise Console.
- Run Tivoli Storage Manager scripts and retrieve output and return codes. See “IBM Tivoli Storage Manager server scripts” on page 578 for details.

Tivoli Storage Manager also implements an SNMP subagent that can be configured to report exception conditions and provide support for a management information base (MIB). The management information base (MIB), which is shipped with Tivoli Storage Manager, defines the variables that will run server scripts and return the server scripts' results. You must register SNMPADMIN, the administrative client the server runs these scripts under. Although a password is not required for the subagent to communicate with the server and run scripts, a password should be defined for SNMPADMIN to prevent access to the server from unauthorized users. An SNMP password (community name) is required, however, to access the SNMP agent, which forwards the request to the subagent.

Note: Because the SNMP environment has weak security, you should consider not granting SNMPADMIN any administrative authority. This restricts SNMPADMIN to issuing only Tivoli Storage Manager queries.

SNMP SET requests are accepted for the name and input variables associated with the script names stored in the MIB by the SNMP subagent. This allows a script to be processed by running a GET request for the `ibmAdsm1ReturnValue` and `ibmAdsm2ReturnValue` variables. A GETNEXT request will not cause the script to run. Instead, the results of the previous script processed will be retrieved. When an entire table row is retrieved, the GETNEXT request is used. When an individual variable is retrieved, the GET request is used.

Here is a typical Tivoli Storage Manager configuration with SNMP:

1. Systems A, B, C: A Tivoli Storage Manager server communicates with a local subagent.
2. System D: A DPI-enabled SNMP agent is installed. This is required for communication between the Tivoli Storage Manager SNMP subagent, `dsmsnmp`, and the SNMP Manager you are using. A DPI-enabled SNMP agent is available as part of the AIX operating system.
3. System E: An SNMP manager, such as NetView, is installed.
4. The subagents on systems A, B, and C communicate with the agent on system D.
5. The agent on system D forwards SNMP traps to NetView on system E.

To run an arbitrary command from an SNMP management application, for example, NetView, follow these steps:

1. Choose the name and parameters for a Tivoli Storage Manager script.
2. Use the application to communicate with the SNMP agent. This agent changes the Tivoli Storage Manager MIB variable for one of the two script names that the Tivoli Storage Manager subagent maintains. The SNMP agent also sets the parameter variables for one of the two scripts.
3. Use the application to retrieve the variable *ibmAdsmReturnValue1.x* or *ibmAdsmReturnValue2.x*, where *x* is the index of the server that is registered with the subagent.

To set the variables associated with the script (for example, *ibmAdsmServerScript1/2* or *ibmAdsmM1Parm1/2/3*), the nodes on which the subagent and the agent are run must have read-write authority to the MIB variables. This is done through the SNMP configuration process on the system that the SNMP agent runs on.

Here is an AIX example:

```
community public 9.115.20.174 255.255.255.254 readWrite
community public 9.115.46.25 255.255.255.254 readWrite
community public 127.0.0.1 255.255.255.254 readWrite
community public 9.115.20.176 255.255.255.254 readWrite
smux 1.3.6.1.4.1.2.3.1.2.2.1.1.2 public
```

The statements grant read-write authority to the MIB for the local node through the loopback mechanism (127.0.0.1), and to nodes with the three 9.115.xx.xx addresses. The smux statement allows the dpid2 daemon to communicate with snmpd.

Here is an example of this command used to set and retrieve MIB variables:

```
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmServerScript1.1=QuerySessions
```

This command issues the set operation (-ms), passing in community name **public**, sending the command to host **tpcnov73**, and setting up variable *ibmAdsmServerScript1* to have the value *QuerySessions*. *QuerySessions* is the name of a server script that has been defined on a server that will register with the Tivoli Storage Manager subagent. In this case, the first server that registers with the subagent is the .1 suffix in *ibmAdsmServerScript1.1*. The following commands set the parameters for use with this script:

```
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm1.1=xyz
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm2.1=uvw
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm3.1=xxx
```

You can set zero to three parameters. Only the script name is needed. To make the *QuerySessions* script run, retrieve the *ibmAdsmM1ReturnValue* variable (in this case, *ibmAdsmM1ReturnValue.1*). For example:

```
snmpinfo -v -mg -c public -h tpcnov73 ibmAdsmM1ReturnValue.1
```

The results of the command are returned as a single string with embedded carriage return/newline characters.

Note: Not all MIB browsers properly handle embedded carriage return/newline characters.

In this case, *ibmAdsmM1ReturnCode.1* will contain the return code associated with the running of the script. If *ibmAdsmM2ReturnValue* is retrieved, the results of running the script named in *ibmAdsmServerScript2* are returned as a single numeric return code. Notice the -mg instead of -ms to signify the GET operation in the command to retrieve *ibmAdsmM1ReturnValue.1*. If the entire row is retrieved, the

command is not run. Instead, the results from the last time the script was run are retrieved. This would be the case if the following command were issued:

```
snmpinfo -v -md -c public -h tpcnov73 ibmAdsm
```

in which all Tivoli Storage Manager MIB variables are displayed.

An SNMP agent is needed for communication between an SNMP manager and its managed systems. The SNMP agent is realized through the **snmpd daemon**. The Distributed Protocol Interface (DPI[®]) Version 2 is an extension of this SNMP agent.

SNMP managers can use the MIB that is shipped with Tivoli Storage Manager to manage the server. Therefore, an SNMP agent supporting DPI Version 2 must be used to communicate with the Tivoli Storage Manager subagent. This SNMP agent is not included with Tivoli Storage Manager. A supported DPI agent ships with AIX. The Tivoli Storage Manager subagent is included with Tivoli Storage Manager and, before server startup, must be started as a separate process communicating with the DPI-enabled SNMP agent.

Note: The `snmpd` daemon on Linux does not support DPI. Therefore, the subagent must be configured to communicate with a DPI-enabled SNMP agent that is running on a Windows or AIX system. Alternatively, you can configure the server to communicate with an existing Tivoli Storage Manager subagent. The Tivoli Storage Manager subagent, in turn, communicates with the DPI-enabled SNMP agent.

The SNMP manager system can reside on the same system as the Tivoli Storage Manager server, but typically would be on another system connected through SNMP. The SNMP management tool can be any application, such as NetView or Tivoli Enterprise Console, which can manage information through SNMP MIB monitoring and traps. The Tivoli Storage Manager server system runs the processes needed to send Tivoli Storage Manager event information to an SNMP management system. The processes are:

- SNMP agent (`snmpd`)
- Tivoli Storage Manager SNMP subagent (`dsmsnmp`)
- Tivoli Storage Manager server (`dsmserv`)

Figure 79 illustrates a typical Tivoli Storage Manager implementation:

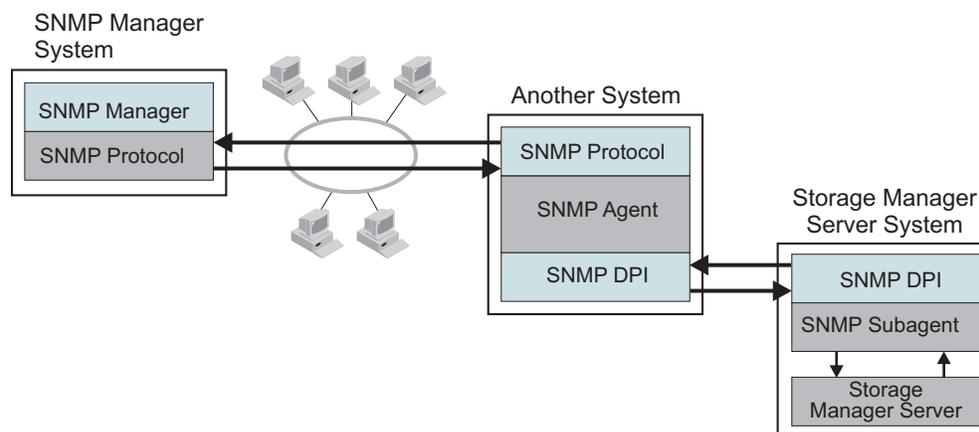


Figure 79. Tivoli Storage Manager SNMP Implementation

Figure 80 shows how the communication for SNMP works in a Tivoli Storage Manager system:

- The SNMP manager and agent communicate with each other through the SNMP protocol. The SNMP manager passes all requests for variables to the agent.
- The agent then passes the request to the subagent and sends the answer back to the manager. The agent responds to the manager's requests and informs the manager about events by sending traps.
- The agent communicates with both the manager and subagent. It sends queries to the subagent and receives traps that inform the SNMP manager about events taking place on the application monitored through the subagent. The SNMP agent and subagent communicate through the Distributed Protocol Interface (DPI). Communication takes place over a stream connection, which typically is a TCP connection but could be another stream-connected transport mechanism.
- The subagent answers MIB queries of the agent and informs the agent about events by sending traps. The subagent can also create and delete objects or subtrees in the agent's MIB. This allows the subagent to define to the agent all the information needed to monitor the managed application.

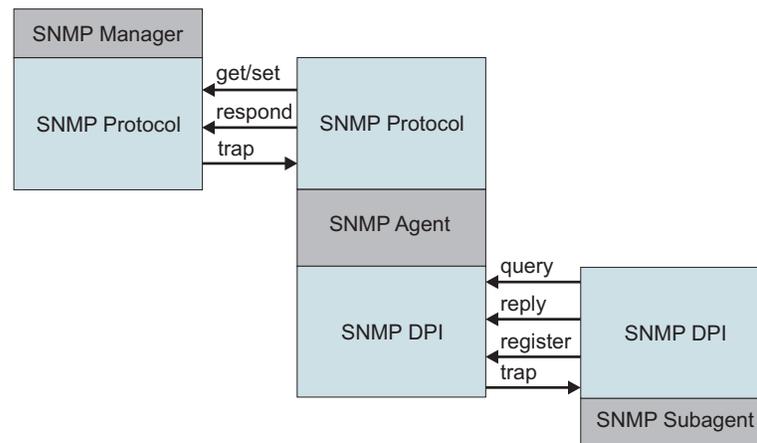


Figure 80. Manager-Agent-Subagent communication

Note:

1. You can start *dsmsnmp* and the server in any order. However, starting *dsmsnmp* first is more efficient in that it avoids retries.
2. The MIB file name is *admserv.mib*. The file name is located in the directory in which the server is installed.
3. Merge the contents of the *admserv.mib* file into the */etc/mib.defs* file.

Configuring IBM Tivoli Storage Manager SNMP

You can configure SNMP by completing the following procedure.

The IBM Tivoli Storage Manager SNMP set up procedure is illustrated by Figure 81 on page 638:

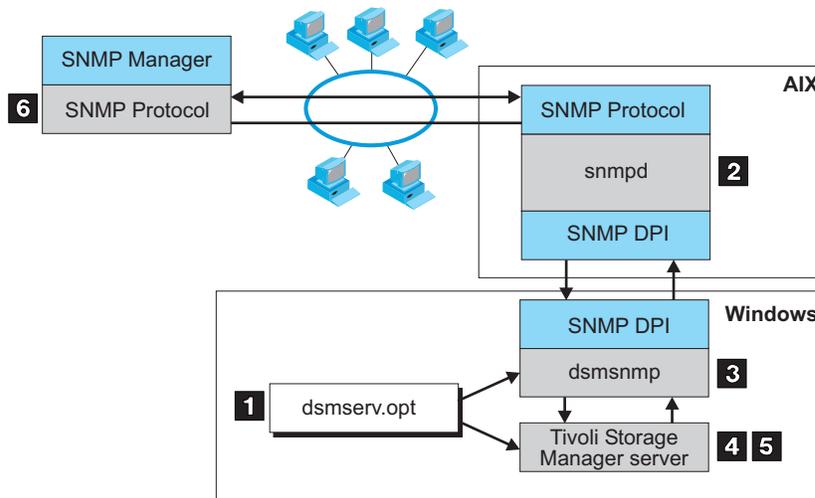


Figure 81. IBM Tivoli Storage Manager SNMP Set Up

To set up Tivoli Storage Manager monitoring through SNMP, do the following:

1. Modify the server options file to specify the SNMP communication method. Figure 82 displays an example of a SNMP communication method setting in the server options file.

You must specify the COMMMETHOD and SNMPSUBAGENT options. The SNMPSUBAGENT option must specify a host that is an AIX system with a DPI-enabled SNMP agent, such as the SystemView® agent.

```

commmethod          snmp
snmpsubagent       hostname jimbo communityname public timeout 600
snmpsubagentport   1521
snmpheartbeatinterval 5
snmpmessagecategory severity

```

Figure 82. Example of SNMP communication method options

For details about server options, see the server options section in *Administrator's Reference*.

2. Install, configure, and start the SNMP agent as described in the documentation for that agent. The SNMP agent must support the DPI Version 2.0 standard. Tivoli Storage Manager supports the SNMP agent that is built into the AIX operating system.

Before starting the agent, ensure that the **dpid2** and **snmpd** subsystems have been started.

Important: The dpid2 subsystem is started automatically through the snmpd subsystem in SNMP, V3. The dpid2 subsystem must be manually started independently of the snmpd subsystem with SNMP, V1.

3. Start the Tivoli Storage Manager SNMP subagent by running the dsmsnmp executable.
4. Start the Tivoli Storage Manager server to begin communication through the configured TCP/IP port with the subagent.
5. Begin event logging for the SNMP receiver, and enable events to be reported to SNMP. For example, issue the following commands:

```

begin eventlogging snmp
enable event snmp all

```

- Define the Tivoli Storage Manager SNMP MIB values for the SNMP manager to help format and display the Tivoli Storage Manager SNMP MIB variables and messages. The *adsmsev.mib* file ships with the Tivoli Storage Manager server and must be loaded by the SNMP manager. This file is in the installation directory of the server. For example, when you run NetView for Windows as an SNMP manager, the *adsmsev.mib* file is copied to the *\netview_path\SNMP_MIB* directory and then loaded through the following command:

```
[C:\] loadmib -load adsmsev.mib
```

Enterprise event logging: logging events to another server

One or more servers can send server events and events from their own clients to another server for logging.

The sending server receives the enabled events and routes them to a designated event server. This is done by a receiver that IBM Tivoli Storage Manager provides. At the event server, an administrator can enable one or more receivers for the events being routed from other servers. Figure 83 shows the relationship of a sending Tivoli Storage Manager server and a Tivoli Storage Manager event server.

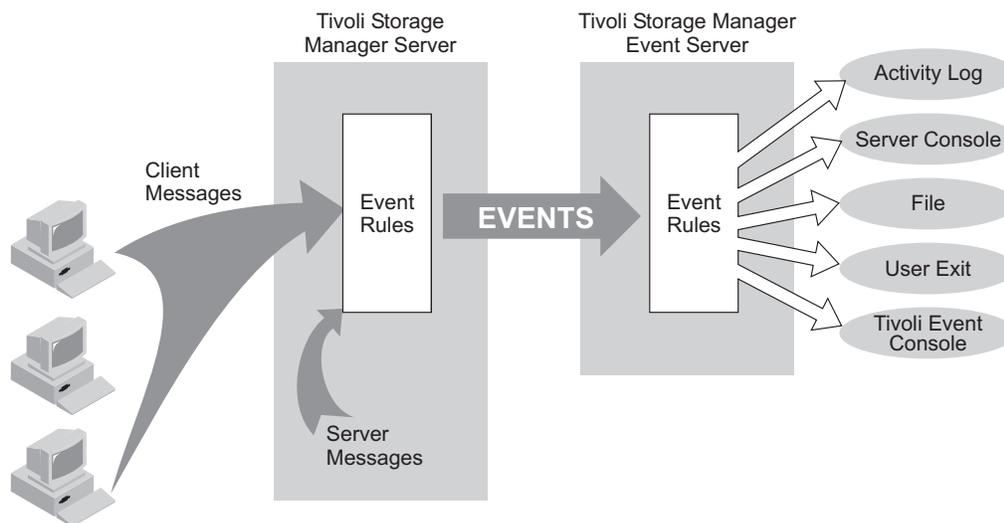


Figure 83. Server-to-server event logging

The following scenario is a simple example of how enterprise event logging can work.

The administrator at each sending server does the following:

- Defines the server that will be the event server. For details about communication set up, see “Setting up communications for enterprise configuration and enterprise event logging” on page 676.


```
define server server_b password=cholla h1address=9.115.3.45 l1address=1505
```
- Identifies the server just defined as the event server:


```
define eventserver server_b
```
- Enables the logging of severe, error, and warning server messages from the sending server and severe and error messages from all clients to the event server receiver by issuing the following commands:


```
enable events eventserver severe,error,warning
enable events eventserver severe,error nodename=*
```
- Begins event logging by issuing the following command:

```
begin eventlogging eventserver
```

The administrator at the event server does the following:

5.

Enables the logging of severe and error messages to a file named `events` that are sent to it from the sending servers. The administrator defines the file with the following option in the server options file:

```
fileexit yes events append
```

Then the administrator enables the events by issuing the `ENABLE EVENTS` command for each sending server. For example, for `SERVER_A` the administrator would enter:

```
enable events file severe,error servername=server_a
```

Note: By default, logging of events from another server is enabled to the event server activity log. However, unlike events originating from a local server, events originating from another server can be disabled for the activity log at an event server.

One or more servers can send events to an event server. An administrator at the event server enables the logging of specific events from specific servers. In the previous example, `SERVER_A` routes severe, error, and warning messages to `SERVER_B`. `SERVER_B`, however, logs only the severe and error messages. If a third server sends events to `SERVER_B`, logging is enabled only if an `ENABLE EVENTS` command includes the third server. Furthermore, the `SERVER_B` determines the receiver to which the events are logged.

Attention: It is important that you do not set up server-to-server event logging in a loop. In such a situation, an event would continue logging indefinitely, tying up network and memory resources. Tivoli Storage Manager will detect such a situation and issue a message. Here are a few configurations to avoid:

- `SERVER_A` logs to `SERVER_B`, and `SERVER_B` logs to `SERVER_A`.
- `SERVER_A` logs to `SERVER_B`; `SERVER_B` logs to `SERVER_C`; `SERVER_C` logs to `SERVER_A`.

Querying event logging

You can use the `QUERY ENABLED` command to display a list of server or client events that are enabled or disabled by a specified receiver.

Because the lists of enabled and disabled events could be very long, Tivoli Storage Manager displays the shorter of the two lists.

For example, assume that 1000 events for client node `HSTANFORD` were enabled for logging to the user exit and that later two events were disabled. To query the enabled events for `HSTANFORD`, you can enter:

```
query enabled userexit nodename=hstanford
```

The output would specify the *number* of enabled events and the *message names* of disabled events:

```
998 events are enabled for node HSTANFORD for the USEREXIT receiver.
```

```
The following events are DISABLED for the node HSTANFORD for the USEREXIT
```

```
receiver:
```

```
ANE4000, ANE49999
```

The QUERY EVENTRULES command displays the history of events that are enabled or disabled by a specific receiver for the server or for a client node.

```
query enabled userexit nodename=hstanford
```

User exit and file exit receivers

The data structure of the user exit receivers applies to the file exit receivers. To use one of these exits with Tivoli Storage Manager, you must specify the corresponding server option (FILEEXIT, FILETEXTEXIT, or USEREXIT) in the server options file.

The samples for the C, H, and make files are shipped with the server code in the /opt/tivoli/tsm/server/bin directory.

Attention:

1. Use caution in modifying these exits. A user exit abend will bring down the server.
2. The file specified in the file exit option will continue to grow unless you prune it.

You can also use Tivoli Storage Manager commands to control event logging. For details, see “Logging IBM Tivoli Storage Manager events to receivers” on page 626 and *Administrator’s Reference*.

Sample user exit declarations

userExitSample.h contains declarations for a user-exit program.

The environment is:

```
Linux/i386
```

Figure 84. Sample user exit declarations

```
/******  
 * Name:          userExitSample.h  
 * Description:   Declarations for a user exit  
******/  
  
#ifndef _H_USEREXITSAMPLE  
#define _H_USEREXITSAMPLE  
  
#include <stdio.h>  
#include <sys/types.h>  
  
/**** Do not modify below this line. ****/  
  
#define BASE_YEAR      1900  
  
typedef short  int16;  
typedef int    int32;  
  
/* uchar is usually defined in <sys/types.h> */  
/* DateTime Structure Definitions - TSM representation of a timestamp*/  
  
typedef struct  
{  
    uchar  year; /* Years since BASE_YEAR (0-255) */  
    uchar  mon;  /* Month (1 - 12) */  
    uchar  day;  /* Day (1 - 31) */  
    uchar  hour; /* Hour (0 - 23) */  
    uchar  min;  /* Minutes (0 - 59) */  
}
```

```

    uchar sec;    /* Seconds (0 - 59) */
} DateTime;

/*****
 * Some field size definitions (in bytes) *
 *****/

#define MAX_SERVERNAME_LENGTH 64
#define MAX_NODE_LENGTH 64
#define MAX_COMMNAME_LENGTH 16
#define MAX_OWNER_LENGTH 64
#define MAX_HL_ADDRESS 64
#define MAX_LL_ADDRESS 32
#define MAX_SCHED_LENGTH 30
#define MAX_DOMAIN_LENGTH 30
#define MAX_MSGTEXT_LENGTH 1600

/*****
 * Event Types (in e1EventRecvData.eventType) *
 *****/

#define TSM_SERVER_EVENT      0x03 /* Server Events */
#define TSM_CLIENT_EVENT     0x05 /* Client Events */

/*****
 * Application Types (in e1EventRecvData.applType) *
 *****/

#define TSM_APPL_BACKARCH    1 /* Backup or Archive client */
#define TSM_APPL_HSM        2 /* Space manage client */
#define TSM_APPL_API        3 /* API client */
#define TSM_APPL_SERVER     4 /* Server (ie. server to server) */

/*****
 * Event Severity Codes (in e1EventRecvData.sevCode) *
 *****/

#define TSM_SEV_INFO        0x02 /* Informational message. */
#define TSM_SEV_WARNING    0x03 /* Warning message. */
/*
#define TSM_SEV_ERROR      0x04 /* Error message. */
#define TSM_SEV_SEVERE    0x05 /* Severe error message. */
#define TSM_SEV_DIAGNOSTIC 0x06 /* Diagnostic message. */
#define TSM_SEV_TEXT      0x07 /* Text message. */

/*****
 * Data Structure of Event that is passed to the User-Exit. *
 * This data structure is the same for a file generated using *
 * the FILEEXIT option on the server. *
 *****/

typedef struct evRdata
{
    int32 eventNum; /* the event number. */
    int16 sevCode; /* event severity. */
    int16 applType; /* application type (hsm, api, etc)*/
    int32 sessId; /* session number */
    int32 version; /* Version number of this structure (1)*/
    int32 eventType; /* event type
                     * (TSM_CLIENT_EVENT, TSM_SERVER_EVENT)*/
    DateTime timeStamp; /* timestamp for event data. */
    uchar serverName[MAX_SERVERNAME_LENGTH+1]; /* server name */
    uchar nodeName[MAX_NODE_LENGTH+1]; /* Node name for session */
    uchar commMethod[MAX_COMMNAME_LENGTH+1]; /* communication method */
    uchar ownerName[MAX_OWNER_LENGTH+1]; /* owner */
    uchar hlAddress[MAX_HL_ADDRESS+1]; /* high-level address */
    uchar llAddress[MAX_LL_ADDRESS+1]; /* low-level address */

```

```

    uchar    schedName[MAX_SCHED_LENGTH+1]; /* schedule name if applicable*/
    uchar    domainName[MAX_DOMAIN_LENGTH+1]; /* domain name for node */
    uchar    event[MAX_MSGTEXT_LENGTH];      /* event text */
} elEventRecvData;

/*****
 * Size of the Event data structure *
 *****/

#define ELEVENTRECVDATA_SIZE      sizeof(elEventRecvData)

/*****
 * User Exit EventNumber for Exiting *
 *****/

#define USEREXIT_END_EVENTNUM    1822 /* Only user-exit receiver to exit*/
#define END_ALL_RECEIVER_EVENTNUM 1823 /* All receivers told to exit */

/*****
 *** Do not modify above this line. ***
 *****/

/***** Additional Declarations *****/

#endif

```

Sample user-exit program

userExitSample.c is a sample user-exit program invoked by the server.

Figure 85. Sample user exit program

```

/*****
 * Name:          userExitSample.c
 * Description:   Example user-exit program invoked by the server
 * Environment:   Linux/i386
 *****/

#include <stdio.h>
#include "userExitSample.h"

/*****
 *** Do not modify below this line. ***
 *****/

extern void adsmV3UserExit( void *anEvent );

/*****
 *** Main ***
 *****/

int main(int argc, char *argv[])
{
/* Do nothing, main() is never invoked, but stub is needed */

exit(0); /* For picky compilers */

} /* End of main() */

/*****
 * Procedure: adsmV3UserExit
 * If the user-exit is specified on the server, a valid and
 * appropriate event causes an elEventRecvData structure (see
 * userExitSample.h) to be passed to adsmV3UserExit that returns a void.
 * INPUT : A (void *) to the elEventRecvData structure
 *****/

```

```

* RETURNS: Nothing
*****/

void adsmV3UserExit( void *anEvent )
{
/* Typecast the event data passed */
elEventRecvData *eventData = (elEventRecvData *)anEvent;

/*****
*** Do not modify above this line. ***
*****/

if( ( eventData->eventNum == USEREXIT_END_EVENTNUM ) ||
    ( eventData->eventNum == END_ALL_RECEIVER_EVENTNUM ) )
{
/* Server says to end this user-exit. Perform any cleanup, *
 * but do NOT exit() !!!                                     */
return;
}

/* Field Access: eventData->... */
/* Your code here ... */

/* Be aware that certain function calls are process-wide and can cause
 * synchronization of all threads running under the TSM Server process!
 * Among these is the system() function call. Use of this call can
 * cause the server process to hang and otherwise affect performance.
 * Also avoid any functions that are not thread-safe. Consult your
 * system's programming reference material for more information.
 */

return; /* For picky compilers */
} /* End of adsmV3UserExit() */

```

Readable text file exit (FILETEXTEXIT) format

If you specify the readable text file exit (FILETEXTEXIT), each logged event is written to a fixed-size, readable line.

The following table presents the format of the output. Fields are separated by blank spaces.

Table 55. Readable text file exit (FILETEXTEXIT) format

Column	Description
0001-0006	Event number (with leading zeros)
0008-0010	Severity code number
0012-0013	Application type number
0015-0023	Session ID number
0025-0027	Event structure version number
0029-0031	Event type number
0033-0046	Date/Time (YYYYMMDDHHmmSS)
0048-0111	Server name (right padded with spaces)
0113-0176 ¹	Node name
0178-0193 ¹	Communications method name
0195-0258 ¹	Owner name
0260-0323 ¹	High-level internet address (n.n.n.n)
0325-0356 ¹	Port number from high-level internet address

Table 55. Readable text file exit (FILETEXTEXIT) format (continued)

Column	Description
0358-0387 ¹	Schedule name
0389-0418 ¹	Domain name
0420-2019	Event text
2020-2499	Unused spaces
2500	New line character

¹ Columns 113 - 418 contain data only for events that originate in a client or in another Tivoli Storage Manager server. Otherwise, columns 113 - 418 contain blanks.

Monitoring IBM Tivoli Storage Manager accounting records

Tivoli Storage Manager accounting records show the server resources that are used during a session. This information lets you track resources that are used by a client node session.

Task	Required Privilege Class
Set accounting records on or off	System

At installation, accounting defaults to OFF. You can turn accounting on by using the SET ACCOUNTING command. When accounting is set to ON, the server creates a session resource usage accounting record whenever a client node session ends.

Accounting records are stored in the *dsmacct.log* file. The DSMSERV_ACCOUNTING_DIR environment variable specifies the directory where the accounting file is opened. If this variable is not set when the server is started, the *dsmacct.log* file is placed in the current directory when the server starts. For example, to set the environment variable to place the accounting records in the */home/engineering* directory, enter this command:

```
export DSMSERV_ACCOUNTING_DIR=/home/engineering
```

The accounting file contains text records that can be viewed directly or can be read into a spreadsheet program. The file remains opened while the server is running and accounting is set to ON. The file continues to grow until you delete it or prune old records from it. To close the file for pruning, either temporarily set accounting off or stop the server.

There are 31 fields, which are delimited by commas (,). Each record ends with a new-line character. Each record contains the following information:

Field	Contents
1	Product version
2	Product sublevel
3	Product name, 'ADSM',
4	Date of accounting (mm/dd/yyyy)
5	Time of accounting (hh:mm:ss)
6	Node name of Tivoli Storage Manager client
7	Client owner name (UNIX)
8	Client Platform
9	Authentication method used

Field	Contents
10	Communication method used for the session
11	Normal server termination indicator (Normal=X'01', Abnormal=X'00')
12	Number of archive store transactions requested during the session
13	Amount of archived files, in kilobytes, sent by the client to the server
14	Number of archive retrieve transactions requested during the session
15	Amount of space, in kilobytes, retrieved by archived objects
16	Number of backup store transactions requested during the session
17	Amount of backup files, in kilobytes, sent by the client to the server
18	Number of backup retrieve transactions requested during the session
19	Amount of space, in kilobytes, retrieved by backed up objects
20	Amount of data, in kilobytes, communicated between the client node and the server during the session
21	Duration of the session, in seconds
22	Amount of idle wait time during the session, in seconds
23	Amount of communications wait time during the session, in seconds
24	Amount of media wait time during the session, in seconds
25	Client session type. A value of 1 or 4 indicates a general client session. A value of 5 indicates a client session that is running a schedule. Values other than 1, 4, or 5 are reserved for the Tivoli Storage Manager server's internal use, and you can ignore them.
26	Number of space-managed store transactions requested during the session
27	Amount of space-managed data, in kilobytes, sent by the client to the server
28	Number of space-managed retrieve transactions requested during the session
29	Amount of space, in kilobytes, retrieved by space-managed objects
30	Product release
31	Product level

The following shows a sample record:

```
3,8,ADSM,08/03/2000,16:26:37,node1,,AIX,1,Tcp/Ip,0,254,1713,0,0,47,1476,0,0,3316,
960,27,5,1,4,0,0,0,0,7,2
```

Daily monitoring scenario

Depending on the configuration of your system, you can add monitoring tasks to the scripts you run daily. If a function does not complete properly, you can review the activity log for errors that occurred around the time of failure.

You can include the commands shown in a command script that you can run daily. Review the output of the script for any errors or problems. See “Requesting information from the activity log” on page 624 for details.

1. Verify that drives are online. If there is a drive in the unavailable state, there may be errors with schedules.
query drive
2. Check the status of disk volumes. If any are offline, check for hardware problems.
query volume devclass=disk
3. Check that scratch volumes are available.
query libvolume
4. Check the access state of the tape volumes. For example, a volume that is not in the read-write state may indicate a problem.
You may need to move data and check the volumes out of the library.
query volume

5. Check database and recovery log statistics.
query db
query log
6. Verify that scheduled database backups completed successfully.
query volhistory type=dbbackup
7. Check the activity log for error messages.
query actlog search=ANR????E

Reporting and monitoring servers and clients

The IBM Tivoli Storage Manager reporting and monitoring feature uses a combination of reporting and monitoring components to offer you historical reports and real-time monitoring information for the IBM Tivoli Storage Manager servers and clients.

You can view the historical reports to see if there are any issues that need attention, such as uncontrolled growth over time. You can also monitor the Tivoli Storage Manager server status, database size, agent status, client node status, scheduled events, server IDs and so on by monitoring workspaces.

The reporting component, sometimes referred to as Tivoli Common Reporting, reports on the retrieved historical data. IBM Tivoli Monitoring acts as a monitoring application that provides workspaces for you to monitor real-time information.

The reporting and monitoring agent communicates with the reporting and monitoring server to retrieve data from its database and return this data to the Tivoli Monitoring server.

The IBM Tivoli Monitoring server stores this data in the Tivoli Data Warehouse. Figure 86 on page 649 describes this process.

The reporting and monitoring feature uses the following components:

IBM Tivoli Monitoring

Consists of a number of components that accumulate and monitor historical data for reporting:

- IBM Tivoli Enterprise Portal
- IBM Tivoli Enterprise Management Server
- Tivoli Data Warehouse

IBM DB2

Stores historical data that is obtained from Tivoli Storage Manager servers that are monitored using IBM Tivoli Monitoring.

Tivoli Storage Manager reporting and monitoring agent

Queries and formats data to be presented to you in the following ways:

- As workspaces using the Tivoli Enterprise Portal
- As reports using the Tivoli Data Warehouse and the reporting portion of the Tivoli Storage Manager reporting and monitoring feature

The reporting and monitoring agent is installed on the Tivoli Storage Manager server or the IBM Tivoli Monitoring server, and is a multi-instance data collection agent.

Tivoli Storage Manager reporting and monitoring reporting infrastructure

Reports on the Tivoli Storage Manager server activities from data that is

collected using the Tivoli Storage Manager monitoring agent. The monitoring feature uses the Tivoli Enterprise Portal to view the current status of the Tivoli Storage Manager server.

Figure 86 on page 649 shows how the data flows between the different components.

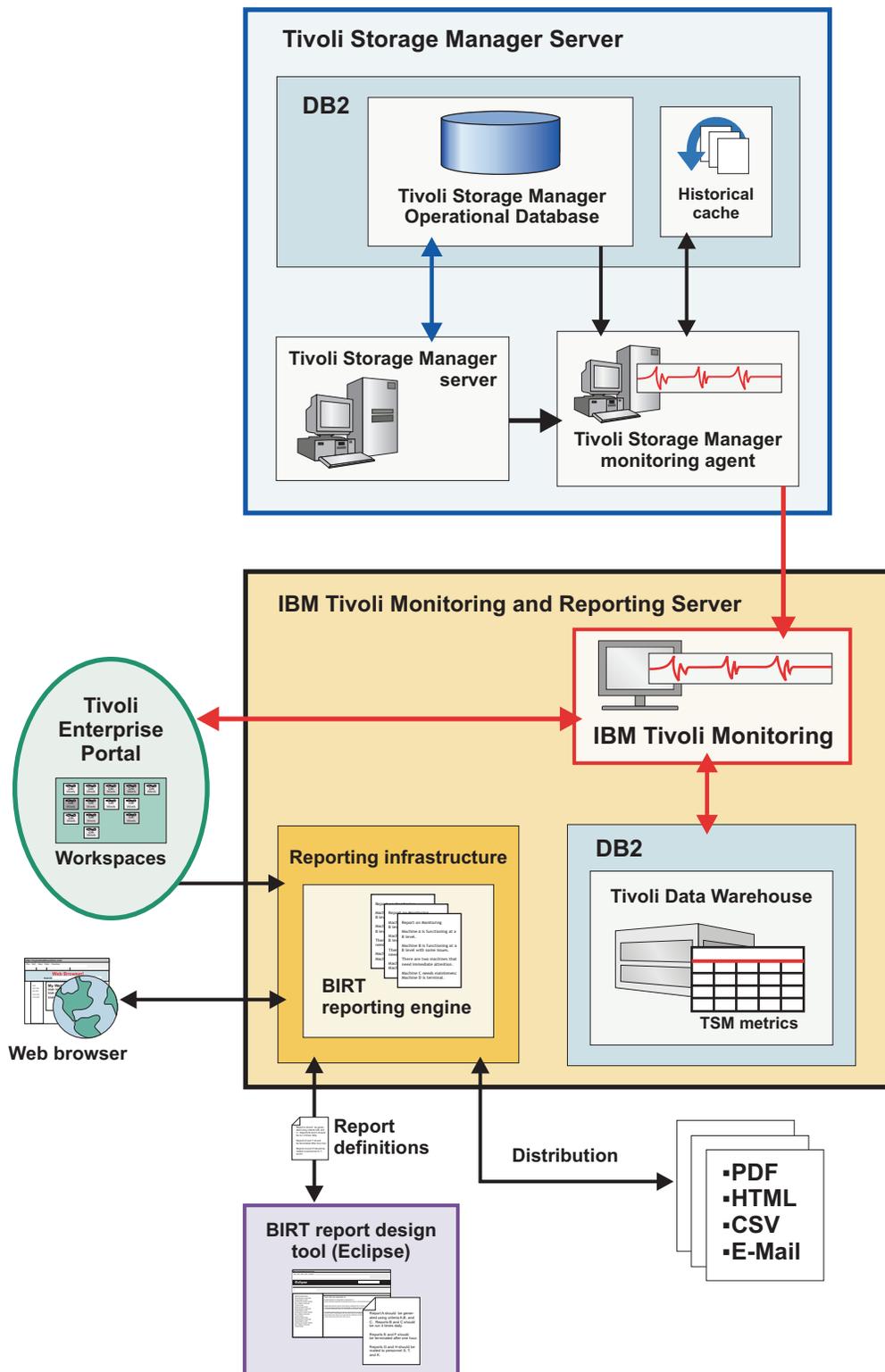


Figure 86. Tivoli Storage Manager reporting and monitoring feature infrastructure and data flow

If you are interested in creating your own custom reports, you are required to install the Business Intelligence and Reporting Tools (BIRT) software. The Tivoli Storage Manager reporting and monitoring feature does not have an option to create custom reports using Tivoli Storage Manager. See the section on installing

software for custom reporting in the *IBM Tivoli Storage Manager Installation Guide, v6.1* for details on how to create custom reports.

Client activity reports

Client activity reports include information about your client activity as well as schedule status, filespace information, backup, and other detailed activity history for your Tivoli Storage Manager clients.

These reports are generated by the Tivoli Storage Manager reporting and monitoring agent and are available in HTML, PDF, PostScript®, and Microsoft Excel format.

Depending on the type of report you want to run, and the parameters available for that report, you can choose the parameters in the On-Demand Report Parameters window to customize how the data is displayed in the reports. Table 56 describes these parameters.

Table 57 on page 651 lists the client report titles and their descriptions.

Table 56. Reporting parameters

Parameter	Description
Activity type	This parameter is used to select the following different client activities: <ul style="list-style-type: none"> • Backup (incremental only) • Archive • Restore • Retrieve
Report period	This parameter is used to select one the following date ranges to display <ul style="list-style-type: none"> • All • Today • Yesterday • The last 24 hours • The last 7 days • The last 30 days • The last 90 days • The last 365 days • The current week • The last month • The last 3 months • Year to date
Start date and end date	This parameter is used to overwrite the report period by choosing a start date and an end date.
Server name	This parameter is used to select which server to report on.
Client node name	This parameter is used to supply a client from the server or a wild card (% or A%) to report on.
Summarization type	This parameter is used to select how to group or summarize the data by either daily (default), hourly, weekly, monthly, quarterly, or yearly.
Number of clients to display	This parameter displays the number of top clients you want to see in the report.

Table 57. Client reports

Report name	Description
Client activity details	<p>This report provides detail about client activities that include backup (incremental only), archive, restore, and retrieve. It does not include information on image or network data management protocol (NDMP) backups.</p> <p>It provides the following data on client activity:</p> <p>Client node name The client node's name on the Tivoli Storage Manager server.</p> <p>Duration The total time to complete the activity</p> <p>Date The date the activity was run.</p> <p>Objects evaluated The number of objects that were evaluated by the activity.</p> <p>Objects processed The number of objects that were processed by the activity</p> <p>Objects failed The number of objects that failed to process by the activity. For example, files missed for a backup or archive.</p> <p>Bytes Moved The total number of bytes that were moved by the activity.</p>
Client activity history	<p>This report provides detail of the client activity, backup (incremental only), archive, restore, and retrieve, from all clients over periods of time. It does not include information on image or NDMP backups .</p> <p>It provides the following line charts for the selected client activity on a selected time frame:</p> <p>Client nodes participating Produces a line chart with the total nodes participating in the activity for a given time frame. A client node participating is a node that has done a backup within the user-selected time frame.</p> <p>Amount of data Produces a line chart with the total amount of bytes that were moved for the selected activity for a given time frame.</p> <p>Objects processed Produces a line chart with the total objects that were processed for the selected activity for a given time frame.</p>

Table 57. Client reports (continued)

Report name	Description
Client backup currency	<p>This report provides you with a list of all client nodes that were identified to the server and includes when they were last backed up successfully, or with warnings.</p> <p>It displays only scheduled backups and does not display manual backups. If a node runs manual backups daily, this report shows that the node has never run a backup.</p> <p>It provides the following entities:</p> <ul style="list-style-type: none"> • A pie chart that displays the following data: <ul style="list-style-type: none"> – Number of clients whose schedule was backed up within 24 hours – Number of clients whose schedule was backed up within 48 hours – Number of clients whose schedule was backed up within one week – Number of clients whose schedule was backed up within one month – Number of clients whose schedule was backed up within a month – Number of clients whose schedule was backed up over a month – Number of clients whose schedule has no backup date • A table with the following data: <ul style="list-style-type: none"> Client node name The node that the backup was performed on Server name The client's server Last backup date The last time the backup was run Backup had warning messages Did the backup run successfully but produce warning messages
Client backup missed files	<p>This operational report identifies the actual files that have been missed in backups. It produces a table with the following data:</p> <ul style="list-style-type: none"> When The date that the file was missed Filename The full path and filename Reason code The error code number Reason The error message

Table 57. Client reports (continued)

Report name	Description
Client schedule status	<p>This report provides data about the activity for clients; jobs that have been run, jobs that have failed, jobs that ended with warnings.</p> <ul style="list-style-type: none"> • A schedule status of <i>success</i> is a schedule that has a return code of 0 and no warning messages associated with it • A schedule status of <i>success with warnings</i> is a schedule that has a return code of 0, but has warnings messages associated with it <p>This report provides the following entities:</p> <ul style="list-style-type: none"> • A bar chart that displays the following statuses for your selected time frame: <ul style="list-style-type: none"> – Successful – Successful with warnings – Missed – Failed schedules • The following status tables: <ul style="list-style-type: none"> – Successful – Successful with warnings – Failed – Missed • A table with the following data: <ul style="list-style-type: none"> Client node name The client node running the schedule Policy domain The policy domain that the node belongs to Schedule start When the schedule is suppose to start Actual start When the schedule started Schedule name The name of the schedule Failure Failure messages

Table 57. Client reports (continued)

Report name	Description
Client storage summary	<p>This report provides information about the filespaces that are being protected by this server, the quantity of data in them, by client, and as a summary of this information over time.</p> <p>It provides the following entities:</p> <ul style="list-style-type: none"> • A line chart that shows the maximum, average, and minimum bytes moved over a selected time frame. • A table with the following data: <p>Summary date The date the data was generated.</p> <p>Client node count The total nodes that are computed on the selected date maximum bytes.</p> <p>Maximum bytes The maximum bytes that are used by the nodes.</p> <p>Average bytes The average bytes that are used by the nodes.</p> <p>Minimum bytes The minimum bytes that are used by the nodes.</p>
Client Top Activity	<p>This report provides the number of users who run the most backups (incremental only), archives, restores, or retrieves on the Tivoli Storage Manager server. It does not include information on image or NDMP backups.</p> <p>It provides the following data:</p> <p>Client node name The client node's name on the Tivoli Storage Manager server.</p> <p>Server The server that the node exists on.</p> <p>Date The date the activity was run.</p> <p>Duration The total time the activity required to complete.</p> <p>Bytes moved The total number of bytes moved by the activity.</p>

Related tasks

“Running the Tivoli Storage Manager client and server reports” on page 667

Server trend reports

Server trend reports include historical information about your Tivoli Storage Manager server trends, including server throughput, resource usage, database details, and tape usage and analysis.

These reports are generated by the Tivoli Storage Manager reporting and monitoring agent and are available in HTML, PDF, PostScript, and Microsoft Excel format.

Depending on the type of report you want to run, and the parameters available for that report, you can choose the parameters in the On-Demand Report Parameters window to customize how the data is displayed in the reports. Table 58 describes these parameters.

Table 59 on page 656 lists the server report titles and their descriptions.

Table 58. Reporting parameters

Parameter	Description
Activity type	This parameter is used to select the following server activity: <ul style="list-style-type: none"> • Database backup
Report period	This parameter is used to select one of the following date ranges to display <ul style="list-style-type: none"> • All • Today • Yesterday • The last 24 hours • The last 7 days • The last 30 days • The last 90 days • The last 365 days • The current week • The last month • The last 3 months • Year to date
Start date and end date	This parameter is used to overwrite the report period by choosing a start date and an end date.
Server name	This parameter is used to select which server to report on.
Summarization type	This parameter is used to select how to group or summarize the data by either daily (default), hourly, weekly, monthly, quarterly, or yearly.

Table 59. Server reports

Report Name	Description
Server activity details	<p>This report provides detail about server activities that include backup (incremental only), archive, restore, and retrieve. It does not include information on image or network data management protocol (NDMP) backups.</p> <p>It provides the following data on server activity:</p> <p>Duration The total time to complete the activity</p> <p>Date The date the activity was run.</p> <p>Objects evaluated The number of objects that were evaluated by the activity.</p> <p>Objects processed The number of objects that were processed by the activity</p> <p>Objects failed The number of objects that failed to process by the activity. For example, files missed for a backup or archive.</p> <p>Bytes Moved The total number of bytes that were moved by the activity.</p> <p>Note: Time differences can occur because the database is rounded to the nearest minute while the time is calculated to a decimal minute.</p>
Server database details	<p>This report provides details of the database size and condition, some information that is stored in other places can be duplicated but it is designed to bring everything together in one place</p> <p>This report provides the following entities:</p> <p>A stacked bar chart Displays the available database space versus in-use database space in GB.</p> <p>A table Displays the database backup status. It has the following data:</p> <p>Date The date the database backup was run.</p> <p>Duration The time it took to complete the database backup. Note: All times are rounded to the nearest minute.</p> <p>Volumes used The total number of backup volumes that were used.</p> <p>Status The error messages that were received.</p> <p>Note: Time differences can occur because the database is rounded to the nearest minute while the time is calculated to a decimal minute.</p>

Table 59. Server reports (continued)

Report Name	Description
Server resource usage	<p>This report provides the scope of storage that is used by the server over time. It produces the following bar charts:</p> <p>Tapes Displays the total number of tapes that were used over a period of time that you select.</p> <p>Storage pool space Displays the total amount of storage pool space (KB) that was used over a period of time that you select.</p> <p>Database size Displays the total amount of database space (KB) that was used over a period of time that you select.</p>
Server throughput	<p>This report provides the data traffic load on the server.</p> <p>The operations bytes reported, client, migration, database backup, storage pool backup, expiration, and reclamation bytes, are calculated for the operation only, then the values reset back to zero. The bytes reported for each operation is not cumulative over time.</p> <p>This report produces a line chart that displays the following data:</p> <ul style="list-style-type: none"> • Total client bytes for a period of time that you select. • Migration for a period of time that you select. • Database backup for a period of time that you select. • Storage pool backup for a period of time that you select. • Expiration for a period of time that you select. • Reclamation for a period of time that you select.
Tape volume capacity analysis	<p>This report provides the efficiency with which tapes are being used. The utilization of a volume includes all space that is occupied by both files and aggregates, including empty space within aggregates. This does not include space formerly occupied by deleted files.</p>

Related tasks

“Running the Tivoli Storage Manager client and server reports” on page 667

Monitoring workspaces

You can open the monitoring workspaces to monitor the server status using the IBM Tivoli Monitoring Tivoli Enterprise Portal. Use these workspaces when you want to monitor areas of your Tivoli Storage Manager server in real time.

An *attribute* is a system or application element that is being monitored by the monitoring agent, such as agent status or client node storage. An *attribute group* is a set of related attributes that can be combined in a workspace view. When you open the view, data samples of the selected attributes are taken.

Table 60 on page 658 lists the attribute groups, their workspaces and descriptions.

Table 60. Tivoli Enterprise Portal attribute groups and workspaces

Attribute group name	Description
<p>Availability:</p> <ul style="list-style-type: none"> • Agent status • Agent performance 	<p>This attribute group provides you with workspaces that display the status and the performance of the agent running for each of the different attribute groups that are listed under the Tivoli Storage Manager agent. It aids you in identifying any problems with gathering the monitoring and historical data for any of the attribute groups.</p> <p>The following data is displayed in a tabular workspace for agent status:</p> <ul style="list-style-type: none"> • Query name • Object name (attribute group name) • Object type • Object status • Error code <p>The following data is displayed in a tabular workspace for agent performance:</p> <ul style="list-style-type: none"> • Application component • Name • Status • Full name • Virtual size • Page faults per second • Working set size • Thread count • Process ID • Percent privileged time • Percent user mode time • Percent processor time • Command line • Functional test status • Functional text message

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
Client node storage	<p>This attribute group provides you with each client node's storage and tape usage. You can identify which clients are using the most resources on the server.</p> <p>Right-click Client Node Storage to see the following workspaces:</p> <ul style="list-style-type: none"> • Client Node Storage: (default workspace) <ul style="list-style-type: none"> - Displays the following tabular data: <ul style="list-style-type: none"> - Client node name - Domain - Server - Disk usage - Tape volume - Maximum file size capacity - Filespace used - Removable used - Server used • Client Node Disk and Tape Usage: <ul style="list-style-type: none"> - Displays the following data as a bar chart: <ul style="list-style-type: none"> - Disk usage - Tape volume count • Client Node Filespace Usage: <ul style="list-style-type: none"> - Displays the following data as a bar chart: <ul style="list-style-type: none"> - File space used - Total file space
Client missed files	<p>This attribute group provides you with the status of missing files that are reported during client backups. You can identify which clients have a large number of missing files.</p> <p>The following data is displayed in a tabular workspace:</p> <ul style="list-style-type: none"> • Client/node name • Server • Time missed • File name • Full path • Missed reason (why the files were missed) <p>The following data is displayed as a bar chart in the workspace:</p> <ul style="list-style-type: none"> • Missed reason (total count, grouped by client/node name)
Client node status	<p>This attribute group provides you with backup-client node status. As client systems send data to the Tivoli Storage Manager server through manual or scheduled backups, the currency of those backups is important to provide you with an idea of what critical data was not backed up over a period of time.</p> <p>The following data is displayed in a tabular workspace:</p> <ul style="list-style-type: none"> • Client node name • Domain • Server • Last successful backup date • Last successful warnings date • Last failed backup

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
Database	<p>This attribute group provides you with information that you can monitor and determine when all of the allocated database space is used up. If all the allocated space is used, expansion operations can be taken to assure the database continues to operate. You are also provided with the database backup status as the last full backup and the last incremental backup.</p> <p>As a Tivoli Storage Manager server processes client requests for backup-archive operations, the Tivoli Storage Manager database is updated with current and historical types of data.</p> <p>The following data is displayed as a bar chart:</p> <ul style="list-style-type: none"> • Current Size • Total Space Used <p>The following data is displayed in a tabular workspace:</p> <ul style="list-style-type: none"> • Server name • Current Size • Percent space used • Total space used • Last backup date • Backup duration • Backup status • Last increment date • Increment duration • Incremental back status • Volumes used <p>Note: Time differences can occur because the database is rounded to the nearest minute while the time is calculated to a decimal minute.</p>

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
<p>Node Activity:</p> <ul style="list-style-type: none"> • Client restore 	<p>This attribute group provides you with each client's and server's activities on the Tivoli Storage Manager server. This workspace has seven sub-workspaces available for you to view.</p> <p>The Node Activity workspace is the main workspace and only displays a table. This table has a link option that you can click on to link to any of the other sub-workspaces.</p> <ul style="list-style-type: none"> • The Node Activity workspace displays the following tabular data for each of the sub-workspaces: <ul style="list-style-type: none"> – Node name – Start time – End time – Type – Schedule name – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time <p>Right-click on the Node Activity workspace to see the following sub-workspaces:</p> <ul style="list-style-type: none"> • The Client Restore sub-workspace displays the following tabular data: <ul style="list-style-type: none"> – Server name – Node name – Start time – End time – Type – Schedule name – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time <p>The Client Restore sub-workspace also displays the following data in a bar chart:</p> <ul style="list-style-type: none"> – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
Node activity - Client backup	<ul style="list-style-type: none"> • The Client Backup sub-workspace displays the following tabular data: <ul style="list-style-type: none"> – Server name – Node name – Start time – End time – Type – Schedule name – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time <p>The Client Backup sub-workspace also displays the following data in a bar chart:</p> <ul style="list-style-type: none"> – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time
Node activity - Client archive	<ul style="list-style-type: none"> • The Client Archive sub-workspace displays the following tabular data: <ul style="list-style-type: none"> – Server name – Node name – Start time – End time – Type – Schedule name – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time <p>The Client Archive sub-workspace also displays the following data in a bar chart:</p> <ul style="list-style-type: none"> – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
Node activity - Client retrieve	<ul style="list-style-type: none"> • Client Retrieve subworkspace displays the following tabular data: <ul style="list-style-type: none"> – Server name – Node name – Start time – End time – Type – Schedule name – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time <p>The Client Retrieve subworkspace also displays the following data in a bar chart:</p> <ul style="list-style-type: none"> – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time
Node activity - Server database backup	<ul style="list-style-type: none"> • The Server Database Backup sub-workspace displays the following tabular data: <ul style="list-style-type: none"> – Server name – Node name – Start time – End time – Type – Schedule name – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time <p>The Server Database Backup subworkspace also displays the following data in a bar chart:</p> <ul style="list-style-type: none"> – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
Node activity - Server file expiration	<ul style="list-style-type: none"> • The Server File Expiration sub-workspace displays the following tabular data: <ul style="list-style-type: none"> – Server name – Start time – End time – Type – Scheduled name – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time <p>The Server File Expiration sub-workspace also displays the following data in a bar chart:</p> <ul style="list-style-type: none"> – Inspected objects – Processed objects – Failed objects – Total KBytes – Elapsed time
Schedule	<p>This attribute group provides you with all the scheduled events for the server and these events' status. You can group the data by node name, schedule name or status to help in identifying any possible problems.</p> <p>The data is displayed in the following tabular workspaces:</p> <ul style="list-style-type: none"> • Client schedules: <ul style="list-style-type: none"> – Schedule name – Domain name – Node name – Node type – Schedule start – Actual start – Schedule status – Schedule results – Error warning • Server schedules: <ul style="list-style-type: none"> – Server name – Schedule name – Node type – Schedule start – Actual start – Schedule status – Schedule results – Error warning – Domain name

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
Storage pool	<p>This attribute group provides you with information about your storage pools. Tivoli Storage Manager can contain multiple storage pools. These storage pools define the methods and resources that are used to store data being backed up or archived to the Tivoli Storage Manager server.</p> <p>The following data is displayed as a bar chart in the workspace:</p> <ul style="list-style-type: none"> • Total space • Total usage • Tape volume used for each storage pool defined on a server <p>The following data is displayed in a tabular workspace:</p> <ul style="list-style-type: none"> • Storage pool name • Server name • Device class • Total space • Space usage • Total volumes used

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
Server	<p>This attribute group provides you with the operational status of the Tivoli Storage Manager server. It gives you the ability to view the following activities or status:</p> <ul style="list-style-type: none"> • What activities are taking time to complete • As the server migrates data or mounts storage onto devices, what are the possible problem activities • The status of server-only activities on the server. <p>The following operations are included in this workspace:</p> <ul style="list-style-type: none"> • Client • Migration • Database backup • Storage pool backup • Expiration • Eeclamation <p>These operations are measured by bytes per operation. Once reported, the values are reset back to zero. The bytes reported for each operation are not cumulative over time.</p> <p>The following data is displayed in a tabular workspace:</p> <ul style="list-style-type: none"> • Data displayed server ID • Client operation byte count • Client operation duration • Data Base Backup operation byte count • Data Base Backup operation duration • Migration operation byte count • Migration operation duration • Reclamation operation byte count • Reclamation operation duration • Storage pool backup operation byte count • Storage pool backup operation duration <p>The following data is displayed as a bar chart in the workspace:</p> <p>Operation byte count Displays all of the operation-byte-count values for the server.</p> <p>Operation duration Displays all of the operation duration values for the server.</p>
Storage device	<p>This attribute group provides you with the read and write error status of the storage devices. This status helps you identify possible problems with any of your storage devices.</p> <p>The following data is displayed in a tabular workspace:</p> <ul style="list-style-type: none"> • Data displayed server name • Device name • Device class • Write errors • Read errors <p>The following data is displayed as a bar chart in a workspace:</p> <ul style="list-style-type: none"> • Write errors • Read errors

Table 60. Tivoli Enterprise Portal attribute groups and workspaces (continued)

Attribute group name	Description
Tape usage	<p>This attribute group provides you with the tape usage per client.</p> <p>The following data is displayed in a tabular workspace:</p> <ul style="list-style-type: none"> • Volume ID • Server name • Client • Storage pool
Tape volume	<p>This attribute group provides you with a status of all your tape storage devices. It helps you identify any storage devices that are near full capacity.</p> <p>The following data is displayed in a tabular workspace:</p> <ul style="list-style-type: none"> • Volume ID • Capacity • Used capacity • Unused capacity • Used Capacity • Unused Capacity <p>The following data is displayed as a pie chart in the workspace:</p> <ul style="list-style-type: none"> • Unused capacity versus used capacity

Running the Tivoli Storage Manager client and server reports

You can view the trends of your client and server systems using the Tivoli Storage Manager Tivoli Integrated Portal.

In order to view historical reports, you must have completed several configuration tasks after installing the Tivoli Storage Manager reporting and monitoring feature. See the section on after installing the reporting and monitoring feature for details.

To run the available Tivoli Storage Manager client and server reports, complete the following steps:

1. Log in to the Tivoli Storage Manager Tivoli Integrated Portal.
 - a. If the Tivoli Integrated Portal is not running, start it by running the following commands from a command line:
 - 1) `cd TIP_home\profiles\TIPProfile\bin`
 - 2) `startServer.sh server1 -user tipuser -password tippassword`
 where *TIP_home* is the root directory for your Tivoli Integrated Portal installation. The default location is `/opt/ibm/ac`.
 - b. Open a Web browser and enter the following address: `http://hostname:port`, where *port* is the port number specified when you installed the Tivoli Integrated Portal. The default port is 16310.

If you are using a remote system, you can access the Tivoli Integrated Portal by entering the IP address or fully qualified host name of the remote system. You might have to authenticate to the remote system if there is a firewall that exists.
 - c. The Tivoli Integrated Portal window opens. In the **User ID** field, enter the Tivoli Integrated Portal user ID that was defined when you installed the Tivoli Storage Manager reporting and monitoring feature.

- d. In the **Password** field, enter the Tivoli Integrated Portal password you defined in the installation wizard and click **Log in**.
2. On the left side of the window, expand and click **Common Reporting** → **Work with reports**.
3. In the **Navigation** tab of the **Common Reporting** pane, complete the following steps: for all reports, click **Report Sets** → **Tivoli Products** → **Common Reporting**
 - a. For client reports: Click **Tivoli Products** → **Tivoli Storage Manager** → **Client Reports**.
 - b. For server reports: Click **Tivoli Products** → **Tivoli Storage Manager** → **Server Reports**.

The report name and descriptions are displayed in the **Reports** pane. Select a report, right-click **View As** → **HTML** or **->PDF**.

Monitoring Tivoli Storage Manager real-time data

You can open the monitoring workspaces to monitor a server through the IBM Tivoli Monitoring Tivoli Enterprise Portal. View these workspaces when you want to monitor areas of your Tivoli Storage Manager Server in real-time.

To view the available Tivoli Storage Manager monitoring workspaces, complete the following steps:

1. Start the IBM Tivoli Monitoring Tivoli Enterprise Portal. You can start the portal using one of the following ways:
 - a. Start the Tivoli Enterprise Monitoring Services console by completing one of the following steps:
 - 1) Run the Candle Manage program issuing the following command:


```
/opt/ibm/itm/bin/CandleManage
```

In the Manage Tivoli Enterprise Monitoring Services window, right-click **Tivoli Enterprise Portal Desktop Client**. Click **Configure**. Enter the system hostname in the **host name** field for the portal server and click **Save**.

- b. Open a web browser and enter the address of the server where the Tivoli Enterprise Portal server is installed, similar to the following example:


```
http://hostname:1920///cnp/kdh/lib/cnp.html
```

 where *hostname* is the server name and *1920///cnp/kdh/lib/cnp.html* is the port and location of the Java Tivoli Enterprise Portal Web Client.

Tip: Always use this port number and location.

2. In the Logon window, enter the User ID in the **Logon ID** field, and the password in the **Password** field. These were defined when you installed the Tivoli Storage Manager reporting and monitoring feature. Click **OK** and the Tivoli Enterprise Portal opens.
3. In the left **Navigator** pane, click to open **Windows Systems** → *server name*.
4. Click the Tivoli Storage Manager attribute group.
5. Select the workspace that you want to view.

Tip: Some of these attribute groups have sub-workspaces that you can view when you right-click the main attribute group. See the section on the overview of the monitoring workspaces to learn more details about using the workspaces.

- The details of your selection are displayed in the workspace in the right panel and in the bottom panel.

Related reference

“Monitoring workspaces” on page 657

Modifying the IBM Tivoli Monitoring environment file

You can change the way the Tivoli Storage Manager monitoring agent behaves by modifying the environment file.

When you create a Tivoli Storage Manager monitoring agent instance in the Tivoli Enterprise Monitoring Services application, a new environment file is created. You can modify this file to change the behavior of the monitoring agent.

There are a large number of variables that can be configured, but care must be taken to not destroy performance of the Tivoli Storage Manager server by setting variables incorrectly.

The environment file is named **sk_xxx.config**, where *xxx* is the instance name of the monitoring agent you created. This file is located in the `/opt/tivoli/tsm/reporting/itm/config` directory on both Linux and AIX systems.

Modifying the IBM Tivoli Monitoring environment file for reporting queries

Using the environment file that was automatically created for you when you added a Tivoli Storage Manager monitoring agent instance, you can modify the environment variables to query the data you want.

The following list contains the environment variables that you can change to modify the monitoring agent and request particular queries of your choice. Use any text editor to edit the file. If you enter a value that is not valid, the query is turned off.

KSK_APITRACE, Default Value= 0

If an error condition occurs and trace information is needed by IBM/Tivoli support, this variable value set to 1 creates a trace file for the Tivoli Storage Manager Administrator's API. This file can grow in size and should only be used if instructed by IBM/Tivoli support personnel. Valid values are 0 and 1.

KSK_CMF_ON, Default Value= 1

Queries the Tivoli Storage Manager server for the Client Missed Files data. 1 =On, 0 = Off

KSK_CNS_ON, Default Value=1

Queries the Tivoli Storage Manager server for the Client Node Status data. 1 =On, 0 = Off

KSK_CNSTG_ON, Default Value=1

Queries the Tivoli Storage Manager server for the Client Node Storage data. 1 =On, 0 = Off

KSK_DB_ON, Default Value=1

Queries the Tivoli Storage Manager server for the Database data. 1 =On, 0 = Off

KSK_NODEA_ON, Default Value= 1

Queries the Tivoli Storage Manager server for the Node Activity data. 1 =On, 0 = Off

KSK_SCHED_ON, Default Value=1

Queries the Tivoli Storage Manager server for the Schedule data. 1 = On, 0 = Off

KSK_SERVER_ON, Default Value=1

Queries the Tivoli Storage Manager server for the server data. 1 = On, 0 = Off

KSK_STGDEV_ON, Default Value=1

Queries the Tivoli Storage Manager server for the Storage Device data. 1 = On, 0 = Off

KSK_TAPEUSG_ON, Default Value=1

Queries the Tivoli Storage Manager server for the Tape Usage data. 1 = On, 0 = Off

KSK_TAPEVOL_ON, Default Value=1

Queries the Tivoli Storage Manager server for the Tape Volume data. 1 = On, 0 = Off

KSK_TRACE, Default Value=0

This value set to 1 allows the Tivoli Storage Manager Tivoli Common Reporting data collection agent to create a log file showing its attempts to query both the Tivoli Storage Manager server and the DERBY pre-fetch data cache. Valid values are 0 and 1.

There are other variables included in this environment file that deal with performance of the server. See the *IBM Tivoli Storage Manager Performance Tuning Guide* for details of these environment variables.

Chapter 23. Managing a network of Tivoli Storage Manager servers

You might have several Tivoli Storage Manager servers in your network, at the same or different locations. Tivoli Storage Manager provides functions to help you configure, manage, and monitor the servers connected to a network.

An administrator working at one Tivoli Storage Manager server can work with Tivoli Storage Manager servers at other locations around the world.

See the following topics:

Concepts:
“Concepts for managing server networks”
“Enterprise configuration” on page 672

Tasks:
“Setting up communications among servers” on page 676
“Setting up communications for enterprise configuration and enterprise event logging” on page 676
“Setting up communications for command routing with multiple source servers” on page 681
“Performing tasks on multiple servers” on page 707
“Using virtual volumes to store data on another server” on page 712

Concepts for managing server networks

In a network of Tivoli Storage Manager servers, a server can play several different roles. For example, a server can send volumes to be archived on another server and also receive routed commands from a different server.

To manage a network of servers, you can use the following Tivoli Storage Manager capabilities:

- Configure and manage multiple servers with enterprise configuration.
Distribute a consistent configuration for Tivoli Storage Manager servers through a configuration manager to managed servers. By having consistent configurations, you can simplify the management of a large number of servers and clients.
- Perform tasks on multiple servers by using command routing, enterprise logon, and enterprise console.
- Send server and client events to another server for logging.
- Monitor many servers and clients from a single server.
- Store data on another server by using virtual volumes.

In the descriptions for working with a network of servers, when a server sends data, that server is sometimes referred to as a *source server*, and when a server receives data, it is sometimes referred to as a *target server*. In other words, one

Tivoli Storage Manager server may be both a source and a target server. At the same time, any Tivoli Storage Manager server can still provide backup, archive, and space management services to clients.

For details, see “Licensing IBM Tivoli Storage Manager” on page 555.

Enterprise configuration

The Tivoli Storage Manager enterprise configuration functions make it easier to consistently set up and manage a network of Tivoli Storage Manager servers. You can set up configurations on one server and distribute the configurations to other servers. You can make changes to configurations and have the changes automatically distributed.

Figure 87 on page 673 illustrates a simple configuration. To use enterprise configuration, select the Tivoli Storage Manager server that is to act as the *configuration manager*. You might want to dedicate a new server for this purpose. At the configuration manager, define the details of the server configurations that you want to distribute. For example:

- Set up backup and archive policies and client option sets
- Designate one or more administrators to have access to the servers, and control their authority levels
- Define the servers that you want the configuration manager to manage or communicate with, and you set up communications among the servers

In one or more *profiles*, point to the definitions of the configuration information that you want to use to manage other servers.

On each server that is to receive the configuration information, identify the server as a *managed server* by defining a *subscription* to one or more profiles owned by the configuration manager. All the definitions associated with the profiles are then copied into the managed server's database. Things defined to the managed server in this way are managed objects that cannot be changed by the managed server. From then on, the managed server gets any changes to the managed objects from the configuration manager via the profiles. Managed servers receive changes to configuration information at time intervals set by the servers, or by command.

For details, see “Setting up enterprise configurations” on page 685.

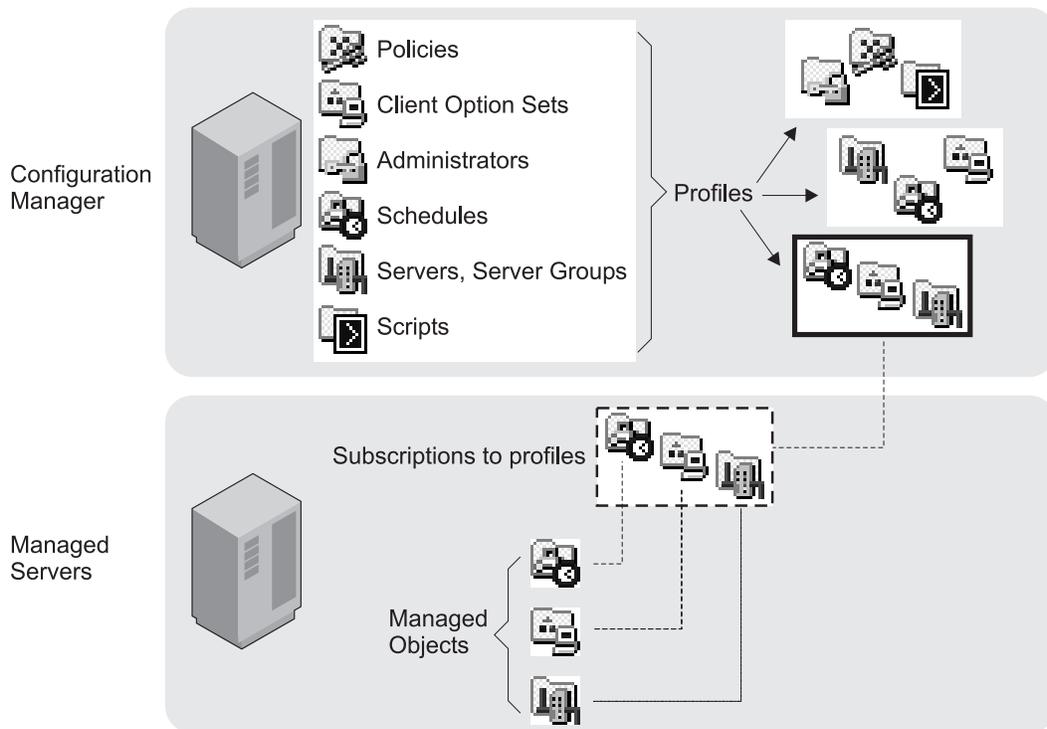


Figure 87. Enterprise configuration

Command routing

Use the administrative client or Administration Center command-line interface to route commands to other servers.

The other servers must be defined to the server to which you are connected. You must also be registered on the other servers as an administrator with the administrative authority that is required for the command. To make routing commands easier, you can define a server group that has servers as members. Commands that you route to a server group are sent to all servers in the group.

For details, see “Setting up server groups” on page 710, “Routing commands” on page 707, and Chapter 18, “Managing servers with the Administration Center,” on page 547.

Central monitoring for the Tivoli Storage Manager server

Tivoli Storage Manager provides you with several ways to centrally monitor the activities of a server network.

The following methods are ways in which you can centrally monitor activities:

- Enterprise event logging, in which events are sent from one or more of servers to be logged at an event server.
For a description of the function, see “Enterprise event logging: logging events to another server” on page 639. For information about communications setup, see “Setting up communications for enterprise configuration and enterprise event logging” on page 676.
- Allowing designated administrators to log in to any of the servers in the network with a single user ID and password.

See “Working with multiple servers using the Administration Center” on page 707.

- Routing query commands to one or more of the servers in the network.

For a description of the function, see “Routing commands to one or more servers” on page 708. For information about communications setup, see “Setting up communications for enterprise configuration and enterprise event logging” on page 676.

Data storage on another server

Tivoli Storage Manager lets one server store data in and retrieve data from the storage pool of another server. This data, stored as *virtual volumes*, can include database and storage pool backups, disaster recovery plan files, and data that is directly backed up, archived, or space managed from client nodes.

To use virtual volumes to store database and storage pool backups and recovery plan files, you must have the disaster recovery manager function. For details, see “Licensing IBM Tivoli Storage Manager” on page 555.

Examples: management of multiple Tivoli Storage Manager servers

The functions for managing multiple servers can be applied in many ways.

Here are two scenarios to give you some ideas about how you can use the functions:

- Setting up and managing Tivoli Storage Manager servers primarily from one location. For example, an administrator at one location controls and monitors servers at several locations.
- Setting up a group of Tivoli Storage Manager servers from one location, and then managing the servers from any of the servers. For example, several administrators are responsible for maintaining a group of servers. One administrator defines the configuration information on one server for distributing to servers in the network. Administrators on the individual servers in the network manage and monitor the servers.

Example: management of multiple servers from one location

Enterprise management lets you set up and manage the servers in your network from one location, the enterprise console.

For example, suppose that you are an administrator who is responsible for Tivoli Storage Manager servers at your own location, plus servers at branch office locations. Servers at each location have similar storage resources and client requirements. You can set up the environment as follows:

- Set up an existing or new Tivoli Storage Manager server as a configuration manager.
- Set up communications so that a configuration manager can send commands to its managed servers.
- Define the configuration you want to distribute by defining policy domains, schedules, and so on. Associate the configuration information with profiles.
- Have the managed servers subscribe to profiles.
- Activate policies and set up storage pools as needed on the managed servers.

- Set up enterprise monitoring by setting up one server as an event server. The event server can be the same server as the configuration manager or a different server.

After you complete the setup, you can manage many servers as if there was just one. You can perform any of the following tasks:

- Have administrators that can manage the group of servers from anywhere in the network by using the enterprise console, an interface available through a Web browser.
- Have consistent policies, schedules, and client option sets on all servers.
- Make changes to configurations and have the changes automatically distributed to all servers. Allow local administrators to monitor and tune their own servers.
- Perform tasks on any server or all servers by using command routing from the enterprise console.
- Back up the databases of the managed servers on the automated tape library that is attached to the server that is the configuration manager. You use virtual volumes to accomplish this.
- Log on to individual servers from the enterprise console without having to re-enter your password, if your administrator ID and password are the same on each server.

Example: management of multiple servers from any server

Enterprise management lets you manage the servers in your network from many locations.

For example, suppose that you are an administrator responsible for servers located in different departments on a college campus. The servers have some requirements in common, but also have many unique client requirements. You can set up the environment as follows:

- Set up an existing or new Tivoli Storage Manager server as a configuration manager.
- Set up communications so that commands can be sent from any server to any other server.
- Define any configuration that you want to distribute by defining policy domains, schedules, and so on, on the configuration manager. Associate the configuration information with profiles.
- Have the managed servers subscribe to profiles as needed.
- Activate policies and set up storage pools as needed on the managed servers.
- Set up enterprise monitoring by setting up one server as an event server. The event server can be the same server as the configuration manager or a different server.

After setting up in this way, you can manage the servers from any server. You can do any of the following tasks:

- Use enterprise console to monitor all the servers in your network.
- Perform tasks on any or all servers using the enterprise console and command routing.
- Manage the group of servers from anywhere in the network. Allow local administrators to monitor and tune their own servers.

Enterprise-administration planning

To take full advantage of the functions of enterprise administration, you should decide on the servers you want to include in the enterprise network, the server from which you want to manage the network, and other important issues.

Consider the following items when planning for Enterprise Administration:

- The servers you want to include in the enterprise network. The servers must have unique names.
- The server or servers from which you want to manage the network. Servers can have multiple roles. For example, one server can act as a server for backup-archive clients, as the configuration manager, and as the event server. You can also set up separate servers to fill each of these roles.
- Whether you want administrators to have the ability to route commands to other servers. If you want administrators to route commands, decide on the servers from which and to which commands will be routed.
- The administrator activities you want to be centrally managed.
- The authority level of the administrators and the servers to which they should have access.

Setting up communications among servers

You can set up communications for enterprise configuration, enterprise event logging, and command routing. When you set up communications among servers for any purpose, ensure that servers have unique names.

Communication setup for server-to-server virtual volumes is described in “Setting up source and target servers for virtual volumes” on page 714. See “Setting the server name” on page 567 for more information before using the SET SERVERNAME command.

Setting up communications for enterprise configuration and enterprise event logging

The communication setup for enterprise configuration and enterprise event logging, which is through TCP/IPv4 or IPv6, is identical.

The examples shown here apply to both functions. If you are set up for one, you are set up for the other. However, be aware that the configuration manager and event server are not defined simply by setting up communications. You must identify a server as a configuration manager (SET CONFIGMANAGER command) or an event server (DEFINE EVENTSERVER command). Furthermore, a configuration manager and an event server can be the same server or different servers.

Enterprise configuration

Each managed server must be defined to the configuration manager, and the configuration manager must be defined to each managed server.

Enterprise event logging

Each server sending events to an event server must be defined to the event server, and the event server must be defined to each source server.

The following examples of setting up communications could be used to create these configurations:

- A server named HEADQUARTERS as a configuration manager and two servers, MUNICH and STRASBOURG, as managed servers.
- HEADQUARTERS as an event server and MUNICH and STRASBOURG as source servers.

For a pair of servers to communicate with each other, each server must be defined to the other. For example, if a configuration manager manages three managed servers, there are three server pairs. You can issue separate definitions from each server in each pair, or you can “cross define” a pair in a single operation. Cross definition can be useful in large or complex networks. The following scenarios and accompanying figures illustrate the two methods.

Using separate definitions – Follow this sequence:

1. **On MUNICH:** Specify the server name and password of MUNICH.
On STRASBOURG: Specify the server name and password of STRASBOURG.
On HEADQUARTERS: Specify the server name and password of HEADQUARTERS.
2. **On HEADQUARTERS:** Define MUNICH (whose password is BERYL and whose address is 9.115.2.223:1919) and STRASBOURG (whose password is FLUORITE and whose address is 9.115.2.178:1715).
On MUNICH and STRASBOURG: Define HEADQUARTERS (whose password is AMETHYST and whose address is 9.115.4.177:1823).

Figure 88 on page 678 shows the servers and the commands issued on each:

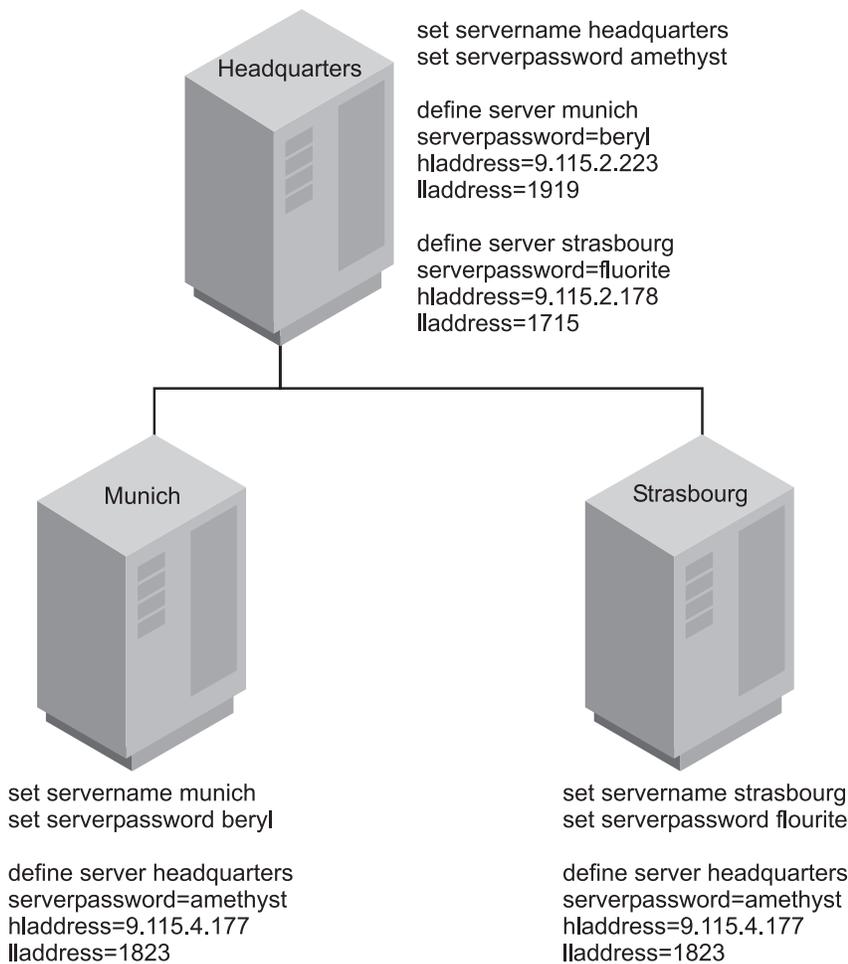


Figure 88. Communication configuration with separate server definitions

Using Cross Definitions – Follow this sequence:

1. **On MUNICH:** Specify the server name, password, and high and low level addresses of MUNICH. Specify that cross define is permitted.
On STRASBOURG: Specify the server name, password, and high and low level addresses of STRASBOURG. Specify that cross define is permitted.
On HEADQUARTERS: Specify the server name, password, and high and low level addresses of HEADQUARTERS.
2. **On HEADQUARTERS:** Define MUNICH and STRASBOURG, specifying that cross define should be done.

Figure 89 on page 679 shows the servers and the commands issued on each:

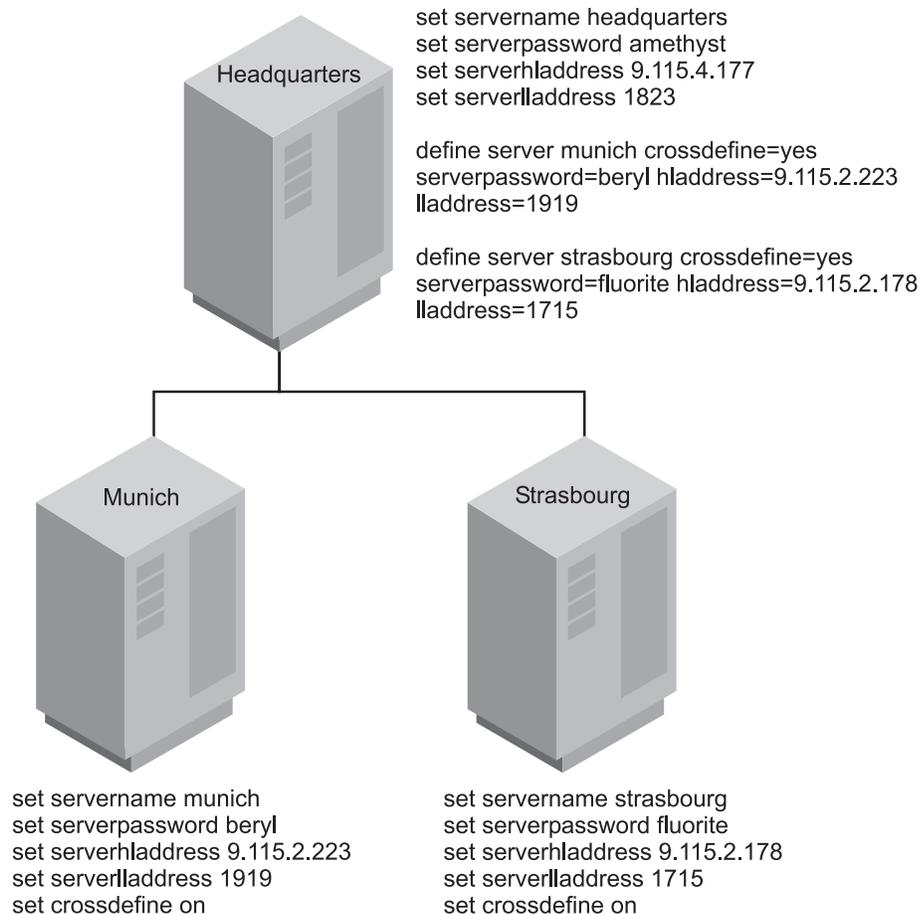


Figure 89. Communication configuration with cross definition

Note: Issuing the SET SERVERNAME command can affect scheduled backups until a password is re-entered. 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 re-enter the passwords. On a network where clients connect to multiple servers, it is recommended that all of the servers have unique names. See the *Administrator's Reference* for more details.

Communication security

Security for this communication configuration is enforced through the exchange of passwords (which are encrypted) and, in the case of enterprise configuration only, verification keys.

Communication among servers, which is through TCP/IP, requires that the servers verify server passwords (and verification keys). For example, assume that HEADQUARTERS begins a session with MUNICH:

1. HEADQUARTERS, the source server, identifies itself by sending its name to MUNICH.
2. The two servers exchange verification keys (enterprise configuration only).
3. HEADQUARTERS sends its password to MUNICH, which verifies it against the password stored in its database.
4. If MUNICH verifies the password, it sends its password to HEADQUARTERS, which, in turn, performs password verification.

Note: If another server named MUNICH tries to contact HEADQUARTERS for enterprise configuration, the attempt fails. This is because the verification key will not match. If MUNICH was moved or restored, you can issue the UPDATE SERVER command with the FORCERESYNC parameter to override the condition.

Setting up communications for command routing

When setting up communications for command routing, you must define the target servers to the source servers, and the same administrator must be registered on all servers. Using enterprise configuration, you can easily distribute the administrator information to all the servers.

Note: You must be registered as an administrator with the same name and password on the source server and all target servers. The privilege classes do not need to be the same on all servers. However, to successfully route a command to another server, an administrator must have the minimum required privilege class for that command on the server from which the command is being issued.

For command routing in which one server will always be the sender, you would only define the target servers to the source server. If commands can be routed from any server to any other server, each server must be defined to all the others.

Setting up communications for command routing with one source server

The process of setting up communications for command routing can, at times, be a challenge.

The example provided shows you how you can set up communications for administrator HQ on the server HEADQUARTERS who will route commands to the servers MUNICH and STRASBOURG. Administrator HQ has the password SECRET and has system privilege class.

The procedure for setting up communications for command routing with one source server is shown in the following list:

- **On HEADQUARTERS:** register administrator HQ and specify the server names and addresses of MUNICH and STRASBOURG:

```
register admin hq secret
grant authority hq classes=system
```

```
define server munich serverpassword=bery1 h1address=9.115.2.223 l1address=1919
define server strasbourg serverpassword=fluorite h1address=9.115.2.178
l1address=1715
```

Note: Command routing uses the ID and password of the Administrator. It does not use the password or server password set in the server definition.

- **On MUNICH and STRASBOURG** Register administrator HQ with the required privilege class on each server:

```
register admin hq secret
grant authority hq classes=system
```

Note: If your server network is using enterprise configuration, you can automate the preceding operations. You can distribute the administrator and server lists to MUNICH and STRASBOURG. In addition, all server definitions and server groups are distributed by default to a managed server when it first subscribes to any profile on a configuration manager. Therefore, it receives all the server definitions that exist on the configuration manager, thus enabling command routing among the servers.

Setting up communications for command routing with multiple source servers

When setting up communications for command routing, you must define all the servers to each other.

The examples provided below show you how to set up communications if the administrator, HQ, can route commands from any of the three servers to any of the other servers. You can separately define each server to each of the other servers, or you can “cross define” the servers. In cross definition, defining MUNICH to HEADQUARTERS also results in automatically defining HEADQUARTERS to MUNICH.

Creating separate definitions:

When setting up communications for command routing, you can define each server to each of the other servers.

To create separate definitions:

1. **On MUNICH:** Specify the server name and password of MUNICH. Register administrator HQ and grant HQ system authority.
On STRASBOURG: Specify the server name and password of STRASBOURG. Register administrator HQ and grant HQ system authority.
On HEADQUARTERS: Specify the server name and password of HEADQUARTERS. Register administrator HQ and grant HQ system authority.
2. **On HEADQUARTERS:** Define MUNICH (whose password is BERYL and whose address is 9.115.2.223:1919) and STRASBOURG (whose password is FLUORITE and whose address is 9.115.2.178:1715).
On MUNICH: Define HEADQUARTERS (whose password is AMETHYST and whose address is 9.115.4.177:1823) and STRASBOURG.
On STRASBOURG: Define HEADQUARTERS and MUNICH.

Figure 90 on page 682 shows the servers and the commands issued on each.

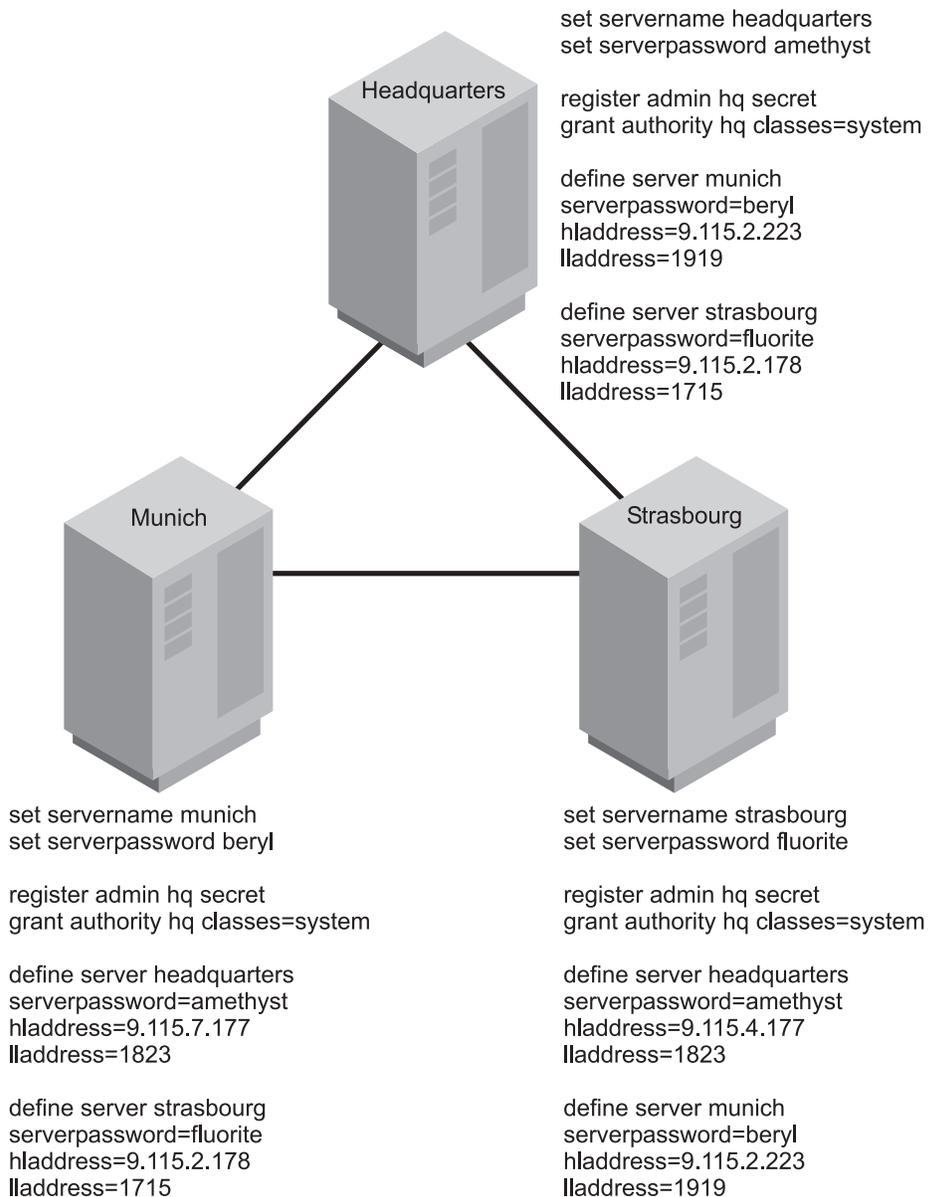


Figure 90. Communication configuration with separate server definitions

Creating cross definitions:

When setting up communications for command routing, you can cross-define the other servers.

To create cross definitions:

1. **On MUNICH:** Specify the server name, password, and high and low level addresses of MUNICH. Specify that cross define is permitted. Register administrator HQ and grant HQ system authority.
 - On STRASBOURG:** Specify the server name, password, and high and low level addresses of STRASBOURG. Specify that cross define is permitted. Register administrator HQ and grant HQ system authority.
 - On HEADQUARTERS:** Specify the server name, password, and high and low level addresses of HEADQUARTERS. Register administrator HQ and grant HQ system authority.

2. **On HEADQUARTERS:** Define MUNICH and STRASBOURG, specifying that cross define should be done.
3. **On MUNICH:** Define STRASBOURG, specifying that cross define should be done.

Note: If your server network is using enterprise configuration, you can automate the preceding operations. You can distribute the administrator lists and server lists to MUNICH and STRASBOURG. In addition, all server definitions and server groups are distributed by default to a managed server when it first subscribes to any profile on a configuration manager. Therefore, it receives all the server definitions that exist on the configuration manager, thus enabling command routing among the servers.

Figure 91 shows the servers and the commands issued on each.

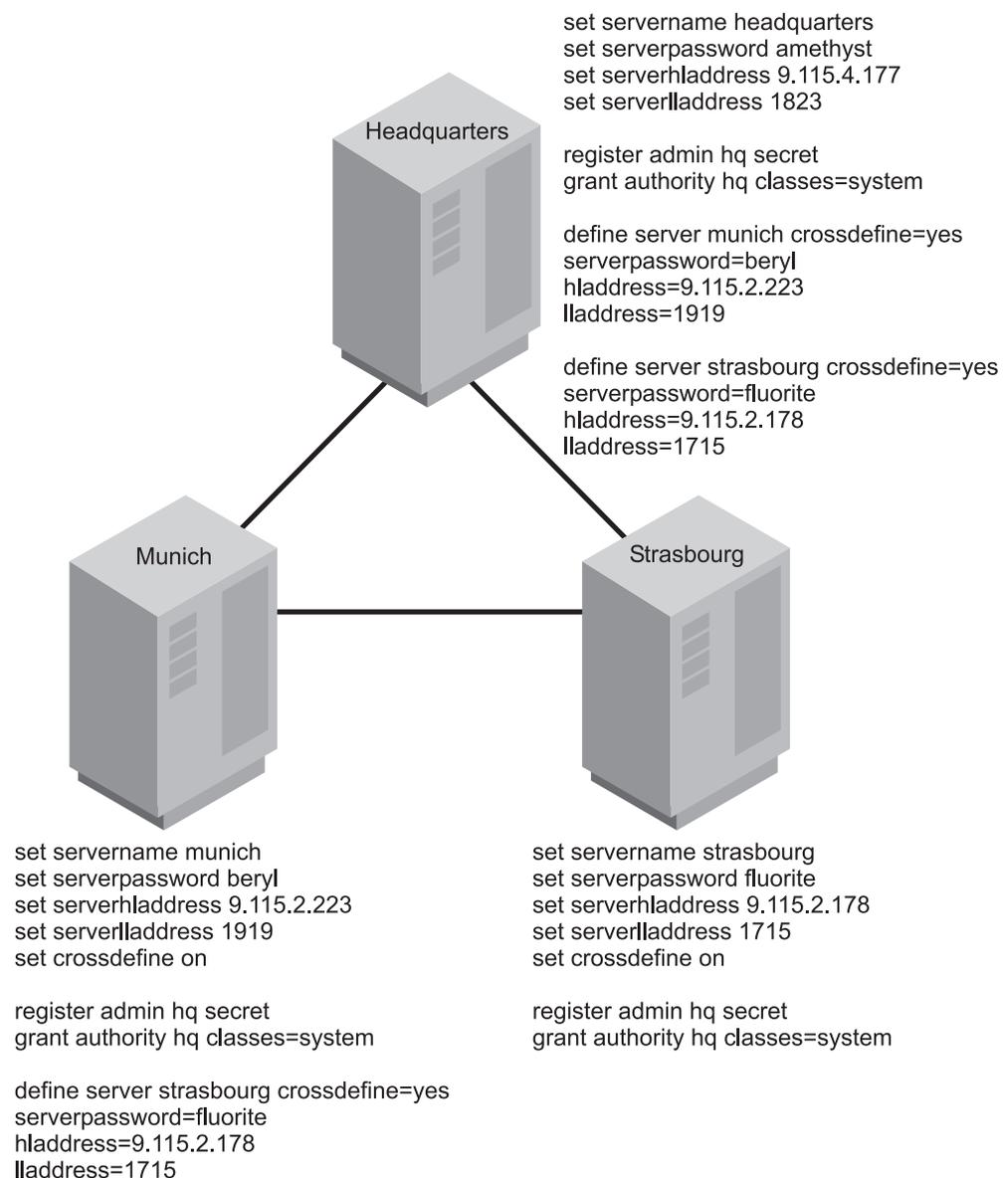


Figure 91. Communication configuration with cross definitions

Updating and deleting servers

You can update and delete server definitions for server-to-server virtual volumes, enterprise configuration, and enterprise event logging.

You can update a server definition by issuing the UPDATE SERVER command.

- For server-to-server virtual volumes:
 - If you update the node name, you must also update the password.
 - If you update the password but not the node name, the node name defaults to the server name specified by the SET SERVERNAME command.
- For enterprise configuration and enterprise event logging: If you update the server password, it must match the password specified by the SET SERVERPASSWORD command at the target server.
- For enterprise configuration: When a server is first defined at a managed server, that definition cannot be replaced by a server definition from a configuration manager. This prevents the definition at the managed server from being inadvertently replaced. Such a replacement could disrupt functions that require communication among servers, for example command routing or virtual volumes.

To allow replacement, update the definition at the managed server by issuing the UPDATE SERVER command with the ALLOWREPLACE=YES parameter. When a configuration manager distributes a server definition, the definition always includes the ALLOWREPLACE=YES parameter.

You can delete a server definition by issuing the DELETE SERVER command. For example, to delete the server named NEWYORK, enter the following:

```
delete server newyork
```

The deleted server is also deleted from any server groups of which it is a member.

You cannot delete a server if any of the following conditions are true:

- The server is defined as an event server.
You must first issue the DELETE EVENTSERVER command.
- The server is a target server for virtual volumes.
A target server is named in a DEFINE DEVCLASS (DEVTYPE=SERVER) command. You must first change the server name in the device class or delete the device class.
- The server is named in a device class definition whose device type is SERVER.
- The server has paths defined to a file drive.
- The server has an open connection to or from another server.
You can find an open connection to a server by issuing the QUERY SESSION command.

See “Setting up server groups” on page 710 for information about server groups.

Setting up enterprise configurations

With profiles, you can designate the configuration information that is distributed to managed servers. Then you can set up other servers as managed servers. The managed servers receive configuration information through subscriptions to profiles on the configuration manager.

Each managed server stores the distributed information as managed objects in its database. Managed servers receive periodic updates of the configuration information from the configuration manager, or an administrator can trigger an update by command.

You can distribute the following configuration information from a configuration manager to managed servers:

- Administrators, including authorities for them
- Policy objects, including policy domains, and the policy sets, management classes, copy groups and client schedules associated with them.
- Administrative command schedules
- Tivoli Storage Manager server scripts
- Client option sets
- Server definitions
- Server groups

“Enterprise configuration scenario” gives you an overview of the steps to take for one possible implementation of enterprise configuration. Sections that follow give more details on each step. For details on the attributes that are distributed with these objects, see “Associating configuration information with a profile” on page 690. After you set up server communication as described in “Setting up communications for enterprise configuration and enterprise event logging” on page 676, you set up the configuration manager and its profiles.

Enterprise configuration scenario

To illustrate how you might use the enterprise configuration functions, suppose that your enterprise has offices around the world, with one or more Tivoli Storage Manager servers at each location. To make managing these servers easier, you want to control the configuration of all Tivoli Storage Manager servers from one Tivoli Storage Manager server in the headquarters office.

Figure 92 on page 686 shows the hierarchy that you want to set up.

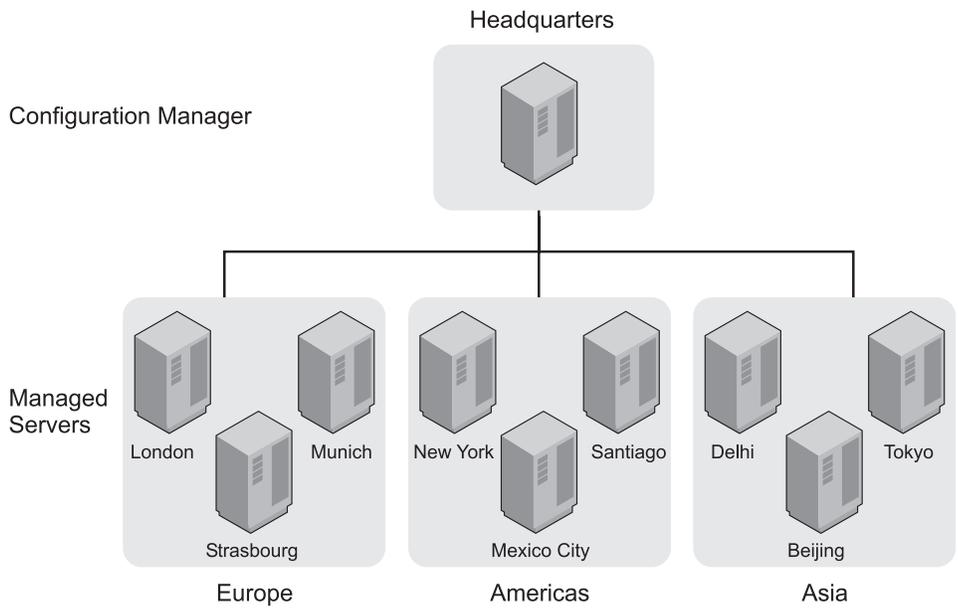


Figure 92. A scenario for implementing enterprise configuration

You want to set up a configuration manager named HEADQUARTERS. Managed servers have the names of cities where they are located. You have three groups of managed servers, one in the Americas, one in Europe, and one in Asia. Each of the servers supports backup and archive services for client machines in that office. For client backup operations, you want to use the default policy that stores backups on disk. Each server has an automated tape library configured to work with Tivoli Storage Manager, and you want to use the tape library at each location for client archive operations and for Tivoli Storage Manager server database backups. You want to be able to monitor activities on all servers. You also want to designate some other users as administrators who can work with these servers.

The following sections give you an overview of the steps to take to complete this setup. For details on each step, see the section referenced.

Setting up a configuration manager

You are required to issue specific commands to set up one Tivoli Storage Manager server as a configuration manager.

Figure 93 illustrates the commands that you must issue to set up one Tivoli Storage Manager server as a configuration manager. The following procedure gives you an overview of the steps required to set up a server as a configuration manager.

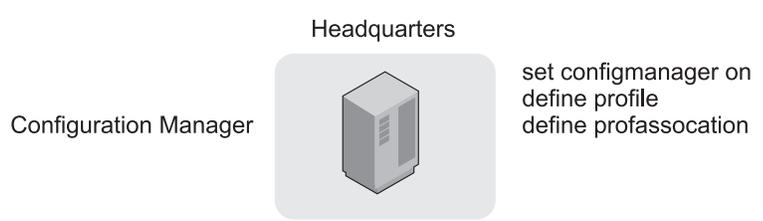


Figure 93. Setting up a configuration manager

1. Decide whether to use the existing Tivoli Storage Manager server in the headquarters office as the configuration manager or to install a new Tivoli Storage Manager server on a system.
2. Set up the communications among the servers.
3. Identify the server as a configuration manager.

Use the following command:

```
set configmanager on
```

This command automatically creates a profile named `DEFAULT_PROFILE`. The default profile includes all the server and server group definitions on the configuration manager. As you define new servers and server groups, they are also associated with the default profile.

4. Create the configuration to distribute.

The tasks that might be involved include:

- Register administrators and grant authorities to those that you want to be able to work with all the servers.
- Define policy objects and client schedules
- Define administrative schedules
- Define Tivoli Storage Manager server scripts
- Define client option sets
- Define servers
- Define server groups

Example 1: You need a shorthand way to send commands to different groups of managed servers. You can define server groups. For example, you can define a server group named `AMERICAS` for the servers in the offices in North America and South America.

Example 2: You want each managed server to back up its database and storage pools regularly. One way to do this is to set up Tivoli Storage Manager server scripts and schedules to automatically run these scripts everyday. You can do the following:

- Verify or define server scripts that perform these operations.
- Verify or define administrative command schedules that run these scripts.

Example 3: You want clients to back up data to the default disk storage pool, `BACKUPPOOL`, on each server. But you want clients to archive data directly to the tape library attached to each server. You can do the following:

- In the policy domain that you will point to in the profile, update the archive copy group so that `TAPEPOOL` is the name of the destination storage pool.
- On each server that is to be a managed server, ensure that you have a tape storage pool named `TAPEPOOL`.

Note: You must set up the storage pool itself (and associated device class) on each managed server, either locally or by using command routing. If a managed server already has a storage pool associated with the automated tape library, you can rename the pool to `TAPEPOOL`.

Example 4: You want to ensure that client data is consistently backed up and managed on all servers. You want all clients to be able to store three backup versions of their files. You can do the following:

- Verify or define client schedules in the policy domain so that clients are backed up on a consistent schedule.
- In the policy domain that you will point to in the profile, update the backup copy group so that three versions of backups are allowed.

- Define client option sets so that basic settings are consistent for clients as they are added.
5. Define one or more profiles.
- For example, you can define one profile named ALLOFFICES that points to all the configuration information (policy domain, administrators, scripts, and so on). You can also define profiles for each type of information, so that you have one profile that points to policy domains, and another profile that points to administrators, for example.
- See “Setting up communications among servers” on page 676 for details. For more information, see “Creating the default profile on a configuration manager” on page 689. See “Defining a server group and members of a server group” on page 710 for details. For details, see “Creating and changing configuration profiles” on page 690.

Setting up a managed server

Setting up the managed server can be done by an administrator working at a central location, or by administrators working at the servers that will be managed servers.

Figure 94 shows the specific commands needed to set up one Tivoli Storage Manager server as a managed server. The following procedure gives you an overview of the steps required to set up a server as a managed server.

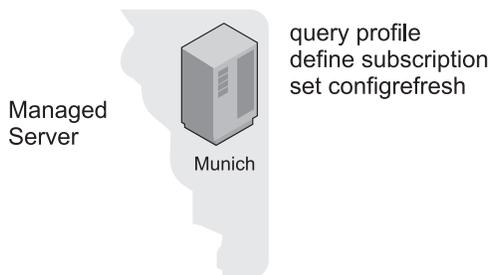


Figure 94. Setting up a managed server

A server becomes a managed server when that server first subscribes to a profile on a configuration manager.

1. Query the server to look for potential conflicts.

Look for definitions of objects on the managed server that have the same name as those defined on the configuration manager. With some exceptions, these objects will be overwritten when the managed server first subscribes to the profile on the configuration manager.

If the managed server is a new server and you have not defined anything, the only objects you will find are the defaults (for example, the STANDARD policy domain).

2. Subscribe to one or more profiles.

A managed server can only subscribe to profiles on one configuration manager.

If you receive error messages during the configuration refresh, such as a local object that could not be replaced, resolve the conflict and refresh the configuration again. You can either wait for the automatic refresh period to be reached, or kick off a refresh by issuing the SET CONFIGREFRESH command, setting or resetting the interval.

3. If the profile included policy domain information, activate a policy set in the policy domain, add or move clients to the domain, and associate any required schedules with the clients.

You may receive warning messages about storage pools that do not exist, but that are needed for the active policy set. Define any storage pools needed by the active policy set, or rename existing storage pools.

4. If the profile included administrative schedules, make the schedules active. Administrative schedules are not active when they are distributed by a configuration manager. The schedules do not run on the managed server until you make them active on the managed server. See “Tailoring schedules” on page 573.

5. Set how often the managed server contacts the configuration manager to update the configuration information associated with the profiles.

The initial setting for refreshing the configuration information is 60 minutes.

For more information, see the following topics:

- “Associating configuration information with a profile” on page 690
- “Defining storage pools” on page 227
- “Getting information about profiles” on page 697
- “Refreshing configuration information” on page 703
- “Renaming storage pools” on page 357
- “Subscribing to a profile” on page 699

Creating the default profile on a configuration manager

To set up one Tivoli Storage Manager server as the source for configuration information for other servers, identify the server as a configuration manager. A configuration manager can be an existing Tivoli Storage Manager server that already provides services to clients, or can be a server dedicated to just providing configuration information to other Tivoli Storage Manager servers.

Task	Required Privilege Class
Set up a server as a configuration manager	System

Issue the following command:

```
set configmanager on
```

When a server becomes a configuration manager, the server automatically creates a default profile named DEFAULT_PROFILE. The default profile contains any definitions of servers and server groups that exist on the configuration manager. You can change or delete the profile named DEFAULT_PROFILE.

When a managed server first subscribes to a profile on a configuration manager, the configuration manager automatically also subscribes the managed server to the profile named DEFAULT_PROFILE, if it exists. The information distributed via this profile gets refreshed in the same way as other profiles. This helps ensure that all servers have a consistent set of server and server group definitions for all servers in the network.

If you do not change the DEFAULT_PROFILE, whenever a managed server subscribed to the DEFAULT_PROFILE profile refreshes configuration information, the managed server receives definitions for all servers and server groups that exist on the configuration manager at the time of the refresh. As servers and server

groups are added, deleted, or changed on the configuration manager, the changed definitions are distributed to subscribing managed servers.

Creating and changing configuration profiles

You create configuration profiles on a configuration manager, which distributes the information associated with the profiles to any managed server that subscribes to those profiles.

Creating a configuration profile includes the following steps:

1. Defining the profile
2. Associating the configuration information with the profile

After you define the profile and its associations, a managed server can subscribe to the profile and obtain the configuration information.

After you define a profile and associate information with the profile, you can change the information later. While you make changes, you can lock the profiles to prevent managed servers from refreshing their configuration information. To distribute the changed information associated with a profile, you can unlock the profile, and either wait for each managed server to refresh its configuration to get the changed information or notify each managed server to refresh its configuration. The following sections provide information on each of these tasks.

Defining the profile

When you define the profile, you select the name and can include a description.

Task	Required Privilege Class
Define profiles	System

For example, to define a profile named ALLOFFICES, issue the following command:

```
define profile alloffices  
  description='Configuration to be used by all offices'
```

Associating configuration information with a profile

After you define a profile, you associate the configuration information that you want to distribute via that profile.

Task	Required Privilege Class
Define profile associations	System

You can associate the following configuration information with a profile:

- Tivoli Storage Manager administrators, including their authorities.
- Policy domains.
- Servers definitions.
- Server groups.
- Administrative command schedules.
- Tivoli Storage Manager server scripts.
- Client option sets.

Before you can associate specific configuration information with a profile, the definitions must exist on the configuration manager. For example, to associate a policy domain named ENGDOMAIN with a profile, you must have already defined the ENGDOMAIN policy domain on the configuration manager.

Suppose you want the ALLOFFICES profile to distribute policy information from the STANDARD and ENGDOMAIN policy domains on the configuration manager. Enter the following command:

```
define profassociation alloffices domains=standard,engdomain
```

You can make the association more dynamic by specifying the special character, * (asterisk), by itself. When you specify the *, you can associate all existing objects with a profile without specifically naming them. If you later add more objects of the same type, the new objects are automatically distributed via the profile. For example, suppose that you want the ADMINISTRATORS profile to distribute all administrators registered to the configuration manager. Enter the following commands on the configuration manager:

```
define profile administrators
  description='Profile to distribute administrators IDs'

define profassociation administrators admins=*
```

Whenever a managed server that is subscribed to the ADMINISTRATORS profile refreshes configuration information, it receives definitions for all administrators that exist on the configuration manager at the time of the refresh. As administrators are added, deleted, or changed on the configuration manager, the changed definitions are distributed to subscribing managed servers.

For more information, see the following topics:

- “Configuring information for administrative command schedules” on page 694
- “Configuring information for policy domains” on page 692
- “Configuring information for servers and server groups” on page 693
- “Configuration information for Tivoli Storage Manager administrators”
- “IBM Tivoli Storage Manager server scripts” on page 578
- “Managing client option sets” on page 413

Configuration information for Tivoli Storage Manager administrators

Be careful if you are distributing definitions of administrators that have the same name as administrators already defined to managed servers. The configuration refresh overwrites the administrator definition and authority defined on the managed server.

If the authority level of an administrator is less on the configuration manager than it was on the managed server, you might have problems with access to the managed server after distributing the administrator definition.

The configuration manager does not distribute information about whether an administrator is locked (preventing access to the server).

The administrator with the name SERVER_CONSOLE is never distributed from the configuration manager to a managed server.

For administrator definitions that have node authority, the configuration manager only distributes information such as password and contact information. Node

authority for the managed administrator can be controlled on the managed server using the GRANT AUTHORITY and REVOKE AUTHORITY commands specifying the CLASS=NODE parameter.

Configuring information for policy domains

When you point to a policy domain in a profile, the configuration information that will be sent to the managed servers includes the policy domain itself and all policy sets with their associated management classes, copy groups, and client schedules in the domain.

A configuration manager does *not* distribute the following:

- An active policy set and any of its associated management classes, copy groups, and client schedules. On each managed server, you must activate a policy set in each managed policy domain.
- Associations between clients and schedules. To have clients in a managed policy domain run client schedules, you must associate the clients with the schedules on the managed server.
- Client actions, which are schedules created by using the DEFINE CLIENTACTION command. On each managed server, you can define and delete client actions, even if the corresponding domain is a managed object.
- Definitions for any storage pools identified as destinations in the policy. Definitions of storage pools and device classes are not distributed by a configuration manager.

Policy domains can refer to storage pool names in the management classes, backup copy groups, and archive copy groups. As you set up the configuration information, consider whether managed servers already have or can set up or rename storage pools with these names.

A subscribing managed server may already have a policy domain with the same name as the domain associated with the profile. The configuration refresh overwrites the domain defined on the managed server unless client nodes are already assigned to the domain. Once the domain becomes a managed object on the managed server, you can associate clients with the managed domain. Future configuration refreshes can then update the managed domain.

If nodes are assigned to a domain with the same name as a domain being distributed, the domain is not replaced. This safeguard prevents inadvertent replacement of policy that could lead to loss of data. To replace an existing policy domain with a managed domain of the same name, perform the following steps on the managed server:

1. Copy the domain.
2. Move all clients assigned to the original domain to the copied domain.
3. Trigger a configuration refresh.
4. Activate the appropriate policy set in the new, managed policy domain.
5. Move all clients back to the original domain, which is now managed.

Configuring information for servers and server groups

The DEFAULT_PROFILE that is automatically created on a configuration manager already points to all servers and server groups defined to that server. If you leave the DEFAULT_PROFILE intact, you do not need to include servers or server groups in any other profile.

Any servers and server groups that you define later are associated automatically with the default profile and the configuration manager distributes the definitions at the next refresh. For a server definition, the following attributes are distributed:

- Communication method
- TCP/IP address (high-level address), Version 4 or Version 6
- Port number (low-level address)
- Server password
- Server URL
- The description

When server definitions are distributed, the attribute for allowing replacement is always set to YES. You can set other attributes, such as the server's node name, on the managed server by updating the server definition.

A managed server may already have a server defined with the same name as a server associated with the profile. The configuration refresh does not overwrite the local definition unless the managed server allows replacement of that definition. On a managed server, you allow a server definition to be replaced by updating the local definition. For example:

```
update server santiago allowreplace=yes
```

This safeguard prevents disruption of existing functions that require communication among servers (such as virtual volumes).

Table 61 summarizes what happens when servers or server groups being distributed have the same names as servers or server groups on the managed server.

Table 61. Results of configuration refresh with duplicate object names

Local definition (on managed server)	Object with duplicate name to be distributed	Result of configuration refresh
Server	Server	The local server definition is replaced by the distributed server definition only if an administrator for the managed server updated the local definition to allow replacement.
Server	Server group	The local server definition remains. The server group definition is not distributed.
Server group	Server	The local server group is deleted. The server definition is distributed.
Server group	Server group	The local server group definition is replaced by the distributed server group definition.

Configuring information for administrative command schedules

When the configuration manager distributes administrative schedules, the schedules are not active on the managed server. An administrator on the managed server must activate any managed schedules to have them run on the managed server.

A configuration refresh does not replace or remove any local schedules that are active on a managed server. However, a refresh can update an active schedule that is already managed by a configuration manager.

Changing a profile

You can change a profile and its associated configuration information.

Task	Required Privilege Class
Define profile associations	System
Update profiles	System

For example, if you want to add a policy domain named FILESERVERS to objects already associated with the ALLOFFICES profile, enter the following command:

```
define profassociation alloffices domains=fileservers
```

You can also delete associated configuration information, which results in removal of configuration from the managed server. Use the DELETE PROFASSOCIATION command.

On a configuration manager, you cannot directly change the names of administrators, scripts, and server groups associated with a profile. To change the name of an administrator, script, or server group associated with a profile, delete the object then define it again with a new name and associate it with the profile again. During the next configuration refresh, each managed server makes the corresponding changes in their databases.

You can change the description of the profile. Enter the following command:

```
update profile alloffices  
description='Configuration for all offices with file servers'
```

See “Removing configuration information from managed servers” on page 695 for details.

Preventing access to profiles while you make changes

If you are making changes to a profile, you might want to prevent any subscribing managed server from refreshing its configuration information until you are done. You can lock the profile to prevent access to the profile by a managed server.

Locking prevents a managed server from getting information that is incomplete because you are still making changes.

Task	Required Privilege Class
Lock and unlock profiles	System

For example, to lock the ALLOFFICES profile for two hours (120 minutes), enter the following command:

```
lock profile alloffices 120
```

You can let the lock expire after two hours, or unlock the profile with the following command:

```
unlock profile alloffices
```

Distributing changed configuration information

To distribute the changed profile, wait for each managed server to refresh its configuration to get the changed information, or notify each managed server from the configuration manager. Managed servers refresh profile information on a configuration refresh period.

Task	Required Privilege Class
Notify servers that subscribe to profiles to refresh configuration information	System

From the configuration manager, to notify all servers that are subscribers to the ALLOFFICES profile, enter the following command:

```
notify subscribers profile=alloffices
```

The managed servers then refresh their configuration information, even if the time period for refreshing the configuration has not passed.

See “Refreshing configuration information” on page 703 for how to set this period.

Removing configuration information from managed servers

To remove configuration information from managed servers, you can delete the association of the object with the profile, or delete the object itself from the configuration manager.

Task	Required Privilege Class
Delete profile associations	System

Note: To remove all configuration information that is defined in the database of a managed server as a result of a profile subscription, you must delete the subscription using the option to discard all managed objects.

On the configuration manager, you can delete the association of objects with a profile. For example, you may want to remove some of the administrators that are associated with the ADMINISTRATORS profile. With an earlier command, you had included all administrators defined on the configuration manager (by specifying `ADMINS=*`). To change the administrators included in the profile you must first delete the association of all administrators, then associate just the administrators that you want to include. Do the following:

1. Before you make these changes, you may want to prevent any servers from refreshing their configuration until you are done. Enter the following command:

```
lock profile administrators
```

2. Now make the change by entering the following commands:

```
delete profassociation administrators admins=*
```

```
define profassociation administrators  
admins=admin1,admin2,admin3,admin4
```

3. Unlock the profile:

```
unlock profile administrators
```

4. You may want to notify any managed server that subscribes to the profile so that servers refresh their configuration information:

```
notify subscribers profile=administrators
```

When you delete the association of an object with a profile, the configuration manager no longer distributes that object via the profile. Any managed server subscribing to the profile deletes the object from its database when it next contacts the configuration manager to refresh configuration information. However, a managed server does not delete the following objects:

- An object that is associated with another profile to which the server subscribes.
- A policy domain that has client nodes still assigned to it. To delete the domain, you must assign the affected client nodes to another policy domain on the managed server.
- An administrator that currently has a session open with the server.
- An administrator that is the last administrator with system authority on the managed server.

Also the managed server does not change the authority of an administrator if doing so would leave the managed server without any administrators having the system privilege class.

You can avoid both problems by ensuring that you have locally defined at least one administrator with system privilege on each managed server.

- An administrative schedule that is active. To remove an active schedule, you must first make the schedule inactive on the managed server.
- A server definition for a server that currently has an open connection from the managed server.
- A server definition that is specified in the definition of a device class that is a SERVER device type.
- A server definition that is the definition for the event server for the managed server.

If you no longer need an object defined on the configuration manager itself or on any managed server, you can delete the object itself. Deleting the object itself from the configuration manager has an effect similar to deleting the association of that object with the profile. The configuration manager no longer distributes that object, and a managed server attempts to delete the object from its database when it refreshes configuration information.

See “Deleting subscriptions” on page 702.

Deleting profiles

You can delete a profile from a configuration manager. Before deleting a profile, you should ensure that no managed server still has a subscription to the profile. If the profile still has some subscribers, delete the subscriptions on each managed server first.

Task	Required Privilege Class
Delete profiles	System

When you delete subscriptions, consider whether you want the managed objects to be deleted on the managed server at the same time. For example, to delete the

subscription to profile ALLOFFICES from managed server SANTIAGO without deleting the managed objects, log on to the SANTIAGO server and enter the following command:

```
delete subscription alloffices
```

Then, on the configuration manager, enter the following command:

```
delete profile alloffices
```

Note: You can use command routing to issue the DELETE SUBSCRIPTION command for all managed servers.

If you try to delete a profile, that still has subscriptions, the command fails unless you force the operation:

```
delete profile alloffices force=yes
```

If you do force the operation, managed servers that still subscribe to the deleted profile will later contact the configuration manager to try to get updates to the deleted profile. The managed servers will continue to do this until their subscriptions to the profile are deleted. A message will be issued on the managed server alerting the administrator of this condition.

See “Deleting subscriptions” on page 702 for more details about deleting subscriptions on a managed server.

Getting information about profiles

You can get information about configuration profiles defined on any configuration manager, as long as that server is defined to the server with which you are working.

Task	Required Privilege Class
Request information about profiles	Any administrator

For example, from a configuration manager, you can display information about profiles defined on that server or on another configuration manager. From a managed server, you can display information about any profiles on the configuration manager to which the server subscribes. You can also get profile information from any other configuration manager defined to the managed server, even though the managed server does not subscribe to any of the profiles.

For example, to get information about all profiles on the HEADQUARTERS configuration manager when logged on to another server, enter the following command:

```
query profile server=headquarters
```

The following shows what the results might look like:

Configuration manager	Profile name	Locked?
-----	-----	-----
HEADQUARTERS	ADMINISTRATORS	No
HEADQUARTERS	DEFAULT_PROFILE	No
HEADQUARTERS	ENGINEERING	No
HEADQUARTERS	MARKETING	No

You may need to get detailed information about profiles and the objects associated with them, especially before subscribing to a profile. You can get the names of the objects associated with a profile by entering the following command:

```
query profile server=headquarters format=detailed
```

The following shows what the results might look like:

```
Configuration manager: HEADQUARTERS
  Profile name: ADMINISTRATORS
  Locked?: No
  Description:
  Server administrators: ADMIN1 ADMIN2 ADMIN3 ADMIN4
  Policy domains:
Administrative command schedules: ** all objects **
Server Command Scripts:
  Client Option Sets:
  Servers:
  Server Groups:

Configuration manager: HEADQUARTERS
  Profile name: DEFAULT_PROFILE
  Locked?: No
  Description:
  Server administrators:
  Policy domains:
Administrative command schedules:
Server Command Scripts:
  Client Option Sets:
  Servers: ** all objects **
  Server Groups: ** all objects **

Configuration manager: HEADQUARTERS
  Profile name: ENGINEERING
  Locked?: No
  Description:
  Server administrators:
  Policy domains: ENGDOMAIN
Administrative command schedules:
Server Command Scripts: QUERYALL
  Client Option Sets: DESIGNER PROGRAMMER
  Servers:
  Server Groups:

Configuration manager: HEADQUARTERS
  Profile name: MARKETING
  Locked?: Yes
  Description:
  Server administrators:
  Policy domains: MARKETDOM
Administrative command schedules:
Server Command Scripts: QUERYALL
  Client Option Sets: BASIC
  Servers:
  Server Groups:
```

If the server from which you issue the query is already a managed server (subscribed to one or more profiles on the configuration manager being queried), by default the query returns profile information as it is known to the managed server. Therefore the information is accurate as of the last configuration refresh done by the managed server. You may want to ensure that you see the latest version of profiles as they currently exist on the configuration manager. Enter the following command:

```
query profile uselocal=no format=detailed
```

To get more than the names of the objects associated with a profile, you can do one of the following:

- If command routing is set up between servers, you can route query commands from the server to the configuration manager. For example, to get details on the ENGDOMAIN policy domain on the HEADQUARTERS server, enter this command:

```
headquarters: query domain engdomain format=detailed
```

You can also route commands from the configuration manager to another server to get details about definitions that already exist.

- If command routing is not set up, log on to the configuration manager and enter the query commands to get the information you need.

Subscribing to a profile

After an administrator at a configuration manager has created profiles and associated objects with them, managed servers can subscribe to one or more of the profiles.

Task	Required Privilege Class
Define subscriptions to profiles	System
Set the period for configuration refreshes	System

Note:

- Unless otherwise noted, the commands in this section would be run on a managed server:
- An administrator at the managed server could issue the commands.
- You could log in from the enterprise console and issue them.
- If command routing is set up, you could route them from the server that you are logged in to.

After a managed server subscribes to a profile, the configuration manager sends the object definitions associated with the profile to the managed server where they are automatically stored in the database. Object definitions created this way in the database of a managed server are called managed objects. With a few exceptions, you cannot change managed objects on the managed server. The exceptions are that you can change:

- The active status of a schedule
- The lock status of an administrator
- Which policy set is active in a policy domain
- The default management class of a policy set
- The attributes of a server definition that are related to the use of virtual volumes (node name, password, and delete grace period)

Before a managed server subscribes to a profile, be aware that if you have defined any object with the same name and type as an object associated with the profile that you are subscribing to, those objects will be overwritten. You can check for such occurrences by querying the profile before subscribing to it.

When a managed server first subscribes to a profile on a configuration manager, it also automatically subscribes to DEFAULT_PROFILE, if a profile with this name is defined on the configuration manager. Unless DEFAULT_PROFILE is modified on the configuration manager, it contains all the server definitions and server groups

defined on the configuration manager. In this way, all the servers in your network receive a consistent set of server and server group definitions.

Note: Although a managed server can subscribe to more than one profile on a configuration manager, it cannot subscribe to profiles on more than one configuration manager at a time.

Changes can be made to a profile, after a managed server subscribes to it. An administrator on the configuration manager can notify your server of a change by issuing the NOTIFY SUBSCRIBERS command. The configuration manager contacts each managed server having a subscription to one of the specified profiles. When a managed server is contacted, it begins refresh processing to get the configuration updates from the configuration manager.

Subscription scenario

The scenario that is documented is a typical one, where a server subscribes to a profile on a configuration manager, in this case HEADQUARTERS.

In this scenario an administrator for the HEADQUARTERS server has defined three profiles, ADMINISTRATORS, ENGINEERING, and MARKETING, each with its own set of associations. In addition, DEFAULT_PROFILE was automatically defined and contains only the server and server group definitions defined on the HEADQUARTERS server. An administrator for HEADQUARTERS has given you the names of the profiles that you should be using. To subscribe to the ADMINISTRATORS and ENGINEERING profiles and keep them current, perform the following steps:

1. Display the names of the objects in the profiles on HEADQUARTERS.

You might want to perform this step to see if the object names on the profiles are used on your server for any objects of the same type. Issue this command:

```
query profile * server=headquarters format=detailed
```

You might want to get detailed information on some of the objects by issuing specific query commands on either your server or the configuration manager.

Note: If any object name matches and you subscribe to a profile containing an object with the matching name, the object on your server will be replaced, with the following exceptions:

- A policy domain is not replaced if the domain has client nodes assigned to it.
- An administrator with system authority is not replaced by an administrator with a lower authority level if the replacement would leave the server without a system administrator.
- The definition of a server is not replaced unless the server definition on the managed server allows replacement.
- A server with the same name as a server group is not replaced.
- A locally defined, active administrative schedule is not replaced

2. Subscribe to the ADMINISTRATORS and ENGINEERING profiles.

After the initial subscription, you do not have to specify the server name on the DEFINE SUBSCRIPTION commands. If at least one profile subscription already exists, any additional subscriptions are automatically directed to the same configuration manager. Issue these commands:

```
define subscription administrators server=headquarters
```

```
define subscription engineering
```

The object definitions in these profiles are now stored on your database. In addition to ADMINISTRATORS and ENGINEERING, the server is also subscribed by default to DEFAULT_PROFILE. This means that all the server and server group definitions on HEADQUARTERS are now also stored in your database.

3. Set the time interval for obtaining refreshed configuration information from the configuration manager.

If you do not perform this step, your server checks for updates to the profiles at start up and every 60 minutes after that. Set up your server to check HEADQUARTERS for updates once a day (every 1440 minutes). If there is an update, HEADQUARTERS sends it to the managed server automatically when the server checks for updates.

```
set configrefresh 1440
```

Note: You can initiate a configuration refresh from a managed server at any time. To initiate a refresh, simply reissue the SET CONFIGREFRESH with any value greater than 0. The simplest approach is to use the current setting:

```
set configrefresh 1440
```

Querying subscriptions

From time to time you might want to view the profiles to which a server is subscribed. You might also want to view the last time that the configuration associated with that profile was successfully refreshed on your server.

Task	Required Privilege Class
Request information about subscriptions	Any administrator
Request information about profiles	Any administrator

The QUERY SUBSCRIPTION command gives you this information. You can name a specific profile or use a wildcard character to display all or a subset of profiles to which the server is subscribed. For example, the following command displays ADMINISTRATORS and any other profiles that begin with the string "ADMIN":

```
query subscription admin*
```

Here is a sample of the output:

```

Configuration      Profile name      Last update
manager            -----
-----
HEADQUARTERS      ADMINISTRATORS   06/04/2002 17:51:49
HEADQUARTERS      ADMIN_1          06/04/2002 17:51:49
HEADQUARTERS      ADMIN_2          06/04/2002 17:51:49

```

To see what objects the ADMINISTRATORS profile contains, use the following command:

```
query profile administrators uselocal=no format=detailed
```

You will see output similar to the following:

```

Configuration manager: HEADQUARTERS
Profile name: ADMINISTRATORS
Locked?: No
Description:
Server administrators: ADMIN1 ADMIN2 ADMIN3 ADMIN4
Policy domains:
Administrative command schedules: ** all objects **
Server Command Scripts:
Client Option Sets:
Servers:
Server Groups:

```

Managed objects are stored in the database of a managed server as a result of subscriptions to profiles on a configuration manager. Any object that was created or updated in the database of the managed server as a result of a subscription has the string `$$CONFIG_MANAGER$$` in place of the name of the administrator who last changed the object. For example, if the policy domain named `ENGDOMAIN` is a managed object and you enter this command on the managed server:

```
query domain engdomain format=detailed
```

You will see output similar to the following:

```

Policy Domain Name: ENGDOMAIN
Activated Policy Set:
Activation Date/Time:
Days Since Activation:
Activated Default Mgmt Class:
Number of Registered Nodes: 0
Description: Policy for design and software engineers
Backup Retention (Grace Period): 30
Archive Retention (Grace Period): 365
Last Update by (administrator): $$CONFIG_MANAGER$$
Last Update Date/Time: 06/04/2002 17:51:49
Managing profile: ENGINEERING

```

The field `Managing profile` shows the profile to which the managed server subscribes to get the definition of this object.

Deleting subscriptions

If you decide that a server no longer needs to subscribe to a profile, you can delete the subscription.

Task	Required Privilege Class
Delete subscriptions to profiles	System

When you delete a subscription to a profile, you can choose to discard the objects that came with the profile or keep them in your database. For example, to request that your subscription to `PROFILEC` be deleted and to keep the objects that came with that profile, issue the following command:

```
delete subscription profilec discardobjects=no
```

After the subscription is deleted on the managed server, the managed server issues a configuration refresh request to inform the configuration manager that the subscription is deleted. The configuration manager updates its database with the new information.

When you choose to delete objects when deleting the subscription, the server may not be able to delete some objects. For example, the server cannot delete a

managed policy domain if the domain still has client nodes registered to it. The server skips objects it cannot delete, but does not delete the subscription itself. If you take no action after an unsuccessful subscription deletion, at the next configuration refresh the configuration manager will again send all the objects associated with the subscription. To successfully delete the subscription, do one of the following:

- Fix the reason that the objects were skipped. For example, reassign clients in the managed policy domain to another policy domain. After handling the skipped objects, delete the subscription again.
- Delete the subscription again, except this time do not discard the managed objects. The server can then successfully delete the subscription. However, the objects that were created because of the subscription remain.

Refreshing configuration information

On a configuration manager, an administrator can make changes to configuration information that is associated with a profile. How quickly the changes get distributed to a subscribing managed server depends on the configuration refresh period set on the managed server and whether the administrator on the configuration manager sent a notification.

Task	Required Privilege Class
Set the period for configuration refreshes	System (on the managed server)
Notify servers that subscribe to profiles to refresh configuration information	System (on the configuration manager)

By default, a managed server refreshes its configuration information every 60 minutes. To cause an immediate refresh, change this period. For example, to immediately refresh the configuration and change the frequency of future refreshes to once a day, enter the following command for the managed server:

```
set configrefresh 1440
```

By issuing this command with a value greater than zero, you cause the managed server to immediately start the refresh process.

At the configuration manager, you can cause managed servers to refresh their configuration information by notifying the servers. For example, to notify subscribers to all profiles, enter the following command:

```
notify subscribers profile=*
```

The managed servers then start to refresh configuration information to which they are subscribed through profiles.

A managed server automatically refreshes configuration information when it is restarted.

Managing problems with configuration refresh

To monitor for any problems during a configuration refresh, watch the server console or activity log of the managed server. One problem that may occur is that the refresh process can skip objects. For example, a policy domain of the same name as an existing policy domain on the managed server is not distributed if the policy domain has client nodes assigned to it.

The configuration manager sends the objects that it can distribute to the managed server. The configuration manager skips (does not send) objects that conflict with local objects. If the configuration manager cannot send all objects that are associated with the profile, the managed server does not record the configuration refresh as complete. The objects that the configuration manager successfully sent are left as local instead of managed objects in the database of the managed server. The local objects left as a result of an unsuccessful configuration refresh become managed objects at the next successful configuration refresh of the same profile subscription.

See “Associating configuration information with a profile” on page 690 for details on when objects cannot be distributed.

Returning managed objects to local control

You might want to return one or more managed objects (objects distributed by a configuration manager via profiles) to local control on the managed servers. You can accomplish this from the configuration manager or from the managed servers.

To do this from the configuration manager, you do not simply delete the association of the object from the profile, because that would cause the object to be deleted from subscribing managed servers. To ensure the object remains in the databases of the managed servers as a locally managed object, you can copy the current profile, make the deletion, and change the subscriptions of the managed servers to the new profile.

For example, servers are currently subscribed to the ENGINEERING profile. The ENGDOMAIN policy domain is associated with this profile. You want to return control of the ENGDOMAIN policy domain to the managed servers. You can do the following:

1. Copy the ENGINEERING profile to a new profile, ENGINEERING_B:
`copy profile engineering engineering_b`
2. Delete the association of the ENGDOMAIN policy domain from ENGINEERING_B:
`delete profassociation engineering_b domains=engdomain`
3. Use command routing to delete subscriptions to the ENGINEERING profile:
`americas,europe,asia: delete subscription engineering
discardobjects=no`
4. Delete the ENGINEERING profile:
`delete profile engineering`
5. Use command routing to define subscriptions to the new ENGINEERING_B profile:
`americas,europe,asia: define subscription engineering_b`

To return objects to local control when working on a managed server, you can delete the subscription to one or more profiles. When you delete a subscription,

you can choose whether to delete the objects associated with the profile. To return objects to local control, you do not delete the objects. For example, use the following command on a managed server:

```
delete subscription engineering discardobjects=no
```

Setting up administrators for the servers

Include any administrators, in your profiles, to whom you want to give access to all servers in the network. These administrators must then maintain their passwords on the configuration manager.

To ensure passwords stay valid for as long as expected on all servers, set the password expiration period to the same time on all servers. One way to do this is to route a SET PASSEXP command from one server to all of the others.

Ensure that you have at least one administrator that is defined locally on each managed server with system authority. This avoids an error on configuration refresh when all administrators for a server would be removed as a result of a change to a profile on the configuration manager.

Managing problems with synchronization of profiles

In rare situations when a managed server contacts a configuration manager to refresh configuration information, the configuration manager might determine that the profile information on the two servers is not synchronized

It might appear that the configuration information is more recent on the managed server than on the configuration manager. This could occur in the following situations:

- The database on the configuration manager has been restored to an earlier time and now has configuration information from profiles that appear to be older than what the managed server has obtained.
- On the configuration manager, an administrator deleted a profile, forcing the deletion even though one or more managed servers still subscribed to the profile. The administrator redefined the profile (using the same name) before the managed server refreshed its configuration information.

If the configuration manager still has a record of the managed server's subscription to the profile, the configuration manager does not send its profile information at the next request for refreshed configuration information. The configuration manager informs the managed server that the profiles are not synchronized. The managed server then issues a message indicating this condition so that an administrator can take appropriate action. The administrator can perform the following steps:

1. If the configuration manager's database has been restored to an earlier point in time, the administrator may want to query the profile and associated objects on the managed server and then manually update the configuration manager with that information.
2. Use the DELETE SUBSCRIPTION command on the managed server to delete subscriptions to the profile that is not synchronized. If desired, you can also delete definitions of the associated objects, then define the subscription again.

It is possible that the configuration manager may not have a record of the managed server's subscription. In this case, no action is necessary. When the managed server requests a refresh of configuration information, the configuration

manager sends current profile information and the managed server updates its database with that information.

Switching a managed server to a different configuration manager

You might want to switch a managed server from one configuration manager to another to organize your policy needs.

Perform the following steps to switch a managed server:

1. Query profiles on the server that will be the new configuration manager to compare with current profiles to which the managed server subscribes.
2. On the managed server, delete all subscriptions to profiles on the current configuration manager. Remember to delete the subscription to the profile named `DEFAULT_PROFILE`. Consider whether to discard the managed objects in the database when you delete the subscriptions.

Verify that all subscriptions have been deleted by querying subscriptions.

3. Change server communications as needed. Define the server that will be the new configuration manager. You can delete the server that was formerly the configuration manager.
4. On the managed server, define subscriptions to profiles on the new configuration manager.

Deleting subscribers from a configuration manager

Under normal circumstances, you do not have to delete subscribers from a configuration manager. You only have to delete a subscription to a profile on the managed server (by using the `DELETE SUBSCRIPTION` command).

When you issue the `DELETE SUBSCRIPTION` command, the managed server automatically notifies the configuration manager of the deletion by refreshing its configuration information. As part of the refresh process, the configuration manager is informed of the profiles to which the managed server subscribes and to which it does not subscribe. If the configuration manager cannot be contacted immediately for a refresh, the configuration manager will find out that the subscription was deleted the next time the managed server refreshes configuration information.

Deleting subscribers from a configuration manager is only necessary as a way to clean up in certain unusual situations. For example, you may need to delete subscribers if a managed server goes away completely or deletes its last subscription without being able to notify the configuration manager. You then use the `DELETE SUBSCRIBER` command to delete all subscriptions for that subscriber (the managed server) from the configuration manager's database.

Renaming a managed server

You might want to rename a managed server to align your policy configuration.

To rename a managed server, perform the following steps:

1. Change the name of the managed server by using command routing or by logging on to the managed server. Use the enterprise console or use the `SET SERVERNAME` command.
2. Change the communication setup.

- a. On the configuration manager, delete the server definition with the old name.
 - b. On the configuration manager, define the server with its new name.
3. On the managed server, refresh the configuration information. You can wait for the configuration refresh period to pass, or you can reset the refresh period to cause an immediate refresh.

See “Setting the server name” on page 567 for more information before using the SET SERVERNAME command.

Performing tasks on multiple servers

To make performing tasks with multiple servers easier, Tivoli Storage Manager provides you with the Administration Center interface, command routing, and server group definitions that you can use to simplify command routing.

Working with multiple servers using the Administration Center

The Administration Center is a Web-based interface that you can use to centrally configure and manage multiple IBM Tivoli Storage Manager servers.

You can log in to the Administration Center and access all of the Tivoli Storage Manager servers and Web clients for which you have administrative authority.

See Chapter 18, “Managing servers with the Administration Center,” on page 547 for more information.

Routing commands

Command routing enables an administrator to send commands for processing to one or more servers at the same time. The output is collected and displayed at the server that issued the routed commands.

If you have set up your servers as described in “Setting up communications for command routing” on page 680, you can route Tivoli Storage Manager administrative commands to one or more servers. A system administrator can configure and monitor many different servers from a central server by using command routing.

You can route commands to one server, multiple servers, servers defined to a named group, or a combination of these servers. A routed command cannot be further routed to other servers; only one level of routing is allowed.

Each server that you identify as the target of a routed command must first be defined with the DEFINE SERVER command. If a server has not been defined, that server is skipped and the command routing proceeds to the next server in the route list.

Tivoli Storage Manager does not run a routed command on the server from which you issue the command unless you also specify that server. To be able to specify the server on a routed command, you must define the server just as you did any other server.

Commands cannot be routed from the SERVER_CONSOLE ID.

Routed commands run independently on each server to which you send them. The success or failure of the command on one server does not affect the outcome on any of the other servers to which the command was sent.

For more information on command routing and return codes generated by command processing, refer to *Administrator's Reference*.

(see "Setting up server groups" on page 710)

Routing commands to one or more servers

You can route commands to one or more servers, and to server groups. To successfully route commands to other servers, you must have the proper administrative authority on all servers that receive the command for processing.

The return codes for command routing can be one of three severities: 0, ERROR, or WARNING. See *Administrator's Reference* for a list of valid return codes and severity levels.

Routing commands to single servers:

To route a command to a single server, enter the defined server's name, a colon, and then the command to be processed.

For example, to route a QUERY STGPOOL command to the server that is named ADMIN1, enter:

```
admin1: 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 server routing information is to use parentheses around the server name, as follows:

```
(admin1) query stgpool
```

Note: When writing scripts, you must use the parentheses for server routing information.

To route a command to more than one server, separate the server names with a comma. For example, to route a QUERY OCCUPANCY command to three servers named ADMIN1, GEO2, and TRADE5 enter:

```
admin1,geo2,trade5: query occupancy
```

or

```
(admin1,geo2,trade5) query occupancy
```

The command QUERY OCCUPANCY is routed to servers ADMIN1, GEO2, and TRADE5. If a server has not been defined with the DEFINE SERVER command, that server is skipped and the command routing proceeds to the next server in the route list.

The routed command output of each server is displayed in its entirety at the server that initiated command routing. In the previous example, output for ADMIN1 would be displayed, followed by the output of GEO2, and then the output of TRADE5.

Processing of a command on one server does not depend upon completion of the command processing on any other servers in the route list. For example, if GEO2

server does not successfully complete the command, the TRADE5 server continues processing the command independently.

Routing commands to server groups:

A server group is a named group of servers. After you set up the groups, you can route commands to the groups.

To route a QUERY STGPOOL command to the server group WEST_COMPLEX, enter:

```
west_complex: query stgpool
```

or

```
(west_complex) query stgpool
```

The QUERY STGPOOL command is sent for processing to servers BLD12 and BLD13 which are members of group WEST_COMPLEX.

To route a QUERY STGPOOL command to two server groups WEST_COMPLEX and NORTH_COMPLEX, enter:

```
west_complex,north_complex: query stgpool
```

or

```
(west_complex,north_complex) query stgpool
```

The QUERY STGPOOL command is sent for processing to servers BLD12 and BLD13 which are members of group WEST_COMPLEX, and servers NE12 and NW13 which are members of group NORTH_COMPLEX.

See “Setting up server groups” on page 710 for how to set up a server group.

Routing commands to single servers and server groups:

You can route commands to multiple single servers and to server groups at the same time.

For example, to route the QUERY DB command to servers HQSRV, REGSRV, and groups WEST_COMPLEX and NORTH_COMPLEX, enter:

```
hqsrv,regsrv,west_complex,north_complex: query db
```

or

```
(hqsrv,regsrv,west_complex,north_complex) query db
```

The QUERY DB command is sent for processing to servers HQSRV, REGSRV, to BLD12 and BLD13 (both members of WEST_COMPLEX), and to NE12 and NW12 (both members of NORTH_COMPLEX).

Duplicate references to servers are removed in processing. For example, if you route a command to server BLD12 and to server group WEST_COMPLEX (which includes BLD12), the command is sent only once to server BLD12.

Setting up server groups

You can make command routing more efficient by creating one or more server groups and adding servers to them. You can then route commands to server groups in addition to, or in place of, routing commands to single servers.

To use server groups, you must perform the following tasks:

1. Define the server groups.
2. Add the servers as members of the appropriate group.

After you have the server groups set up, you can manage the groups and group members.

Defining a server group and members of a server group

You can define groups of servers to which you can then route commands. The commands are routed to all servers in the group.

Task	Required Privilege Class
Define a server group	System
Define a server group member	System

To route commands to a server group you must perform the following steps:

1. Define the server with the DEFINE SERVER command if it is not already defined.
2. Define a new server group with the DEFINE SERVERGROUP command. Server group names must be unique because both groups and server names are allowed for the routing information.
3. Define servers as members of a server group with the DEFINE GRPMEMBER command.

The following example shows how to create a server group named WEST_COMPLEX, and define servers BLD12 and BLD13 as members of the WEST_COMPLEX group:

```
define servergroup west_complex  
define grpmember west_complex bld12,bld13
```

(see “Setting up communications for command routing” on page 680)

Managing server groups

You can query, copy, rename, update, and delete server groups as necessary.

Task	Required Privilege Class
Query a server group	System
Copy a server group	System
Rename a server group	System
Update a server group description	System
Delete a server group	System

Querying a server group:

You can obtain information about server groups using the `QUERY SERVERGROUP` command.

To query server group `WEST_COMPLEX`, enter:

```
query servergroup west_complex
```

The following is sample output from a `QUERY SERVERGROUP` command:

Server Group	Members	Description	Managing profile
WEST_COMPLEX	BLD12, BLD13		

Copying a server group:

You can copy a server group using the `COPY SERVERGROUP` command.

To copy the entire server group contents of `WEST_COMPLEX` to a different server group named `NEWWEST`, enter:

```
copy servergroup west_complex newwest
```

This command creates the new group. If the new group already exists, the command fails.

Renaming a server group:

You can rename a server group using the `RENAME SERVERGROUP` command.

To rename an existing server group `NORTH_COMPLEX` to `NORTH`, enter:

```
rename servergroup north_complex north
```

Updating a server group description:

You can update a server group using the `UPDATE SERVERGROUP` command.

To update the `NORTH` server group to modify its description, enter:

```
update servergroup north description="Northern marketing region"
```

Deleting a server group:

You can delete a server group using the `DELETE SERVERGROUP` command.

To delete `WEST_COMPLEX` server group from the Tivoli Storage Manager server, enter:

```
delete servergroup west_complex
```

This command removes all members from the server group. The server definition for each group member is not affected. If the deleted server group is a member of other server groups, the deleted group is removed from the other groups.

Managing group members

You can move and delete group members from a previously defined group.

Task	Required Privilege Class
Move a group member to another group	System
Delete a group member	

Moving a group member to another group:

You can move group members to another group using the MOVE GRPMEMBER command.

To move group member TRADE5 from the NEWWEST group to the NORTH_COMPLEX group, enter:

```
move grpmember trade5 newwest north_complex
```

Deleting a group member from a group:

You can delete group members from a group using the DELETE GROUPMEMBER command.

To delete group member BLD12 from the NEWWEST server group, enter:

```
delete grpmember newwest bld12
```

When you delete a server, the deleted server is removed from any server groups of which it was a member.

Querying server availability

You can test a connection from your local server to a specified server using the PING SERVER command.

To ping the server GEO2, enter:

```
ping server geo2
```

The PING SERVER command uses the user ID and password of the administrative ID that issued the command. If the administrator is not defined on the server being pinged, the ping fails even if the server may be running.

Using virtual volumes to store data on another server

You can store the results of database backups and other items on another server as a virtual volume.

Tivoli Storage Manager allows a server (a *source server*) to store these items on another server (a *target server*):

- database backups
- export operations
- storage pool operations

The data is stored as *virtual volumes*, which appear to be sequential media volumes on the source server, but which are actually stored as archive files on a target server. Virtual volumes can be any of these:

- Database backups

- Storage pool backups
- Data that is backed up, archived, or space managed from client nodes
- Client data migrated from storage pools on the source server
- Any data that can be moved by EXPORT and IMPORT commands

The source server is a client of the target server, and the data for the source server is managed only by the source server. In other words, the source server controls the expiration and deletion of the files that comprise the virtual volumes on the target server. You cannot use virtual volumes when the source server and the target server reside on the same Tivoli Storage Manager server.

At the target server, the virtual volumes from the source server are seen as archive data. The source server is registered as a client node (of TYPE=SERVER) at the target server and is assigned to a policy domain. The archive copy group of the default management class of that domain specifies the storage pool for the data from the source server.

Note: If the default management class does not include an archive copy group, data cannot be stored on the target server.

You can benefit from the use of virtual volumes in the following ways:

- Smaller Tivoli Storage Manager source servers can use the storage pools and tape devices of larger Tivoli Storage Manager servers.
- For incremental database backups, virtual volumes can decrease wasted space on volumes and under-utilization of high-end tape drives.
- The source server can use the target server as an electronic vault for recovery from a disaster.

Be aware of the these items when you use virtual volumes:

- When you copy or move data from a deduplicated storage pool to a non-deduplicated storage pool that uses virtual volumes, the data is reconstructed. When you copy or move data to a deduplicated storage pool that uses virtual volumes, the data is deduplicated.
- If you use virtual volumes for database backups, you might have the following situation: SERVER_A backs up its database to SERVER_B, and SERVER_B backs up its database to SERVER_A. If this is the only way databases are backed up, if both servers are at the same location, and if a disaster occurs that location, you might have no backups with which to restore your databases.
- Under certain circumstances, inconsistencies might arise among virtual volume definitions on the source server and the archive files on the target server. You can use the RECONCILE VOLUMES command to reconcile these inconsistencies.
- If you want to enable data validation between a source and target server, enable the settings using both the DEFINE SERVER and REGISTER NODE commands. For more information see “Validating a node's data” on page 495 and *Administrator's Reference*.
- Storage space limitations on the target server affect the amount of data that you can store on that server.

Note: When you issue a DEFINE SERVER command, the source server sends a verification code to the target server. When the source server begins a session with the target server, it also sends the verification code. If the code matches what was previously stored on the target, the session is opened in read-write mode. If the

verification code is lost at the source server (for example, after a database restore), the code can be reset by issuing an UPDATE SERVER command with the FORCESYNC=YES parameter.

For details, see “Reconciling virtual volumes and archive files” on page 718.

Related concepts

“Performance limitations for virtual volume operations” on page 715

Related tasks

“Setting up source and target servers for virtual volumes”

Setting up source and target servers for virtual volumes

In the source and target relationship, the source server is defined as a client node of the target server. To set up this relationship, a number of steps must be performed at the two servers.

In the following example (illustrated in Figure 95 on page 715), the source server is named TUCSON and the target server is named MADERA.

• At Tucson site:

1. Define the target server:
 - MADERA has a TCP/IP address of 127.0.0.1:1845
 - Assign the password CALCITE to MADERA.
 - Assign TUCSON as the node name by which the source server TUCSON will be known by the target server. If no node name is assigned, the server name of the source server is used. To see the server name, you can issue the QUERY STATUS command.
2. Define a device class for the data to be sent to the target server. The device type for this device class must be SERVER, and the definition must include the name of the target server.

• At Madera site:

Register the source server as a client node. The target server can use an existing policy domain and storage pool for the data from the source server. However, you can define a separate management policy and storage pool for the source server. Doing so can provide more control over storage pool resources.

1. Use the REGISTER NODE command to define the source server as a node of TYPE=SERVER. The policy domain to which the node is assigned determines where the data from the source server is stored. Data from the source server is stored in the storage pool specified in the archive copy group of the default management class of that domain.
2. You can set up a separate policy and storage pool for the source server.
 - a. Define a storage pool named SOURCEPOOL:

```
define stgpool sourcepool autotapeclass maxscratch=20
```
 - b. Copy an existing policy domain STANDARD to a new domain named SOURCEDOMAIN:

```
copy domain standard sourcedomain
```
 - c. Assign SOURCEPOOL as the archive copy group destination in the default management class of SOURCEDOMAIN:

```
update copygroup sourcedomain standard standard type=archive destination=sourcepool
```
3. After issuing these commands, ensure that you assign the source server to the new policy domain (UPDATE NODE) and activate the policy.

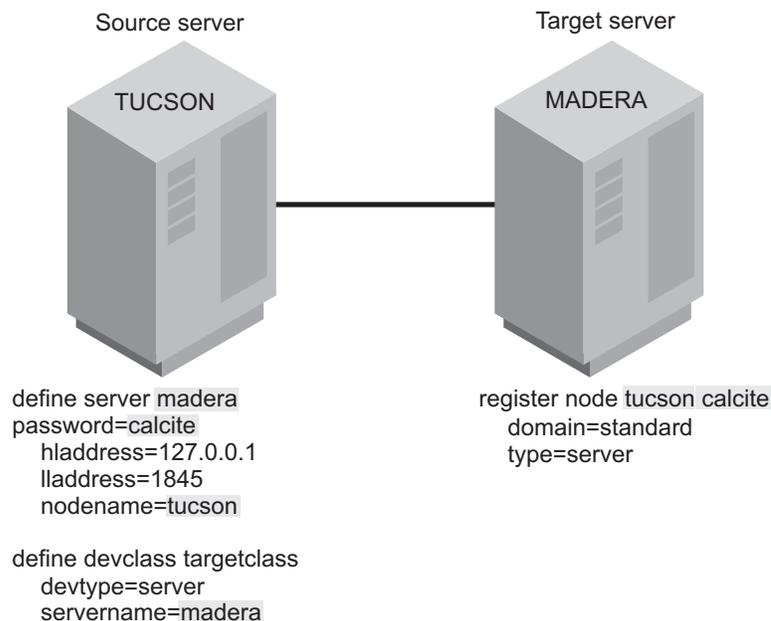


Figure 95. Communication configuration for virtual volumes

Related tasks

“Changing policy” on page 439

Performance limitations for virtual volume operations

Server-to-server virtual volume performance can vary depending on your environment and other variables, and might not be optimal for all data recovery needs.

Some of the factors that can affect volume performance when using virtual volumes are:

- Distance between locations
- Network infrastructure and bandwidth between locations
- Network configuration
- Data size and distribution
- Data read and write patterns

Given this performance variability, testing must be performed in environments that are representative of the final production configuration. In your testing, include throughput evaluations for both data-write operations (storing data from the source server to the target server), and for data-read operations (reading data from the target server to the source server, such as restoring data for a client).

Use the server-to-server virtual volumes feature to share a single tape library with multiple servers. Although there are other situations that can use this feature, such as cross-server or off-site vaulting, this feature is not optimized for long distances. For best results, use server-to-server virtual volumes for situations where the two servers (source and target) are not communicating over long distances. For example:

- The source server and target server reside within the same building.

- The source server and target server reside in different buildings but are typically covering short geographic distances such as a mile or a couple miles.
- The source server and target server reside in the same metropolitan area and are located in separate buildings 10 - 50 miles apart.

Although network technologies allow for server-to-server communications covering greater distances than discussed here, be careful when implementing a server-to-server virtual volume configuration. Network latency and other factors can significantly affect performance for long-distance implementations and might not meet recovery-time or recovery-point objectives. In these situations, consider using other technologies such as Virtual Tape Library replication, Metro Mirror, or other storage hardware approaches where tuning and bandwidth management are available to mitigate the distances involved.

Avoid moving large amounts of data between the servers, which might slow down communications significantly, depending on the network bandwidth and availability.

Specify, in the device class definition (DEVTYPE=SERVER) how often, and how long a time period you want the source server to attempt to contact the target server. Keep in mind that frequent attempts to contact the target server over an extended period can affect your communications.

To minimize mount wait times, set the total mount limit for all server definitions that specify the target server to a value that does not exceed the mount total limit at the target server. For example, a source server has two device classes, each specifying a mount limit of 2. A target server has only two tape drives. In this case, the source server mount requests might exceed the target server tape drives.

Performance can also vary by operation. For example, in some network configurations data store operations perform better than data read operations depending on how the TCPWINDOWSIZE option is specified. Test all data read operations to verify that adequate data read retrieval rates can be achieved.

Performing operations at the source server

You can perform certain operations at the source server that cause data to be stored in a storage pool at the target server.

These operations are:

- Database backups
- Storage pool backups
- Client data backup, archive, or migration
- Data migration from one storage pool to another
- Export of server information

The following sections describe how to perform these operations. In the examples, the following is assumed:

- The definitions shown in the previous section have been done.
- An operational TCP/IP connection, Version 4 or Version 6, exists between both servers.
- Both servers are running.

Backing up the database

You can back up the database of a source server to a target server.

For example, to perform an incremental backup of the source server and send the volumes to the target server, issue the following command:

```
backup db type=incremental devclass=targetclass
```

Expiration Processing of Database Backup Volumes and Recovery Plan Files with Disaster Recovery Manager: If your server uses DRM, expiration processing can delete volumes containing expired database backups and recovery plan files. One or more database backup volumes may be deleted from the volume history during expiration processing if the following conditions are true:

- The volume has a device type of SERVER
- The volume is not part of the most recent database backup series
- The last volume of the database backup series has exceeded the expiration value specified with the SET DRMDBBACKUPEXPIREDDAYS command

See “Moving copy storage pool and active-data pool volumes on-site” on page 817 for more information.

Backing up a storage pool

You can back up a storage pool of a source server to a target server.

For example, a primary storage pool named TAPEPOOL is on the source server. You can define a copy storage pool named TARGETCOPYPOOL, also on the source server. TARGETCOPYPOOL must have an associated device class whose device type is SERVER. When you back up TAPEPOOL to TARGETCOPYPOOL, the backup is sent to the target server. To accomplish this, issue the following commands:

```
define stgpool targetcopypool targetclass pooltype=copy
    maxscratch=20
backup stgpool tapepool targetcopypool
```

Storing client data on a target server

You can configure your Tivoli Storage Manager system so that when client nodes registered to the source server back up, archive, or migrate their data, that data is sent to the target server. When clients restore, retrieve, or recall their data, the source server gets the data from the target server.

To configure your system, ensure that the management policy for those nodes specifies a storage pool that has a device class whose device type is SERVER. For example, the following command defines the storage pool named TARGETPOOL.

```
define stgpool targetpool targetclass maxscratch=20
    reclaim=100
```

Note: Reclamation of a storage pool automatically begins when the percentage of reclaimable space, which is specified by the RECLAIM parameter, is reached. Reclamation of a target storage pool can involve the movement of a great deal of data from the target server to the source server and back to the target. If this operation occurs automatically during peak operating periods, it could slow network performance significantly. If you set the value to 100, reclamation will not occur automatically.

For details about storage pool reclamation and how to begin it manually, see “Reclaiming space in sequential-access storage pools” on page 319.

Migrate data from a source server storage pool to a target server storage pool

You can set up your storage pool hierarchy so that client data is migrated from a storage pool on the source server to the target server.

For example, storage pool TAPEPOOL is on the source server. The TAPEPOOL definition specifies NEXTSTGPOOL=TARGETPOOL. TARGETPOOL has been defined on the source server as a storage pool of device type SERVER. When data is migrated from TAPEPOOL, it is sent to the target server.

```
define stgpool tapepool tapeclass nextstgpool=targetpool
    maxscratch=20
```

Exporting server information to a target server

You can use any of the Tivoli Storage Manager EXPORT commands to export data from one Tivoli Storage Manager source server to sequential media on a target Tivoli Storage Manager server. You must specify a device class with a device type specified as SERVER.

For example, to copy server information directly to a target server, issue the following command:

```
export server devclass=targetclass
```

Importing server information from a target server:

If data has been exported from a source server to a target server, you can import that data from the target server to a third server. The server that will import the data uses the node ID and password of the source server to open a session with the target server. That session is in read-only mode because the third server does not have the proper verification code.

For example, to import server information from a target server, issue the following command:

```
import server devclass=targetclass
```

Reconciling virtual volumes and archive files

When you restore the database on the source or target server, you should reconcile the virtual volumes on the source server and the archive files on the target server. Reconciliation can also be used if you have any other reason to suspect inconsistencies.

To complete reconciliation, issue the RECONCILE VOLUMES command specifying a device class with the device type of SERVER. In the following example, TARGETCLASS is a server device class:

```
reconcile volumes targetclass fix=yes
```

The reconciliation action is determined by the FIX parameter as shown in Table 62 on page 719.

Table 62. FIX parameter reconciliation

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 For storage pool volumes: Mark volumes as unavailable
		Files exist but marked for deletion	Report error For storage pool volumes: 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 further verify the data. If attributes do not match, mark volumes as unavailable.
		Active files exist but attributes do not match	Report error For storage pool volumes: Mark volumes as unavailable and recommend that an AUDIT VOLUME be done to further 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

Chapter 24. Exporting and importing data

Tivoli Storage Manager provides an export and import facility that allows you to copy all or part of a server (export) so that data can be transferred to another server (import).

Two methods are available to perform the export and import operation:

- Export 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.
- Export to sequential media. Later, you can use the media to import the information to another server that has a compatible device type.

Task	Required Privilege Class
Export and import operations	System
Display information about export and import operations	Any administrator

This chapter takes you through the export and import tasks. See the following sections:

Concepts:
“Reviewing data that can be exported and imported”
Tasks for Exporting Directly to Another Server:
“Exporting data directly to another server” on page 724
“Preparing to export to another server for immediate import” on page 728
“Monitoring the server-to-server export process” on page 730
Tasks for Exporting to Sequential Media:
“Exporting and importing data using sequential media volumes” on page 732
“Exporting tasks” on page 734
“Importing data from sequential media volumes” on page 737

Reviewing data that can be exported and imported

Administrators can export or import server control information and file data from server storage.

See the following lists for definitions of these kinds of data:

- Server control information, which includes:
 - Administrator definitions
 - Client node definitions
 - Policy and scheduling definitions
- File data from server storage, which includes file space definitions and authorization rules. You can request that file data be exported in any of the following groupings of files:

- Active and inactive versions of backed up files, archive copies of files, and space-managed files
- Active versions of backed up files, archive copies of files, and space-managed files
- Active and inactive versions of backed up files
- Active versions of backed up files
- Archive copies of files
- Space-managed files

Exporting restrictions

The export function does have some limitations and restrictions. One restriction is that you can export information from an earlier version and release of Tivoli Storage Manager to a later version and release, but not from a later version and release to an earlier version and release.

For example, you can export from a V6.1 server to a V6.2 server, but you cannot export from V6.2 server to V6.1 server.

The following list shows the restrictions for exporting data:

- Export between servers that are at the same version and release but different fix packs might be possible. However, functional changes made in some fix packs might prevent a successful export operation. For example, you cannot export from a V6.1.3 server to a V6.1.2 server, or from a V6.1.2 server to a V6.1.1 or earlier server.
- Data can be exported from a server with retention protection enabled and will not be retention protected when imported on another server.
- You cannot export nodes of type NAS (network attached storage). Export processing will exclude these nodes.
- The EXPORT NODE and EXPORT SERVER commands will not export data from a shred pool unless you explicitly permit it by setting the ALLOWSHREDDABLE parameter to YES. If this value is specified, and the exported data includes data from shred pools, that data cannot be shredded. The server will not issue a warning if the export operation includes data from shred pools.

Related concepts

“Securing sensitive client data” on page 499

Deciding what information to export

Your decision on what information to export depends upon why you are exporting that information.

See the possible causes for exporting information below:

- To balance the workload across servers. For example, when many client nodes access the same server, users contend for communication paths, server resources, and tape mounts during a restore or retrieve operation.

To relieve a server of some work load and improve its performance, you may want to take one or all of the following actions:

- Move a group of client nodes to a target server
- Move policy definitions associated with these client nodes
- Move administrator definitions for administrators who manage these client nodes

To copy information to a second server (the target server), use the EXPORT NODE, EXPORT POLICY, and EXPORT ADMIN commands.

When you complete the import, you can delete file spaces, client nodes, policy objects, scheduling objects and administrators from the source server. This will reduce contention for server resources.

- To copy data for the purpose of installing a new server, use the EXPORT SERVER command.

Important:

1. Because results could be unpredictable, ensure that expiration, migration, backup, or archive processes are not running when the EXPORT NODE command is issued.
2. The EXPORT NODE and EXPORT SERVER commands will not export data from shred pools unless you explicitly permit it by setting the ALLOWSHREDDABLE parameter to YES. If this value is specified, and the exported data includes data from shred pools, but that data can no longer be shredded.

Related concepts

“Securing sensitive client data” on page 499

Deciding when to export

When you issue an EXPORT command, the operation runs as a background process. This process allows you to continue performing administrative tasks. In addition, users can continue to back up, archive, migrate, restore, retrieve, or recall files using the server.

If you choose to perform an export operation during normal working hours, be aware that administrators can change server definitions and users may modify files that are in server storage.

When you export to sequential media, administrators or users may modify data shortly after it has been exported, then the information copied to tape may not be consistent with data stored on the source server. If you want to export an exact point-in-time copy of server control information, you can prevent administrative and other client nodes from accessing the server.

When you export directly to another server, administrators or users may modify data shortly after it has been exported. You can decide to merge file spaces, use incremental export, or prevent administrative and other client nodes from accessing the server.

Related concepts

“Preventing administrative clients from accessing the server” on page 724

Related tasks

“Preventing client nodes from accessing the server” on page 724

Related reference

“Options to consider before exporting” on page 724

Preventing administrative clients from accessing the server

Administrators can change administrator, policy, or client node definitions during an export process. To prevent administrators from modifying these definitions, you can lock out administrator access to the server and cancel any administrative sessions before issuing an EXPORT command.

After the export process is complete, unlock administrator access.

Related tasks

“Canceling an IBM Tivoli Storage Manager session” on page 415

“Locking and unlocking administrators from the server” on page 430

Preventing client nodes from accessing the server

If client node information is exported while that client is backing up, archiving, or migrating files, the latest file copies for the client may not be exported to tape.

To prevent users from accessing the server during export operations, cancel existing client sessions. Then you can perform one of the following steps:

1. Disable server access to prevent client nodes from accessing the server.

This option is useful when you export all client node information from the source server and want to prevent all client nodes from accessing the server.

2. Lock out particular client nodes from server access.

This option is useful when you export a subset of client node information from the source server and want to prevent particular client nodes from accessing the server until the export operation is complete.

After the export operation is complete, allow client nodes to access the server again by:

- Enabling the server
- Unlocking client nodes

Exporting data directly to another server

You can export all server control information or a subset of that same information.

Specify one or more of the following export commands:

- EXPORT ADMIN
- EXPORT NODE
- EXPORT POLICY
- EXPORT SERVER

When you export data to a target server, you must specify the server name that will receive the data as an import operation.

Options to consider before exporting

There are several options to consider before you export, such as whether to merge file spaces, to have an incremental export, whether to replace definitions, which source of active client backup data to export, and the possibility of restarting your export operation.

Merging file spaces before exporting

You can merge imported client backup, archive, and space-managed files into existing file spaces and automatically skip duplicate files that may exist in the target file space on the server. Optionally, you can have new file spaces created.

If you do not want to merge file spaces, see the topic on how duplicate file spaces are managed.

Choosing to merge file spaces allows you to restart a cancelled import operation because files that were previously imported can be skipped in the subsequent import operation. This option is available when you issue an EXPORT SERVER or EXPORT NODE command.

When you merge file spaces, the server performs versioning of the imported objects based on the policy bound to the files. An import operation may leave the target file space with more versions than policy permits. Files are versioned to maintain the policy intent for the files, especially when incremental export (using the FROMDATE and FROMTIME parameters) is used to maintain duplicate client file copies on two or more servers.

The following definitions show how the server merges imported files, based on the type of object, when you specify MERGEFILESPPACES=YES.

Archive Objects

If an archive object for the imported node having the same TCP/IP address, TCP/IP port, name, insert date, and description is found to already exist on the target server, the imported object is skipped. Otherwise, the archive object is imported.

Backup Objects

If a backup object for the imported node has the same TCP/IP address, TCP/IP port, insert date, and description as the imported backup object, the imported object is skipped. When backup objects are merged into existing file spaces, versioning will be done according to policy just as it occurs when backup objects are sent from the client during a backup operation. Setting their insert dates to zero (0) will mark excessive file versions for expiration.

Otherwise, the server performs the following tasks:

- If the imported backup object has a later (more recent) insert date than an active version of an object on the target server with the same node, file space, TCP/IP address, and TCP/IP port, then the imported backup object becomes the new active copy, and the active copy on the target server is made inactive. Tivoli Storage Manager expires this inactive version based on the number of versions that are allowed in policy.
- If the imported backup object has an earlier (less recent) insert date than an active copy of an object on the target server with the same node, file space, TCP/IP address, TCP/IP port, then the imported backup object is inserted as an inactive version.
- If there are no active versions of an object with the same node, file space, TCP/IP address, and TCP/IP port on the target server, and the imported object has the same node, file space, TCP/IP address, and TCP/IP port as the versions, then:
 - An imported active object with a later insert date than the most recent inactive copy will become the active version of the file.

- An imported active object with an earlier insert date than the most recent inactive copy will be imported as an inactive version of the file
- Any imported inactive objects will be imported as other inactive versions of the object.

Space Managed Objects

If the imported node's space-managed object has the same external object ID, that is unique to each space managed object, already exists on the target server then the imported object is skipped. Otherwise, the space-managed object is imported.

The number of objects imported and skipped is displayed with the final statistics for the import operation.

Related concepts

“Managing duplicate file spaces” on page 745

Related tasks

“Querying the activity log for export or import information” on page 750

Incremental export before exporting

The system administrator can limit the file data exported to objects that were stored on the server on or after the date and time specified.

You can use the **FROMDATE** and **FROMTIME** parameters to export data based on the date and time the file was originally stored in the server. The **FROMDATE** and **FROMTIME** parameters only apply to client user file data; these parameters have no effect on other exported information such as policy. If clients continue to back up to the originating server while their data is moving to a new server, you can move the backup data that was stored on the originating server after the export operation was initiated. This option is available when you issue an **EXPORT SERVER** or **EXPORT NODE** command.

You can use the **TODATE** and **TOTIME** parameters to further limit the time you specify for your export operation.

Replace definitions before exporting

You can specify whether definitions (not file data) are replaced on the target server. If duplicate definitions exist on the target server, they can be replaced with the imported definitions.

Alternatively, you can have the server skip duplicate definitions. This option is available when you issue any of the **EXPORT** commands.

Related concepts

“Determining whether to replace existing definitions” on page 739

Sources of active client backup data

When exporting active versions of client backup data, the server searches for active file versions in an active-data storage pool associated with a FILE device class, if such a pool exists.

Related concepts

“Active-data pools as sources of active file versions for server operations” on page 225

Restartable export operations

You can restart a suspended server-to-server export operation if you define the FILEDATA value as anything other than NONE when issuing an EXPORT NODE or EXPORT SERVER command. An export might be suspended during the export operation if a failure occurs.

The resumed export continues at a point where the suspension took place. Therefore, data that has already been exported is not exported again and only the data that was not sent is included in the restarted export. Issue the QUERY EXPORT command to view all running and suspended restartable export operations, the RESTART EXPORT command to restart an export operation, or the SUSPEND EXPORT to suspend a running server-to-server EXPORT NODE or EXPORT SERVER process. Suspended server-to-server export operations are not affected by a server restart.

Note: Do not issue the CANCEL PROCESS command if you want to restart the operation at a later time. CANCEL PROCESS ends the export process and deletes all saved status.

An export operation cannot be suspended before it completes writing definitions to the target server. The export operation might be restarted from the point at which the operation fails or suspends if the export operation fails after writing the file space definitions to the target server.

If an export operation fails prior to identifying all eligible files, when the export operation is restarted it continues to identify eligible files and may export files that were backed up while the operation was suspended.

A restarted export operation will export only the data that was identified. During a suspension, some files or nodes identified for export might be deleted or might expire. To ensure that all data is exported, restart the export operation at the earliest time and restrict operations on the selected data.

A restartable server-to-server export operation goes through (in order) the following three phases:

1. Creating definitions on target server
2. Identifying and exporting eligible files
3. File list complete. Exporting eligible files

At any given time, a restartable export operation will be in one of the following states:

Running - Not Suspending

This state directly corresponds to phase 1 of a restartable export, “Creating definitions on target server.”

Running

The operation is running as an active process and is either in phase 2 on page 727: "Identifying and exporting eligible files" or phase 3 on page 727: "File list complete. Exporting eligible files."

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 can be in either phase 2 on page 727: "Identifying and exporting eligible files" or phase 3 on page 727: "File list complete. Exporting eligible files."

Suspended

The operation is not currently running. It may still be in either phase 2 on page 727 or phase 3 on page 727.

An export operation that fails while it is in phase 1 on page 727 cannot be restarted, and you must re-specify the export command. The new export operation starts and all definitions are retransmitted. Before issuing the export command, check the state of the target server to guarantee that the definitions will be properly imported during the new operation. If the original export command specified MERGEFILESPPACES=NO, delete any filespace definitions imported on the target server prior to the failure to ensure that filespace definitions are correctly imported during the new operation. If the original export command specified REPLACEDEFS=NO and you want to export any changes in the definitions made since the original operation failed, delete all definitions on the target that were imported prior to the failure.

When a server-to-server export operation restarts, the data continues to be imported onto the target server into the same file spaces that were selected in the original export process, regardless of the value of the MERGEFILESPPACES parameter. For server-to-server export operations, data is transmitted in transactions to ensure the source server can restart at the correct location in the data should the process fail. The target server's TXNGROUPMAX and MOVESIZETHRESH values are used to control the amount of data within each export transaction.

Attention: Ensure that the target server's Tivoli Storage Manager level is newer or the same as the source server's level. If you suspend export operations and upgrade the source server's database, the target server may stop the export operation if the new source server's Tivoli Storage Manager level is incompatible with the target server's level.

Preparing to export to another server for immediate import

When you export data to another server on the network, the export results in an immediate import on the target server. You can export data to a Tivoli Storage Manager server of the same or different operating system as the originating server.

A server-to-server export operation performs the following steps:

1. Opens a session with the target server.
2. Authenticates with the administrator's user ID and password.
3. Starts the equivalent of an IMPORT SERVER process.

Before you export data to another server on the network, perform the following tasks:

1. Install Tivoli Storage Manager on the target server. This includes defining disk space for the database and recovery log, and defining initial server storage. For more information, refer to *Installation Guide*.
2. Consider setting up enterprise configuration for the target server so you can distribute consistent backup and archive policies to the target server.
3. Issue the DEFINE SERVER command to define the name of the target server or the originating server.
4. Ensure that the administrator that issues the export command is defined with the same administrator name and password on the target server, and has System authority on the target server.

Related tasks

Chapter 23, “Managing a network of Tivoli Storage Manager servers,” on page 671

“Setting up communications among servers” on page 676

Previewing results of an export operation for immediate import

When you export data to another server, you can use the PREVIEWIMPORT option to determine how much data will be transferred without actually moving any data. When PREVIEWIMPORT=NO, the export operation is performed, and the data is immediately imported to the target server. This option is available when you issue any EXPORT command.

Issue each EXPORT command with PREVIEWIMPORT=YES to determine which objects and how much data will be copied to the target server. Use this information to determine how much storage pool space is required on the target server. The server sends the messages to the activity log and to the following place for each operation:

- Server console

To determine how much space is required to export all server data, issue the following command:

```
export server filedata=all previewimport=yes
```

After you issue the EXPORT SERVER command, a message similar to the following message is issued when the server starts a background process:

```
EXPORT SERVER started as Process 4
```

You can view the preview results by querying the activity log.

Related tasks

“Requesting information about an export or import process” on page 748

“Canceling server processes” on page 565

Directing import messages to an output file

The information generated by the validation process can help you define a storage hierarchy that supports the storage destinations currently defined in the import data.

You can direct import messages to an output file to capture any error messages that are detected during the import process. Do this by starting an administrative client session in console mode before you invoke the import command.

To direct messages to an output file named `IMPSEV.OUT`, issue the following command:

```
> dsmadm -consolemode -outfile=impserv.out
```

Monitoring the server-to-server export process

You can view information on the server console about an import or export process or you can view the information from an administrative client that is running in console mode.

Be watchful of mount messages, because the server might request mounts of volumes that are not in the library. Check-in of volumes may be required.

If you want to view the status of any server-to-server exports that can be suspended, issue the `QUERY EXPORT` command. The `QUERY EXPORT` command lists all running or suspended operations.

If a process completes, you can query the activity log for status information from an administrative client running in batch or interactive mode.

You can also query the activity log for status information from the server console.

The process first builds a list of what is to be exported. The process can therefore be running for some time before any data is transferred. The connection between the servers might time-out. You may need to adjust the `COMMTIMEOUT` and `IDLETIMEOUT` server options on one or both servers.

If a process completes, you can query the activity log for status information from the server console or from an administrative client running in batch or interactive mode. The process first builds a list of what is to be exported. The process can therefore be running for some time before any data is transferred. The connection between the servers might time-out. You may need to adjust the `COMMTIMEOUT` and `IDLETIMEOUT` server options on one or both servers.

Exporting administrator information to another server

When you issue the `EXPORT ADMIN` command, the server exports administrator definitions.

Each administrator definition includes:

- Administrator name, password, and contact information
- Any administrative privilege classes the administrator has been granted
- Whether the administrator ID is locked from server access

You can specify a list of administrator names, or you can export all administrator names.

You can preview the result on the server console or by querying the activity log.

Issue the following command to export all the administrator definitions to the target server defined as OTHERSERVER.

```
export admin * toserver=otherserver previewimport=yes
```

This lets you preview the export without actually exporting the data for immediate import.

Exporting client node information to another server

When you issue the EXPORT NODE command, the server exports client node definitions.

Each client node definition includes:

- User ID, password, and contact information
- Name of the policy domain to which the client is assigned
- File compression status
- Whether the user has the authority to delete backed up or archived files from server storage
- Whether the client node ID is locked from server access

You can also specify whether to export file data. File data includes file space definitions and authorization rules. You can request that file data be exported in any of the following groupings of files:

- Active and inactive versions of backed up files, archive copies of files, and space-managed files
- Active versions of backed up files, archive copies of files, and space-managed files
- Active and inactive versions of backed up files
- Active versions of backed up files
- Archive copies of files
- Space-managed files

To export client node information and all client files for NODE1 directly to SERVERB, issue the following example command:

```
export node node1 filedata=all toserver=serverb
```

Important: When you specify a list of node names or node patterns, the server will not report the node names or patterns that do not match any entries in the database. Check the summary statistics in the activity log to verify that the server exported all intended nodes.

Exporting policy information to another server

When you issue the EXPORT POLICY command, the server exports information belonging to each specified policy domain.

The following items are exported:

- 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
- Associations between client nodes and schedules

To export policy information directly to SERVERB, issue the following example command:

```
export policy replacedefs=yes toserver=serverb
```

Exporting server data to another server

When you issue the EXPORT SERVER command, the server exports all server control information. You can also export file data information with the EXPORT SERVER command.

To export server data to another server on the network and have the file spaces merged with any existing file spaces on the target server, as well as replace definitions on the target server and have the data, that is to be exported, to begin with any data inserted in the originating server beginning on 10/25/2007, issue the following command:

```
export server toserver=serv23 fromdate=10/25/2007 filedata=all  
mergefilespaces=yes dates=relative
```

Exporting and importing data using sequential media volumes

Before you export or import data, you can use the EXPORT or IMPORT command with the PREVIEW parameter to verify what data will be moved and prepare sequential media for exporting and importing data.

Using preview before exporting or importing data

You can specify PREVIEW=YES on the EXPORT and IMPORT commands to generate a report that shows how much data will be transferred without actually moving any data. When PREVIEW=NO, the export or import operation is performed.

1. To determine how much data and which objects are moved, issue both EXPORT or IMPORT commands with PREVIEW=YES.

The server sends export or import messages to the activity log and to the following place:

- Server console

Export Reports the types of objects, number of objects, and number of bytes that would be copied to sequential media volumes. Use this information to determine how many sequential media volumes you will need.

Import

Reports the number and types of objects found on the sequential media

volumes that meet your import specifications. Also reports information about any detected problems, such as corrupted data. Use this information to determine which data to move to the server and to determine if you have enough storage pool space allocated on the server.

2. To determine how much space is required to export all server data, issue the following command:

```
export server filedata=all preview=yes
```

After you issue the EXPORT SERVER command, the server starts a background process and issues a message similar to the following output:

```
EXPORT SERVER started as Process 4
```

You can view the preview results by querying the activity log or the following place:

- Server console

You can request information about the background process. If necessary, you can cancel an export or import process.

Related tasks

“Requesting information about an export or import process” on page 748

“Canceling server processes” on page 565

Planning for sequential media used to export data

To export data, you must specify a device class that supports sequential media and identify the volumes that will be used to store the exported data.

1. Select a device class.

You can query the source and target servers to select a device class on each server that supports the same device type. If you cannot find a device class on each server that supports a matching device type, define a new device class for a device type that is available to both servers.

Note:

- a. If the mount limit for the device class selected is reached when you request an export (that is, if all the drives are busy), the server automatically cancels lower priority operations, such as reclamation, to make a mount point available for the export.
 - b. You can export data to a storage pool on another server by specifying a device class whose device type is SERVER.
2. Estimate the number of removable media volumes to label.

To estimate the number of removable media volumes to label, divide the number of bytes to be moved by the estimated capacity of a volume.

You can estimate the following forms of removable media volumes:

- The number of tapes needed to store export data

For example, cartridge system tape volumes used with 3490 tape devices have an estimated capacity of 360 MB. If the preview shows that you need to transfer 720 MB of data, label at least two tape volumes before you export the data.

3. Use scratch media. The server allows you to use scratch media to ensure that you have sufficient space to store all export data. If you use scratch media, record the label names and the order in which they were mounted.

Or, use the **USEDVOLUMELIST** parameter on the export command to create a file containing the list of volumes used.

4. Label the removable media volumes.

During an import process, you must specify the order in which volumes are to be mounted.

This order must match the order in which the following media were mounted during the export process:

- tapes

To ensure that these are mounted in the correct order, label them with information that identifies the order in which they are mounted during the import process. For example, label them as DSM001, DSM002, DSM003, and so on.

When you export data, record the date and time for each labeled volume. Store this information in a safe location, because you will need the information when you import the data. Or, if you used the **USEDVOLUMELIST** parameter on the export command, save the resulting file. This file can be used on the import command volumes parameter.

Related tasks

“Using virtual volumes to store data on another server” on page 712
Chapter 9, “Defining device classes,” on page 199

Exporting tasks

You can export all server control information or a subset of server control information.

Specify one or more of the following export commands:

- EXPORT ADMIN
- EXPORT NODE
- EXPORT POLICY
- EXPORT SERVER

When you export data, you must specify the device class to which export data will be written. You must also list the volumes in the order in which they are to be mounted when the data is imported.

You can specify the **USEDVOLUMELIST** parameter to indicate the name of a file where a list of volumes used in a successful export operation will be stored. If the specified file is created without errors, it can be used as input to the **IMPORT** command on the **VOLUMENAMES=FILE:filename** parameter. This file will contain comment lines with the date and time the export was done, and the command issued to create the export.

Note: An export operation will not overwrite an existing file. If you perform an export operation and then try the same operation again with the same volume name, the file is skipped, and a scratch file is allocated. To use the same volume name, delete the volume entry from the volume history file.

Related tasks

“Planning for sequential media used to export data” on page 733

Exporting administrator information

When you issue the EXPORT ADMIN command, the server exports administrator definitions.

Each administrator definition includes:

- Administrator name, password, and contact information
- Any administrative privilege classes the administrator has been granted
- Whether the administrator ID is locked from server access

You can specify a list of administrator names, or you can export all administrator names.

Issue the following command to export definitions for the DAVEHIL and PENNER administrator IDs to the DSM001 tape volume, which the TAPECLASS device class supports, and to not allow any scratch media to be used during this export process:

```
export admin davehil,penner devclass=tapeclass  
volumenames=dsm001 scratch=no
```

Exporting client node information

When you issue the EXPORT NODE command, the server exports client node definitions.

Each client node definition includes:

- User ID, password, and contact information
- Name of the policy domain to which the client is assigned
- File compression status
- Whether the user has the authority to delete backed up or archived files from server storage
- Whether the client node ID is locked from server access

You can also specify whether to export file data. File data includes file space definitions and authorization rules. You can request that file data be exported in any of the following groupings of files:

- Active and inactive versions of backed up files, archive copies of files, and space-managed files
- Active versions of backed up files, archive copies of files, and space-managed files
- Active and inactive versions of backed up files
- Active versions of backed up files
- Archive copies of files
- Space-managed files

When exporting active versions of client backup data, the server searches for active file versions in an active-data pool associated with a FILE device class, if such a pool exists. This process minimizes the number of mounts that are required during the export process.

If you do not specify that you want to export file data, then the server only exports client node definitions.

For example, suppose that you want to perform the following steps:

- Export definitions for client nodes and file spaces in the ENGPOLDOM policy domain
- Export any active backup versions of files belonging to these client nodes
- Export this information to scratch volumes in the TAPECLASS device class

To issue this command, enter:

```
export node filespace=* domains=engpoldom
filedata=backupactive devclass=tapeclass
```

In this example, the server exports:

- Definitions of client nodes assigned to ENGPOLDOM
- File space definitions and backup authorizations for each client node in ENGPOLDOM
- Active versions of backed up files belonging to the client nodes assigned to ENGPOLDOM

Related concepts

“Active-data pools as sources of active file versions for server operations” on page 225

Exporting policy information

You must issue the EXPORT POLICY command to export policy information.

When you issue the EXPORT POLICY command, the server exports the following information belonging to each specified policy domain:

- 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
- Associations between client nodes and schedules

For example, suppose that you want to export policy and scheduling definitions from the policy domain named ENGPOLDOM. You want to use tape volumes DSM001 and DSM002, which belong to the TAPECLASS device class, but allow the server to use scratch tape volumes if necessary.

To issue this command, enter:

```
export policy engpoldom
devclass=tapeclass volumenames=dsm001,dsm002
```

Exporting server data

When you issue the EXPORT SERVER command, the server exports all server control information. You can also export file data information with the EXPORT SERVER command.

For example, you want to export server data to four defined tape cartridges, which the TAPECLASS device class supports. You want the server to use scratch volumes if the four volumes are not enough, and so you use the default of SCRATCH=YES.

To issue this command, enter:

```
export server devclass=tapeclass
volumenames=dsm001,dsm002,dsm003,dsm004 filedata=all
```

During the export process, the server exports definition information before it exports file data information. This ensures that definition information is stored on the first tape volumes. This process allows you to mount a minimum number of tapes during the import process, if your goal is to copy only control information to the target server.

In the example above, the server exports:

- Administrator definitions
- Client node definitions
- Policy domain, policy set, management class, and copy group definitions
- Schedule definitions and client node associations
- File space definitions
- File space authorization rules

Important: If you are using wildcard characters when retrieving or restoring data, file space authorization rules might prohibit data from being included. Wildcard characters might be ignored if your client access rules also use wildcard characters when the SET ACCESS command is issued. See the Backup-Archive Clients Installation and User's Guide for your platform.

- Backed up, archived, and space-managed files

When exporting active versions of client backup data, the server searches for active file versions in an active-data storage pool associated with a FILE device class, if such a pool exists.

Related concepts

“Active-data pools as sources of active file versions for server operations” on page 225

Importing data from sequential media volumes

After you understand how to import server control information and file data information, you can import any subset of data to the target server.

Before you import data to a new target server, you must:

1. Install Tivoli Storage Manager for the target server. This step includes defining disk space for the database and recovery log.

For information on installing Tivoli Storage Manager, see *Installation Guide*.

2. Define server storage for the target server.

Because each server operating system handles devices differently, server storage definitions are not exported. Therefore, you must define initial server storage for the target server. The target server must at least be able to use a drive that is compatible with the export media. This task can include defining libraries, drives, device classes, storage pools, and volumes. See the *Administrator's Guide* that applies to the target server.

After Tivoli Storage Manager is installed and set up on the target server, a system administrator can import all server control information or a subset of server control information by specifying one or more of the following import commands:

- IMPORT ADMIN
- IMPORT NODE
- IMPORT POLICY
- IMPORT SERVER

The entire process of importing all server control information and file data from tape volumes to a new target server includes:

- Previewing information before you import data
- Importing definitions
- Tailoring server storage definitions on the target server
- Importing file data

Options to consider before importing

Before you import data from sequential media you might consider whether to merge file spaces, replace existing definitions, or use a relative date when importing file data.

Merge file spaces:

You can merge imported client backup, archive, and space-managed files into existing file spaces, and automatically skip duplicate files that may exist in the target file space on the server. Optionally, you can have new file spaces created.

If you do not want to merge file spaces, look into how duplicate file spaces are managed. Choosing to merge file spaces allows you to restart a cancelled import operation since files that were previously imported can be skipped in the subsequent import operation.

When you merge file spaces, the server performs versioning of the imported objects based on the policy bound to the files. An import operation may leave the target file space with more versions than policy permits. Files are versioned to maintain the policy intent for the files, especially when incremental export (using the **FROMDATE** and **FROMTIME** parameters) is used to maintain duplicate client file copies on two or more servers.

The following definitions show how the server merges imported files, based on the type of object, when you specify **MERGEFILESPPACES=YES**.

Archive Objects

If an archive object for the imported node having the same TCP/IP address, TCP/IP port, insert date, and description is found to already exist on the target server, the imported object is skipped. Otherwise, the archive object is imported.

Backup Objects

If a backup object for the imported node has the same TCP/IP address, TCP/IP port, insert date, and description as the imported backup object, the imported object is skipped. When backup objects are merged into existing file spaces, versioning will be done according to policy just as it occurs when backup objects are sent from the client during a backup operation. Setting their insert dates to zero (0) will mark excessive file versions for expiration.

Otherwise, the server performs the following tasks:

- If the imported backup object has a later (more recent) insert date than an active version of an object on the target server with the same node, file space, TCP/IP address, and TCP/IP port, then the imported backup object becomes the new active copy. The active copy on the target server is made inactive. Tivoli Storage Manager expires this inactive version based on the number of versions that are allowed in policy.

- If the imported backup object has an earlier (less recent) insert date than an active copy of an object on the target server with the same node, file space, TCP/IP address, and TCP/IP port, then the imported backup object is inserted as an inactive version.
- If there are no active versions of an object with the same node, file space, TCP/IP address, TCP/IP port on the target server, and the imported object has the same node, TCP/IP address, TCP/IP port as the versions, then:
 - An imported active object with a later insert date than the most recent inactive copy will become the active version of the file.
 - An imported active object with an earlier insert date than the most recent inactive copy will be imported as an inactive version of the file
- Any imported inactive objects will be imported as other inactive versions of the object.

Space Managed Objects

If the imported node's space-managed object has an external file ID which already exists on the target server, then the imported object is skipped. Otherwise, the space-managed object is imported.

The number of objects imported and skipped is displayed with the final statistics for the import operation.

Related concepts

“Managing duplicate file spaces” on page 745

Related tasks

“Querying the activity log for export or import information” on page 750

Determining whether to replace existing definitions:

By using the **REPLACEDFS** parameter with the **IMPORT** command, you can specify whether to replace existing definitions on the target server when Tivoli Storage Manager encounters an object with the same name during the import process.

For example, if a definition exists for the **ENGPOLDOM** policy domain on the target server before you import policy definitions, then you must specify **REPLACEDFS=YES** to replace the existing definition with the data from the export tape.

Definitions that can be replaced include administrator, client node, policy, or schedule definitions. The default is to not replace existing definitions on the target server.

Deciding whether to use a relative date when importing file data:

When you import file data, you can keep the original creation date for backup versions and archive copies, or you can specify that the server use an adjusted date.

If you want to keep the original dates set for backup versions and archive copies, use **DATES=ABSOLUTE**, which is the default. If you use the absolute value, any files whose retention period has passed will be expired shortly after they are imported to the target server.

When you specify a relative date, the dates of the file versions are adjusted to the date of import on the target server. This is helpful when you export from a server that is in a different time zone than the target server.

Previewing information before you import data

Before you import any data to the target server, preview each IMPORT command to determine what data you want to import to the target server. You can import all or a subset of export data from tapes.

When you set PREVIEW=YES, tape operators must mount export tape volumes so that the target server can calculate the statistics for the preview.

Issue the following design to preview information for the IMPORT SERVER command:

```
import server devclass=tapeclass preview=yes  
volumenames=dsm001,dsm002,dsm003,dsm004
```

Figure 96 on page 741 shows an example of the messages sent to the activity log and the following place:

Server console

```

ANR0402I Session 3 started for administrator SERVER_CONSOLE (Server).
ANR1363I Import volume DSM001 opened (sequence number 1).
ANR0610I IMPORT SERVER started by SERVER_CONSOLE as process 2.
ANR0612I IMPORT SERVER: Reading EXPORT SERVER data from server SERV1
exported 05/07/1996 12:39:48.
ANR0639I IMPORT SERVER: Processing domain ENGPOLDOM.
ANR0640I IMPORT SERVER: Processing policy set ACTIVE in policy domain
ENGPOLDOM.
ANR0640I IMPORT SERVER: Processing policy set STANDARD in policy domain
ENGPOLDOM.
ANR0641I IMPORT SERVER: Processing management class STANDARD in domain
ENGPOLDOM, set ACTIVE.
ANR0641I IMPORT SERVER: Processing management class MCENG in domain
ENGPOLDOM, set STANDARD.
ANR0641I IMPORT SERVER: Processing management class STANDARD in domain
ENGPOLDOM, set STANDARD.
ANR0643I IMPORT SERVER: Processing archive copy group in domain
ENGPOLDOM, set ACTIVE, management class STANDARD.
ANR0643I IMPORT SERVER: Processing archive copy group in domain ENGPOLDOM,
set STANDARD, management class MCENG.
ANR0643I IMPORT SERVER: Processing archive copy group in domain ENGPOLDOM,
set STANDARD, management class STANDARD.
ANR0642I IMPORT SERVER: Processing backup copy group in domain ENGPOLDOM,
set ACTIVE, management class STANDARD.
ANR0642I IMPORT SERVER: Processing backup copy group in domain ENGPOLDOM,
set STANDARD, management class MCENG.
ANR0642I IMPORT SERVER: Processing backup copy group in domain ENGPOLDOM,
set STANDARD, management class STANDARD.
ANR0638I IMPORT SERVER: Processing administrator DAVEHIL.
ANR0638I IMPORT SERVER: Processing administrator PENNER.
ANR0635I IMPORT SERVER: Processing node TOMC.
ANR0636I IMPORT SERVER: Processing file space OS2 for node TOMC as file
space OS1.
ANR0636I IMPORT SERVER: Processing file space DRIVED for node TOMC as file
space DRIVE1.
ANR0636I IMPORT SERVER: Processing file space OS2VDISK for node TOMC as file
space OS2VDIS1.
ANR1365I Import volume DSM001 closed (end reached).
ANR1363I Import volume DSM002 opened (sequence number 2).
ANR1365I Import volume DSM002 closed (end reached).
ANR1363I Import volume DSM003 opened (sequence number 3).
ANR1365I Import volume DSM003 closed (end reached).
ANR1363I Import volume DSM004 opened (sequence number 4).
ANR1365I Import volume DSM004 closed (end reached).
ANR0617I IMPORT SERVER: Processing completed successfully.
ANR0620I IMPORT SERVER: Copied 1 domain(s).
ANR0621I IMPORT SERVER: Copied 2 policy set(s).
ANR0622I IMPORT SERVER: Copied 2 management class(es).
ANR0623I IMPORT SERVER: Copied 6 copy group(s).
ANR0625I IMPORT SERVER: Copied 2 administrator(s).
ANR0626I IMPORT SERVER: Copied 1 node definition(s).
ANR0627I IMPORT SERVER: Copied 3 file space(s), 0 archive file(s) and 462
backup file(s).
ANR0629I IMPORT SERVER: Copied 8856358 bytes of data.
ANR0611I IMPORT SERVER started by SERVER_CONSOLE as process 2 has ended.

```

Figure 96. Sample report created by issuing preview for an import server command

Use the value reported for the total number of bytes copied to estimate storage pool space needed to store imported file data.

For example, Figure 96 shows that 8 856 358 bytes of data will be imported. Ensure that you have at least 8 856 358 bytes of available space in the backup storage pools defined to the server. You can issue the `QUERY STGPOOL` and `QUERY VOLUME` commands to determine how much space is available in the server storage hierarchy.

In addition, the preview report shows that 0 archive files and 462 backup files will be imported. Because backup data is being imported, ensure that you have sufficient space in the backup storage pools used to store this backup data.

Related tasks

“Tailoring server storage definitions on the target server” on page 744

“Using preview before exporting or importing data” on page 732

Related reference

“Monitoring export and import processes” on page 748

Importing definitions

When previewing information before importing data, you must import server control information. This includes administrator definitions, client node definitions, policy domain, policy set, management class, and copy group definitions, schedule definitions, and client node associations.

However, do not import file data at this time, because some storage pools named in the copy group definitions may not exist yet on the target server.

Before you import server control information, perform the following tasks:

1. Read the following topics:
 - “Determining whether to replace existing definitions” on page 739
 - “Determining how the server imports active policy sets”
2. Start an administrative client session in console mode to capture import messages to an output file.
3. Import the server control information from specified tape volumes.

Related tasks

“Directing import messages to an output file” on page 743

“Importing server control information” on page 743

Determining how the server imports active policy sets:

When the server imports policy definitions, several objects are imported to the target server.

The following items are imported:

- Policy domain definitions
- Policy set definitions, including the ACTIVE policy set
- Management class definitions
- Backup copy group definitions
- Archive copy group definitions
- Schedule definitions defined for each policy domain
- Client node associations, if the client node definition exists on the target server

If the server encounters a policy set named ACTIVE on the tape volume during the import process, it uses a temporary policy set named \$\$ACTIVE\$\$ to import the active policy set.

If `replacedefs=yes`, a copy of the active policy set named \$\$ACTIVE\$\$ is made on the target server. The \$\$ACTIVE\$\$ policy set is updated with the definitions from the ACTIVE policy set in the export of the source. The new active policy set at the target server will contain all the management classes from the ACTIVE policy set from the export server and from the ACTIVE policy set that already exists on the importing server.

After `$$ACTIVE$$` is imported to the target server, the server activates this policy set. During the activation process, the server validates the policy set by examining the management class and copy group definitions. If any of the following conditions occur, the server issues warning messages during validation:

- The storage destinations specified in the backup and archive copy groups do not refer to defined storage pools.
- The default management class does not contain a backup or archive copy group.
- The current ACTIVE policy set contains management class names that are not defined in the policy set to be activated.
- The current ACTIVE policy set contains copy group names that are not defined in the policy set to be activated.

After each `$$ACTIVE$$` policy set has been activated, the server deletes that `$$ACTIVE$$` policy set from the target server. To view information about active policy on the target server, you can use the following commands:

- `QUERY COPYGROUP`
- `QUERY DOMAIN`
- `QUERY MGMTCLASS`
- `QUERY POLICYSET`

Results from issuing the `QUERY DOMAIN` command show the activated policy set as `$$ACTIVE$$`. The `$$ACTIVE$$` name shows you that the policy set which is currently activated for this domain is the policy set that was active at the time the export was performed.

Directing import messages to an output file:

The information generated by the validation process can help you define a storage hierarchy that supports the storage destinations currently defined in the import data.

You can direct import messages to an output file to capture any error messages that are detected during the import process. Do this by starting an administrative client session in console mode before you invoke the import command.

To direct messages to an output file named `IMPSEV.OUT`, issue the following command:

```
> dsmadm -consolemode -outfile=impserv.out
```

Importing server control information:

If you have completed the prerequisite steps, you might be ready to import the server control information.

Based on the information generated during the preview operation, you know that all definition information has been stored on the first tape volume named `DSM001`. Specify that this tape volume can be read by a device belonging to the `TAPECLASS` device class.

Issue the following command:

```
import server filedata=none devclass=tapeclass volumenames=dsm001
```

You can issue the command from an administrative client session or from the following:

server console

Tailoring server storage definitions on the target server

If you have already imported definition information, you can use the reports generated by the import process to help you tailor storage for the target server.

To tailor server storage definitions on the target server, complete the following steps:

1. Identify any storage destinations specified in copy groups and management classes that do not match defined storage pools:
 - If the policy definitions you imported included an ACTIVE policy set, that policy set is validated and activated on the target server. Error messages generated during validation include whether any management classes or copy groups refer to storage pools that do not exist on the target server. You have a copy of these messages in a file if you directed console messages to an output file.
 - Query management class and copy group definitions to compare the storage destinations specified with the names of existing storage pools on the target server.

To request detailed reports for all management classes, backup copy groups, and archive copy groups in the ACTIVE policy set, enter these commands:

```
query mgmtclass * active * format=detailed
query copygroup * active * standard type=backup format=detailed
query copygroup * active * standard type=archive format=detailed
```

2. If storage destinations for management classes and copy groups in the ACTIVE policy set refer to storage pools that are not defined, perform one of the following tasks:
 - Define storage pools that match the storage destination names for the management classes and copy groups.
 - Change the storage destinations for the management classes and copy groups. perform the following steps:
 - a. Copy the ACTIVE policy set to another policy set
 - b. Modify the storage destinations of management classes and copy groups in that policy set, as required
 - c. Activate the new policy set

Depending on the amount of client file data that you expect to import, you may want to examine the storage hierarchy to ensure that sufficient storage space is available. Storage pools specified as storage destinations by management classes and copy groups may fill up with data. For example, you may need to define additional storage pools to which data can migrate from the initial storage destinations.

Related tasks

“Directing import messages to an output file” on page 743

“Defining storage pools” on page 227

Related reference

“Defining and updating a policy set” on page 460

Importing file data information

After you have defined the appropriate storage hierarchy on the target server, you can import file data from the tape volumes. File data includes file space definitions and authorization rules.

You can request that file data be imported in any of the following groupings:

- Active and inactive versions of backed up files, archive copies of files, and space-managed files
- Active versions of backed up files, archive copies of files, and space-managed files
- Active and inactive versions of backed up files
- Active versions of backed up files
- Archive copies of files
- Space-managed files

Data being imported will not be stored in active-data pools. Use the COPY ACTIVATEDATA command to store newly imported data into an active-data pool.

Before you import file data information:

- Understand how the server handles duplicate file space names
- Decide whether to keep the original creation date for backup versions and archive copies or to import file data using an adjusted date

Managing duplicate file spaces:

When the server imports file data information, it imports any file spaces belonging to each specified client node. If a file space definition already exists on the target server for the node, the server does not replace the existing file space name.

If the server encounters duplicate file space names when it imports file data information, it creates a new file space name for the imported definition by replacing the final character or characters with a number. A message showing the old and new file space names is written to the system log and to the activity log. A message showing the old and new file space names is written to the activity log and to the following place:

- server console

For example, if the C_DRIVE and D_DRIVE file space names reside on the target server for node FRED and on the tape volume for FRED, then the server imports the C_DRIVE file space as C_DRIV1 file space and the D_DRIVE file space as D_DRIV1 file space, both assigned to node FRED.

Deciding whether to use a relative date when importing file data:

When you import file data, you can keep the original creation date for backup versions and archive copies, or you can specify that the server use an adjusted date.

Because tape volumes containing exported data might not be used for some time, the original dates defined for backup versions and archive copies may be old enough that files are expired immediately when the data is imported to the target server.

To prevent backup versions and archive copies from being expired immediately, specify `DATES=RELATIVE` on the `IMPORT NODE` or `IMPORT SERVER` commands to adjust for the elapsed time since the files were exported to tape.

For example, assume that data exported to tape includes an archive copy archived five days prior to the export operation. If the tape volume resides on the shelf for six months before the data is imported to the target server, the server resets the archival date to five days prior to the import operation.

If you want to keep the original dates set for backup versions and archive copies, use `DATES=ABSOLUTE`, which is the default. If you use the absolute value, any files whose retention period has passed will be expired shortly after they are imported to the target server.

Issuing an import server or import node command:

You can import file data, either by issuing the `IMPORT SERVER` or `IMPORT NODE` command. When you issue either of these commands, you can specify which type of files should be imported for all client nodes specified and found on the export tapes.

You can specify any of the following values to import file data:

All Specifies that all active and inactive versions of backed up files, archive copies of files, and space-managed files for specified client nodes are imported to the target server

None Specifies that no files are imported to the target server; only client node definitions are imported

Archive

Specifies that only archive copies of files are imported to the target server

Backup

Specifies that only backup copies of files, whether active or inactive, are imported to the target server

Backupactive

Specifies that only active versions of backed up files are imported to the target server

Allactive

Specifies that only active versions of backed up files, archive copies of files, and space-managed files are imported to the target server

Spacemanaged

Specifies that only files that have been migrated from a user's local file system (space-managed files) are imported

For example, suppose you want to import all backup versions of files, archive copies of files, and space-managed files to the target server. You do not want to replace any existing server control information during this import operation.

Specify the four tape volumes that were identified during the preview operation. These tape volumes can be read by any device in the `TAPECLASS` device class. To issue this command, enter:

```
import server filedata=all replacedefs=no
devclass=tapeclass volumenames=dsm001,dsm002,dsm003,dsm004
```

You can limit the import to nodes that were assigned to specific policy domains on the source server. For example, suppose you exported from the source server the data for all nodes in all domains. To import to the target server the data only for nodes that were in the ENGDOM on the source server, enter this command:

```
import node filedata=all domains=engdom devclass=tapeclass
volumenames=dsm001,dsm002,dsm003,dsm004
```

If the ENGDOM policy domain exists on the target server, the imported nodes are assigned to that domain. If ENGDOM does not exist on the target server, the imported nodes are assigned to the STANDARD policy domain. If you do not specify a domain on the IMPORT NODE command, the imported node is assigned to the STANDARD policy domain.

Importing subsets of information

You can use an IMPORT command to copy a subset of the information from export tapes to the target server. For example, if a tape was created with EXPORT SERVER, you can import only node information from the tape by using IMPORT NODE.

While the server allows you to issue any import command, data cannot be imported to the server if it has not been exported to tape. For example, if a tape is created with the EXPORT POLICY command, an IMPORT NODE command will not find any data on the tape because node information is not a subset of policy information.

See Table 63 for the commands that you can use to import a subset of exported information to a target server.

Table 63. Importing a subset of information from tapes

If tapes were created with this export command:	You can issue this import command:	You cannot issue this import command:
EXPORT SERVER	IMPORT SERVER IMPORT ADMIN IMPORT NODE IMPORT POLICY	Not applicable.
EXPORT NODE	IMPORT NODE IMPORT SERVER	IMPORT ADMIN IMPORT POLICY
EXPORT ADMIN	IMPORT ADMIN IMPORT SERVER	IMPORT NODE IMPORT POLICY
EXPORT POLICY	IMPORT POLICY IMPORT SERVER	IMPORT ADMIN IMPORT NODE

Recovering from errors during the import process

During import processing, the server might encounter invalid data due to corruption during storage on tape or in the database prior to the export operation.

If invalid data is encountered during an import operation, the server uses the default value for the new object's definition. If the object already exists, the existing parameter is not changed.

During import and export operations, the server reports on the affected objects to the activity log and also to the:

- server console

You should query these objects when the import process is complete to see if they reflect information that is acceptable.

Each time you run the IMPORT NODE or IMPORT SERVER command with the FILEDATA parameter equal to a value other than NONE, Tivoli Storage Manager creates a new file space and imports data to it. This process ensures that the current import does not overwrite data from a previous import.

A file space definition may already exist on the target server for the node. If so, an administrator with system privilege can issue the DELETE FILESPACE command to remove file spaces that are corrupted or no longer needed. For more information on the DELETE FILESPACE command, refer to the *Administrator's Reference*.

Related concepts

"Managing duplicate file spaces" on page 745

Renaming a file space:

An imported file space can have the same name as a file space that already exists on a client node. In this case, the server does not overlay the existing file space, and the imported file space is given a new system generated file space name.

This new name may match file space names that have not been backed up and are unknown to the server. In this case, you can use the RENAME FILESPACE command to rename the imported file space to the naming convention used for the client node.

Monitoring export and import processes

The server lets you monitor export or import processes while they are running or after they have completed.

You can use the following two ways to monitor export or import processes:

- You can view information about a process that is running on the server console or from an administrative client running in console mode.
- After a process has completed, you can query the activity log for status information from an administrative client running in batch or interactive mode.

Watch for mount messages, because the server might request mounts of volumes that are not in the library. The process first builds a list of what is to be exported. The process can therefore be running for some time before any data is transferred.

Check-in of volumes may be required.

Requesting information about an export or import process

After you issue an EXPORT or IMPORT command, the server starts a background process, assigns a process ID to the operation, and displays the process ID when the operation starts.

You can query an export or import process by specifying the process ID number.

For example, to request information about the EXPORT SERVER operation, which started as process 4, enter:

```
query process 4
```

If you issue a preview version of an EXPORT or IMPORT command and then query the process, the server reports the types of objects to be copied, the number

of objects to be copied, and the number of bytes to be copied.

When you export or import data and then query the process, the server displays the number and types of objects copied so far, and the total number of bytes that have been transferred, along with information on any media mount requests that may be outstanding for the process.

Related tasks

“Requesting information about server processes” on page 615

Viewing information from the server console

When you issue an EXPORT or IMPORT command, either from the server console or from an administrative client, information is displayed on the server console.

Figure 97 shows an example of the information that is displayed after issuing an EXPORT SERVER command.

```
ANR0610I EXPORT SERVER started by SERVER_CONSOLE as process 1.
ANR0639I EXPORT SERVER: Processing domain ENGPOLDOM.
ANR0640I EXPORT SERVER: Processing policy set ACTIVE in policy domain
ENGPOLDOM.
ANR0640I EXPORT SERVER: Processing policy set STANDARD in policy domain
ENGPOLDOM.
ANR0641I EXPORT SERVER: Processing management class STANDARD in domain
ENGPOLDOM, set ACTIVE.
ANR0641I EXPORT SERVER: Processing management class STANDARD in domain
ENGPOLDOM, set STANDARD.
ANR0643I EXPORT SERVER: Processing archive copy group in domain
ENGPOLDOM, set STANDARD, management class ACTIVE.
ANR0643I EXPORT SERVER: Processing archive copy group in domain
ENGPOLDOM, set STANDARD, management class STANDARD.
ANR0643I EXPORT SERVER: Processing backup copy group in domain
ENGPOLDOM, set STANDARD, management class ACTIVE.
ANR0643I EXPORT SERVER: Processing backup copy group in domain
ENGPOLDOM, set STANDARD, management class STANDARD.
ANR0604I EXPORT SERVER: No schedules were found in policy domain * for
exporting.
ANR0635I EXPORT SERVER: Processing node TOMC.
ANR0605I EXPORT SERVER: No schedule associations were found in
policy domain * for exporting.
ANR0637I EXPORT SERVER: Processing file space DRIVED for node TOMC.
ANR0637I EXPORT SERVER: Processing file space OS2 for node TOMC.
ANR0637I EXPORT SERVER: Processing file space OS2VDISK for node TOMC.
ANR0617I EXPORT SERVER: Processing completed successfully.
ANR0620I EXPORT SERVER: Copied 1 domain(s).
ANR0621I EXPORT SERVER: Copied 2 policy set(s).
ANR0622I EXPORT SERVER: Copied 2 management class(es).
ANR0623I EXPORT SERVER: Copied 4 copy group(s).
ANR0626I EXPORT SERVER: Copied 1 node definition(s).
ANR0627I EXPORT SERVER: Copied 3 file space(s), 16 archive file(s)
and 0 backup file(s).
ANR0629I EXPORT SERVER: Copied 3045632 bytes of data.
ANR0611I EXPORT SERVER started by SERVER_CONSOLE as process 1 has ended.
```

Figure 97. Sample export server output

Viewing information from an administrative client

You can use the console mode from an administrative client to monitor export or import operations or to capture processing messages to an output file.

1. To start an administrative session in console mode, issue the following command:

```
> dsmadm -consolemode
```

While the system is running in console mode, you cannot enter any administrative commands from the client session. You can, however, start another administrative client session for entering commands (for example, QUERY PROCESS) if you are using a multitasking workstation, such as AIX.

2. If you want the server to write all terminal output to a file, specify the OUTFILE option with a destination. For example, to write output to the SAVE.OUT file, enter:

```
> dsmadm -consolemode -outfile=save.out
```

For information about using the CONSOLE mode option and ending an administrative session in console mode, see the *Administrator's Reference*.

Querying the activity log for export or import information

After an export or import process has completed, you can query the activity log for status information and possible error messages.

To minimize processing time when querying the activity log for export or import information, restrict the search by specifying **EXPORT** or **IMPORT** in the **SEARCH** parameter of the QUERY ACTLOG command.

To determine how much data will be moved after issuing the preview version of the EXPORT SERVER command, query the activity log by issuing the following command:

```
query actlog search=export
```

Figure 98 on page 751 displays a sample activity log report.

Date/Time	Message
07/03/2002 10:50:28	ANR0610I EXPORT SERVER started by ADMIN as process 1.
07/03/2002 10:50:28	ANR0639I EXPORT SERVER: Processing domain ENGPOLDOM.
07/03/2002 10:50:28	ANR0640I EXPORT SERVER: Processing policy set ACTIVE in policy domain ENGPOLDOM.
07/03/2002 10:50:28	ANR0640I EXPORT SERVER: Processing policy set STANDARD in policy domain ENGPOLDOM.
07/03/2002 10:50:29	ANR0641I EXPORT SERVER: Processing management class STANDARD in domain ENGPOLDOM, set ACTIVE.
07/03/2002 10:50:29	ANR0641I EXPORT SERVER: Processing management class STANDARD in domain ENGPOLDOM, set STANDARD.
07/03/2002 10:50:29	ANR0643I EXPORT SERVER: Processing archive copy group in domain ENGPOLDOM, set STANDARD, management class ACTIVE.
07/03/2002 10:50:29	ANR0643I EXPORT SERVER: Processing archive copy group in domain ENGPOLDOM, set STANDARD, management class STANDARD.
07/03/2002 10:50:29	ANR0642I EXPORT SERVER: Processing backup copy group in domain ENGPOLDOM, set STANDARD, management class ACTIVE.
07/03/2002 10:50:29	ANR0642I EXPORT SERVER: Processing backup copy group in domain ENGPOLDOM, set STANDARD, management class STANDARD.
07/03/2002 10:50:29	ANR0604I EXPORT SERVER: No schedules were found in policy domain * for exporting.
07/03/2002 10:50:29	ANR0635I EXPORT SERVER: Processing node TOMC.
07/03/2002 10:50:29	ANR0605I EXPORT SERVER: No schedule associations were found in policy domain * for exporting.
07/03/2002 10:50:29	ANR0637I EXPORT SERVER: Processing file space DRIVED for node TOMC.
07/03/2002 10:50:29	ANR0637I EXPORT SERVER: Processing file space OS2 for node TOMC.
07/03/2002 10:50:29	ANR0637I EXPORT SERVER: Processing file space OS2VDISK for node TOMC.
07/03/2002 10:50:32	ANR0617I EXPORT SERVER: Processing completed successfully.
07/03/2002 10:50:32	ANR0620I EXPORT SERVER: Copied 1 domain(s).
07/03/2002 10:50:32	ANR0621I EXPORT SERVER: Copied 2 policy set(s).
07/03/2002 10:50:32	ANR0622I EXPORT SERVER: Copied 2 management class(es).
07/03/2002 10:50:32	ANR0623I EXPORT SERVER: Copied 4 copy group(s).
07/03/2002 10:50:32	ANR0626I EXPORT SERVER: Copied 1 node definition(s).
07/03/2002 10:50:32	ANR0627I EXPORT SERVER: Copied 3 file space(s), 16 export file(s) and 0 backup file(s).
07/03/2002 10:50:32	ANR0629I EXPORT SERVER: Copied 3045632 bytes of data.
07/03/2002 10:50:32	ANR0611I EXPORT SERVER started by ADMIN as process 1 has ended.

Figure 98. Sample activity log report on exported data

Exporting and importing data from virtual volumes

You can perform all the EXPORT and IMPORT operations to virtual volumes that are described in the sequential media topics.

Data stored as virtual volumes appear to be sequential storage pool volumes on the source server, but are actually stored as archive files on another server. Those archive files can be in random or sequential access storage pools. The EXPORT and IMPORT commands are identical to those previously shown except that the device class specified in the commands must have a device type of SERVER.

Related tasks

“Using virtual volumes to store data on another server” on page 712

“Exporting and importing data using sequential media volumes” on page 732

Part 5. Protecting the server

When implementing a Tivoli Storage Manager solution, one of the most important issues is infrastructure protection. The ability to recover from a disaster is essential. Various procedures are available to protect your Tivoli Storage Manager server, database, recovery log, and storage pools.

Chapter 25. Protecting and recovering your server

Failure or loss of the database, the recovery log, or storage pools can cause loss of client data. You can protect and, if necessary, recover your server.

See the following topics:

Concepts:
“Levels of protection”
“Storage pool protection overview” on page 756
“Database and recovery log protection overview” on page 758
“Snapshot database backup” on page 760
“Choosing when to enable data validation” on page 785
Protecting Data:
“Active log mirroring” on page 759
“Backing up storage pools” on page 760
“Backing up the server database” on page 767
“Data validation during audit volume processing” on page 784
Recovering Data:
“Recovering the server using database and storage pool backups” on page 772
“Restoring storage pool volumes” on page 780
“Auditing storage pool volumes” on page 782
“Fixing damaged files” on page 790
“Restoring a library manager database” on page 797
“Restoring a library client database” on page 798
Scenarios:
“Backup and recovery scenarios” on page 792

DRM: The disaster recovery manager (DRM) can automate some disaster recovery tasks. A note like this one identifies those tasks.

Levels of protection

For the best protection of your data, there is a variety of resources you can use.

You should perform all of the following actions:

- Place your recovery log directories in different file systems.
- Mirror your active and archive recovery logs.

Note: The archive log cannot be mirrored through Tivoli Storage Manager, but it can be mirrored using hardware mirroring, such as that provided by RAID 5.

- Back up your primary storage pools to copy storage pools.
- Copy active client backup data in primary storage pools to active-data pools.

- Create backup copies of the server device configuration file and the volume history file.
- Perform full and incremental backups of your database.

In addition to full and incremental database backups, you can also run snapshot database backups and move the backups offsite.

Storage pool protection overview

If one or more storage pool volumes is lost or damaged, the client data can be permanently lost. However, you can back up storage pools to sequential access copy storage pools and then move the volumes offsite.

If data is lost or damaged, you can restore individual volumes or entire storage pools from the copy storage pools. You can also use special storage pools called *active-data pools* to store active client backup data. Like volumes in copy storage pools, volumes in active-data pools can be moved offsite. Active-data pools are ideally suited for fast client restores.

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.

As a best practice and to protect your inactive data, you should create a minimum of two storage pools: one active-data pool, which contains only active data, and one copy storage pool, which contains both active and inactive data. You can use the active-data pool volumes to restore critical client node data, and afterward you can restore the primary storage pools from the copy storage pool volumes. Active-data pools should not be considered for recovery of a primary pool or volume unless the loss of inactive data is acceptable.

The server tries to access files from a copy storage pool or an active-data pool if the primary file copies cannot be obtained for one of the following reasons:

- The primary file copy has been previously marked damaged (for information about damaged files, see “Fixing damaged files” on page 790).
- The primary file is stored on a volume that UNAVAILABLE or DESTROYED.
- The primary file is stored on an offline volume.
- The primary file is located in a storage pool that is UNAVAILABLE, and the operation is for restore, retrieve, or recall of files to a user, or export of file data.

When restoring active file versions, the server searches in an active-data storage pool associated with a FILE device class, if such a pool exists. For details about the complete storage-pool search-and-selection order, see “Active-data pools as sources of active file versions for server operations” on page 225.

For details, see “Restoring storage pools” on page 777, “Backing up storage pools” on page 760, “Recovering a lost or damaged storage pool volume” on page 796, and “Ensuring the integrity of files” on page 791.

Storage pool restore processing

You can restore files from copy storage pools and active-data pools.

RESTORE STGPOOL

Restores all storage pool files that have been identified as having read errors. These files are known as *damaged* files or *unreadable* files. This command also restores all files on any volumes that have been designated as *destroyed* by using the UPDATE VOLUME command. See “Restoring storage pools” on page 777 for details.

Attention:

- Restoring 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.

RESTORE VOLUME

Recreates files that reside on a volume or volumes in the same primary storage pool. You can use this command to recreate files for one or more volumes that have been lost or damaged. See “Restoring storage pool volumes” on page 780 for details.

Attention:

- Restoring 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.

Tivoli Storage Manager uses database information to determine which files should be restored for a volume or storage pool. As a result, restore processing does not require that the original volumes be accessed. For example, if a primary storage pool volume is damaged, you can use the RESTORE VOLUME command to recreate files that were stored on that volume, even if the volume itself is not readable. However, if you delete the damaged files (DISCARDATA=YES on the DELETE VOLUME command), the server removes references from the database to the files in the primary storage pool volume and to copies of the files in copy storage pool volumes and active-data pool volumes. You cannot restore those files.

Restore processing copies files from a copy storage pool or an active-data pool onto new primary storage pool volumes. The server then deletes database references to files on the original primary storage pool volumes. A primary storage pool volume will become empty if all files that were stored on that volume are restored to other volumes. In this case, the server automatically deletes the empty volume from the database.

Marking volumes as destroyed

The *destroyed* volume access mode designates primary volumes for which files are to be restored

This mode permits the restoration of entire volumes. If a volume is designated as destroyed, the server does not mount that volume for either read or write access. You can designate a volume as destroyed with either of two commands:

- The RESTORE VOLUME command automatically changes the access mode of the specified volumes to destroyed.
- Issue the UPDATE VOLUME command with the ACCESS parameter set to DESTROYED.

The destroyed designation is important during restore processing, particularly when the RESTORE STGPOOL command is used to restore a large number of primary storage pool volumes after a major disaster.

1. Designate as destroyed only those volumes that must be restored. If a volume is known to be usable after a disaster, do not set its access mode to destroyed.
2. After you have identified the primary volumes to be restored and set their access mode to destroyed, you can add new volumes to the storage pool. The new volumes are used to contain the files as they are restored from the copy storage pool volumes or active-data pool volumes. The new volumes can also be used for new files that users back up, archive, or migrate.
3. The destroyed designation permits tracking the files that must still be restored from copy storage pools or active-data pools. If restore processing ends before completion, you can restart the restore. Only the files that still reside on destroyed volumes would need to be restored.

Database and recovery log protection overview

If you lose the active or archive logs, you can lose the changes that have been made since the last database backup. If you lose the database, you can lose all your client data.

You have several ways to protect this information against loss:

- Mirror the active log by using the MIRRORLOGDIR parameter of the DSMSEV FORMAT command or by specifying the MIRRORLOGDIR option in the server options file.
- Mirror the archive log directory using file system or disk drive subsystem level facilities. Insure that the mirror directories are in different file systems on different physical hardware.
- Back up the database to tape or remote virtual volumes. See “Using virtual volumes to store data on another server” on page 712 for more information.

Full database backups are run automatically according to criteria that you can set. You can also run full backups and incremental backups manually. Tivoli Storage Manager can perform full and incremental database backups to tape while the server is running and available to clients. The backup media can then be stored in on-site or off-site locations, and can be used to recover the database up to the point of the backup. You can run full or incremental backups as often as needed to ensure that the database can be restored to an acceptable point in time.

For the fastest recovery time and greatest availability of the database, mirror the active and archive logs, and periodically back up the database. Mirroring helps to ensure that you have an intact log, which is necessary to restore the database to its most current state.

Protecting the Secure Sockets Layer (SSL) digital certificate file

Attention: If client data object encryption is in use and the encryption key is not available, data cannot be restored or retrieved under any circumstance. When using ENABLECLIENTENCRYPTKEY for encryption, the encryption key is stored on the server database. This means that for objects using this method, the server database must exist and have the proper values for the objects for a proper restore. Ensure that you back up the server database frequently to prevent data loss. See *Tivoli Storage Manager Using the Application Program Interface* for more information about encryption keys.

Types of server database restores

There are two types of database restores: point-in-time and most current.

Point-in-time restore

- Removes and recreates the active log directory and archive log directory specified in dsmserv.opt file
- Restores the database image from backup volumes to the database directories recorded in a database backup or to new directories
- Restores archive logs from backup volumes to the overflow directory
- Applies logs from the overflow directory up to specified point in time

Restores using snapshot backups are a form of point-in-time restore.

Most current restore

- Does not remove and recreate the active log directory or archive log directory.
- Restores a database image from the backup volumes to the database directories recorded in a database backup or to new directories.
- Restores archive logs from backup volumes to the overflow directory.
- Applies logs from overflow directory, archive logs from archive log directory, and active logs from active log directory.

Active log mirroring

You can prevent the loss of the active log due to a hardware failure by mirroring the active log in a different file system that resides on a different disk drive.

Consider the following scenario: Because of a sudden power outage, a partial page write occurs. The active log is corrupted and not completely readable. Without mirroring, recovery operations cannot complete when the server is restarted. However, if the active log is mirrored and a partial write is detected, the log mirror can be used to construct valid images of the missing data.

Mirroring simultaneously writes the same data to another disk. However, mirroring does not protect against a disaster or a hardware failure that affects multiple drives or causes the loss of the entire system. While Tivoli Storage Manager is running, you can dynamically start or stop mirroring and change the capacity of the database. Mirroring provides the following benefits:

- Protection against database and log media failures
- Uninterrupted operations if the active log fails
- Avoidance of costly database recoveries

However, there are also costs:

- Mirroring doubles the required disk space for the mirrored logs
- Mirroring results in decreased performance

Note: Mirroring the active log and the archive log should be considered when retention protection is enabled. If a database restore is needed, the database can be brought back to the current point in time with no data loss. Mirror the archive log directory through operating system or hardware facilities.

Snapshot database backup

A snapshot database backup is a full database backup that does not interrupt the current full and incremental backup series.

Snapshot database tapes can then be taken off-site for recovery purposes and therefore be kept separate from the normal full and incremental backup tapes. For information about doing a snapshot of the database, see “Running snapshot database backups” on page 772.

Secure Sockets Layer digital certificate file protection

As part of the process of setting up Tivoli Storage Manager to use Secure Sockets Layer (SSL) for client-server authentication, a digital certificate file, cert.kdb, is created. This file includes the server's public key, which allows the client to encrypt data. The digital certificate file cannot be stored in the server database because the GSKit requires a separate file in a certain format. Therefore, you should keep backup copies of the cert.kdb file and cert.arm file. If, however, both the original files and any copies are lost or corrupted, you can regenerate a new certificate file. For details about this procedure, see “Maintaining the certificate key database” on page 424.

Attention: If client data object encryption is in use and the encryption key is not available, data cannot be restored or retrieved under any circumstance. When using ENABLECLIENTENCRYPTKEY for encryption, the encryption key is stored on the server database. This means that for objects using this method, the server database must exist and have the proper values for the objects for a proper restore. Ensure that you back up the server database frequently to prevent data loss. See *Tivoli Storage Manager Using the Application Program Interface* for more information about encryption keys.

Backing up storage pools

You can back up primary storage pools to copy storage pools to improve data availability.

Task	Required Privilege Class
Define, back up, or restore storage pools	System, unrestricted storage, or restricted storage (only for those pools to which you are authorized)
Restore volumes	

When you back up a primary storage pool, you create copies of client files (active and inactive backup files, archive files, and space-managed files) that are stored in primary storage pools. By using copy storage pools, you maintain multiple copies of files and reduce the potential for data loss due to media failure. If the primary file is not available or becomes corrupted, the server accesses and uses the duplicate file from a copy storage pool.

You can also copy active client backup data from primary storage pools to active-data pools. Archive and space-migrated data are not permitted in active-data pools. If a primary storage pool does not contain an active backup-file version, the server obtains the file from the active-data pool. Like copy storage pools, active-data pools reduce the potential for data loss due to media failure.

Fast client restores are the main benefit of active-data pools. To achieve optimal restore times, you can associate an active-data pool with a random-access sequential disk device class with a device type of FILE. With a FILE-type active-data pool, the server does not have to mount tape volumes and does not have to position past inactive files on a volume. In addition, client sessions can access active-data pool FILE volumes concurrently with RESTORE STGPOOL and RESTORE VOLUME processes. This greatly speeds up client restores. However, FILE-type active-data pools are onsite. In the event of an onsite disaster, the data in these pools can be lost.

You can also create active-data pools using a device class associated with removable tape or optical media that can be taken offsite, stored, and then brought back onsite if a disaster occurs. To restore client data, these tapes need to be mounted, but the server does not have to position past inactive files. Active-data pools can also be associated with a SERVER device class. The volumes in a SERVER-type, active-data pool can be located offsite for protection in case of an onsite disaster. If a disaster occurs, you save time and bandwidth by restoring only the active versions of client backup data.

As a best practice, you should create a minimum of two storage pools: one active-data pool and one conventional copy storage pool. You can use the active-data pool to restore critical client node data, and afterward you can restore the primary storage pools from the copy storage pool volumes that include the active and inactive versions. If an active-data pool becomes lost or damaged, you can restore it from the primary storage pool using the COPY ACTIVATEDATA command.

Figure 99 on page 762 shows a configuration with an onsite FILE-type active-data pool and an offsite copy storage pools.

Note: A BACKUP STGPOOL command does not back up a shred storage pool to a copy storage pool unless you explicitly permit it by setting the SHREDTONOSHRED parameter to YES. If this value is not specified, the server issues an error message and does not allow the backup. If this value is specified, the server does not issue a warning when the BACKUP STGPOOL command for the shred pool is run. See “Securing sensitive client data” on page 499 for more information about shredding.

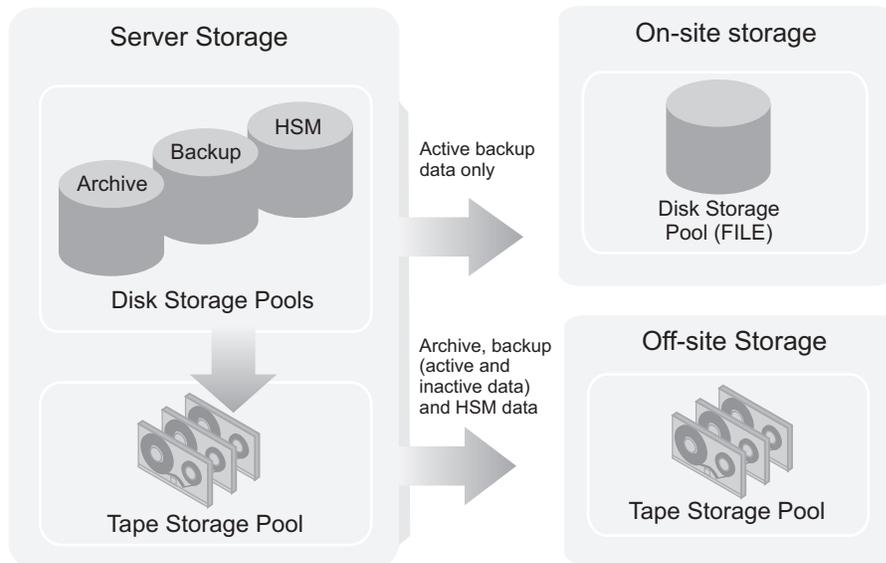


Figure 99. Copy storage pools and active-data pools

Primary storage pools should be backed up each day to the same copy storage pool, and active backup data should be copied each day to the same active-data pool. Backing up to the same copy storage pool and active-data pool ensures that files do not need to be recopied if they have migrated to the next pool.

For example, to back up the ARCHIVEPOOL primary storage pool to the DISASTER-RECOVERY copy storage pool, enter:

```
backup stgpool archivepool disaster-recovery maxprocess=4
```

The only files backed up to the DISASTER-RECOVERY pool are files for which a copy does not already exist in the copy storage pool. The data format of the copy storage pool and the primary storage pool can be NATIVE, NONBLOCK, or any of the NDMP formats (NETAPPDUMP, CELERRADUMP, or NDMPDUMP). The server will copy data from the primary storage pool only to a copy storage pool that has the same format.

To copy active data from the BACKUPPOOL primary storage pool to the CLIENT-RESTORE active-data pool, issue the following command:

```
copy activedata backuppool client-restore maxprocess=4
```

The primary storage pool must have a data format of NATIVE or NONBLOCK. Copies from primary storage pools with any of the NDMP formats are not permitted. The only files copied to the CLIENT-RESTORE pool are active backup files for which a copy does not already exist in the active-data pool.

Each of the command examples above uses four parallel processes (MAXPROCESS=4) to perform an incremental backup of the primary storage pool to the copy storage pool or a copy to the active-data pool. Set the MAXPROCESS parameter in the BACKUP STGPOOL command to the number of mount points or drives that can be dedicated to this operation.

Because the backups and active-data copies are made incrementally, you can cancel the processes. Reissuing the BACKUP STGPOOL or COPY ACTIVEDATA command lets the backup or active-data copy continue from the spot the process was canceled.

You can back up multiple primary storage pools to one copy storage pool. You can also copy active data from multiple primary storage pools to one active-data pool. If multiple backups and copies are necessary, you can also back up a primary storage pool to multiple copy storage pools and copy active data to multiple active-data pools. However, for easier management of storage volumes, you should back up the entire primary storage pool hierarchy to the same copy storage pools. Similarly, you should copy the active data in the primary storage pool hierarchy to the same active-data pools. See “Using multiple copy storage pools and active-data pools” on page 765.

See the following sections:

“Scheduling storage pool backups” on page 764

“Scenario: scheduling a backup with one copy storage pool” on page 764

“Simultaneous-write operations to copy storage pools and active-data storage pools” on page 765

“Using multiple copy storage pools and active-data pools” on page 765

“Delaying reuse of volumes for recovery purposes” on page 766

For recovery scenarios that use backed-up copies of storage pools, see “Recovering to a point-in-time from a disaster” on page 794 and “Recovering a lost or damaged storage pool volume” on page 796.

Backing up storage pools requires an additional 200 bytes of space in the database for each file copy. As more files are added to the copy storage pools and active-data pools, reevaluate your database size requirements.

Be aware of the following storage pools behaviors:

- If a backup is to be made to a copy storage pool and the file already exists with the same insertion date, no action is taken. Similarly, if a copy is to be made to an active-data pool and the file already exists with the same insertion data, no action is taken.
- When a disk storage pool is backed up, cached files, (copies of files that remain on disk after being migrated to the next storage pool) are not backed up.
- Files in a copy storage pool or an active-data pool do not migrate to another storage pool.
- After a file is backed up to a copy storage pool or a copy is made to an active-data pool, the file might be deleted from the primary storage pool. When an incremental backup of the primary storage pool occurs, the file is then deleted from the copy storage pool. Inactive files in active-data pools are deleted during the process of reclamation. If an aggregate being copied to an active-data pool contains some inactive files, the aggregate is reconstructed into a new aggregate without the inactive files.

Scheduling storage pool backups

For the best protection, primary storage pools should be backed up regularly, preferably each day, to copy storage pools.

Copies of active backup data to active-data pools should also be made. You can define schedules to begin backups and copies of files in the primary storage pools. For example, to back up the BACKUPPOOL, ARCHIVEPOOL, and TAPEPOOL storage pools every night, schedule the following commands:

```
backup stgpool backuppool disaster-recovery maxprocess=4
```

```
backup stgpool archivepool disaster-recovery maxprocess=4
```

```
backup stgpool tapepool disaster-recovery maxprocess=4
```

```
copy activedata backuppool client-restore maxprocess=4
```

See Chapter 20, “Automating server operations,” on page 571 for information about scheduling commands.

If you schedule storage pool backups and migrations and have enough disk storage, you can copy most files from the disk storage pool before they are migrated to tape and thus avoid unnecessary tape mounts. Here is the sequence:

1. Clients back up or archive data to disk
2. You issue or schedule the BACKUP STGPOOL command to back up the primary storage pools to copy storage pools. Similarly, you can issue or schedule the COPY ACTIVEDATA command to copy active client backup data from the primary storage pools to active-data pools.
3. Data migrates from disk storage pools to primary tape storage pools

Scenario: scheduling a backup with one copy storage pool

This scenario show how to create a schedule for backing up two primary storage pools to the same copy storage pool.

Assume that you have two primary storage pools: one random access storage pool (DISKPOOL) and one tape storage pool (TAPEPOOL, with device class TAPECLASS). Files stored in DISKPOOL are migrated to TAPEPOOL. You want to back up the files in both primary storage pools to a copy storage pool.

To schedule daily incremental backups of the primary storage pools, do the following steps:

1. Define a copy storage pool called COPYPOOL, with the same device class as TAPEPOOL, by issuing the following command:

```
define stgpool copypool tapeclass pooltype=copy maxscratch=50
```

Note:

- a. Because scratch volumes are allowed in this copy storage pool, you do not need to define volumes for the pool.
 - b. All storage volumes in COPYPOOL are located onsite.
2. Perform the initial backup of the primary storage pools by issuing the following commands:

```
backup stgpool diskpool copypool maxprocess=2
```

```
backup stgpool tapepool copypool maxprocess=2
```

3. Define schedules to automatically run the commands for backing up the primary storage pools. The commands to schedule are those that you issued in step 2 on page 764

To minimize tape mounts, back up the disk storage pool first, then the tape storage pool.

For more information about scheduling, see Chapter 20, “Automating server operations,” on page 571.

Simultaneous-write operations to copy storage pools and active-data storage pools

You can set up a primary storage pool so that when a client backs up, archives, or migrates a file, the file is written to the primary storage pool and is simultaneously stored into each copy storage pool specified for the primary storage pool.

You can also enable the simultaneous-write function so that active client backup data is written to active-data pools at the same time it is written to the primary storage pool. The active-data pools must be specified in the definition of the primary storage pool, and the clients whose active data is to be saved must be members of a policy domain that specifies the active-data pool as the destination for active backup data.

Use of the simultaneous-write function is not intended to replace regular backups of storage pools. If you use the function to simultaneously write to copy storage pools and active-data pools, ensure that the copy of each primary storage pool and of the active-data in each primary storage pool is complete by regularly issuing the BACKUP STGPOOL command. See “Writing data simultaneously to primary, copy, and active-data pools” on page 285 for more information.

Using multiple copy storage pools and active-data pools

When Tivoli Storage Manager restores data, there might be some duplication of restored files. This can occur if primary volumes are not available, and Tivoli Storage Manager does not have a complete copy storage pool or active-data pool from which to perform the restore.

In such cases, Tivoli Storage Manager uses volumes from multiple copy storage pools or active-data pools to restore the data. This process can result in duplicate data being restored. To prevent this duplication, keep one complete set of copy storage pools and one complete set of active-data pools available to the server, or ensure that only one copy storage pool or one active-data pool has an access of read/write during the restore operation.

Duplication of restored files only occurs when these conditions exist:

- Primary volumes are unavailable or offsite.
- Multiple copy storage pools or active-data pools are available.
- Copy storage pools and active-data pools do not contain all of the files that are in the primary storage pools.

The following example explains this scenario:

The primary storage pool Main contains volumes Main1, Main2, and Main3.

- Main1 contains files File11, File12, File13
- Main2 contains files File14, File15, File16
- Main3 contains files File17, File18, File19

The copy storage pool DuplicateA contains volumes DupA1, DupA2, and DupA3.

- DupA1 contains copies of File11, File12
- DupA2 contains copies of File13, File14
- DupA3 contains copies of File15, File16, File17, File18 (File19 is missing because BACKUP STGPOOL was run on the primary pool before the primary pool contained File 19.)

The copy storage pool DuplicateB contains volumes DupB1 and DupB2.

- DupB1 contains copies of File11, File12
- DupB2 contains copies of File13, File14, File15, File16, File17, File18, File19

If you have not designated copy storage pool DuplicateB as the only copy storage pool to have read/write access for the restore operation, then Tivoli Storage Manager can choose the copy storage pool DuplicateA, and use volumes DupA1, DupA2, and DupA3. Because copy storage pool DuplicateA does not include file File19, Tivoli Storage Manager would then use volume DupB2 from the copy storage pool DuplicateB. The program does not track the restoration of individual files, so File15, File16, File17, and File18 will be restored a second time, and duplicate copies will be generated when volume DupB2 is processed.

Delaying reuse of volumes for recovery purposes

When you define or update a sequential access storage pool, you can use the REUSEDELAY parameter. This parameter specifies the number of days that must elapse before a volume can be reused or returned to scratch status after all files have been expired, deleted, or moved from the volume.

When you delay reuse of such volumes and they no longer contain any files, they enter the *pending* state. Volumes remain in the pending state for as long as specified with the REUSEDELAY parameter for the storage pool to which the volume belongs.

Delaying reuse of volumes can be helpful under certain conditions for disaster recovery. When files are expired, deleted, or moved from a volume, they are not actually erased from the volumes: The database references to these files are removed. Thus the file data may still exist on sequential volumes if the volumes are not immediately reused.

A disaster may force you to restore the database using a database backup that is not the most recent backup. In this case, some files may not be recoverable because the server cannot find them on current volumes. However, the files may exist on volumes that are in pending state. You may be able to use the volumes in pending state to recover data by doing the following steps:

1. Restore the database to a point-in-time prior to file expiration.
2. Use a primary, copy-storage, or active-data pool volume that has not been rewritten and contains the expired file at the time of database backup.

If you back up your primary storage pools, set the REUSEDELAY parameter for the primary storage pools to 0 to efficiently reuse primary scratch volumes. For your copy storage pools and active-data pools, you should delay reuse of volumes for as long as you keep your oldest database backup.

For an example of using database backup and delaying volume reuse, see “Protecting the database and storage pools” on page 792. For information about expiration, see “Running expiration processing to delete expired files” on page 472.

Backing up the server database

You can back up the database with full and incremental backups or by taking a snapshot of a specific point-in-time of the database.

See “Running full and incremental backups” on page 771 and “Running snapshot database backups” on page 772 for more information.

To set up regular database backups, complete the following steps:

1. Define device classes for backups.
2. Estimate the active log size.
3. Schedule database backups to occur as needed.

To restore your database, the following information is required:

- Ensure that you have copies of the volume history file and the device configuration file.
- Ensure that you have copies of, or are able to create, the server options file and the database and recovery log set up information (the output from detailed queries of your database and recovery log).

DRM: DRM helps you save this required information.

Defining device classes for backups

You can use existing device classes for backups or define new ones. You can also specify different device classes for incremental backups and for full backups. For example, you might want to write full backups to tape and incremental backups to disk.

You should also reserve a device class, and therefore a device, for backups only. In this way, the server does not try to back up the database with no device available. If a database backup shares a device class with a low priority operation, such as reclamation, and all the devices are in use, the lower priority operation is automatically canceled. This frees a device for the database backup.

You must issue the `SET DBRECOVERY.` command to specify the device class to be used for backups. See “Preparing the system for database backups” on page 606 for details.

Device class definitions are saved in the device configuration files (see “Saving the device configuration file” on page 770).

Estimating the size of the active log

The number of transactions affect how large you should make your active log. As you add more clients and increase concurrent transactions, you can increase the size of the log.

To determine the size that the active log should be, calculate how much active log space is used between database backups. For example, if you perform daily incremental backups, check your daily usage over a period of time.

For information on how to adjust the active log size, see “Increasing the size of the active log” on page 605.

Scheduling database backups

Database backups require devices, media, and time. Consider scheduling backups to occur at specific times of the day and after specific activities.

For example, you might want to schedule backups after the following types of activities:

- Major client backup or archive operations
- Storage pool migration and reclamation
- Storage pool backups
- MOVE DATA or DELETE VOLUME commands

Depending on the frequency of these activities and the amount of client data, you might back up your storage pools daily and then immediately back up the database.

When deciding what kind of backups to do and when to do them, consider the following properties of backups:

- Full backups take longer than incremental backups
- Full backups have shorter recovery times than incremental backups (you must load only one set of volumes to restore the entire database)
- Full backups are required:
 - For the first backup
 - After extending the database size

Saving the volume history file

To perform a database restore, the server needs information from the volume history file. Volume history information is stored in the database, but during a database restore, it is not available from there. It is critical that you make a copy of your volume history file and save it. The file cannot be recreated.

It is essential to save your volume history file. Without it, you cannot restore your database. The following volume information is stored in the database:

- Sequential access storage pool volumes that have been added, reused (through reclamation or move data operations), or deleted (during delete volume or reclamation operations)
- Full and incremental database backup volume information
- Export volumes for administrator, node, policy, and server data
- Snapshot database backup volume information
- Backup set volume information.

The server updates the volume history file as volumes are added. However, you must periodically run a delete operation to discard outdated information about volumes (see “Deleting volume history information” on page 769 for details).

To ensure the availability of volume history information, it is extremely important to take one of the following steps:

- Store at least one copy of the volume history file offsite or on a disk separate from the database
- Store a printout of the file offsite
- Store a copy of the file offsite with your database backups and device configuration file

- Store a remote copy of the file, for example, on an NFS-mounted file system.

DRM: DRM saves a copy of the volume history file in its disaster recovery plan file.

The VOLUMEHISTORY server option lets you specify backup volume history files. Then, whenever the server updates volume information in the database, it also updates the same information in the backup files.

You can also back up the volume history information at any time, by entering:
backup volhistory

If you do not specify file names, the server backs up the volume history information to all files specified with the VOLUMEHISTORY server option.

In order to ensure updates are complete before the server is halted, the following steps are recommended:

- Do 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 is updated.

Deleting volume history information

You should periodically delete outdated information from the volume history file.

For example, if you keep backups for seven days, information older than seven days is not needed. When information about database backup volumes or export volumes is deleted, the volumes return to scratch status. For scratch volumes of device type FILE, the files are deleted. When information about storage pools volumes is deleted, the volumes themselves are not affected.

To display volume history information up to yesterday, enter:

```
query volhistory enddate=today-1
```

To delete information that is seven days old or older, enter:

```
delete volhistory type=all todate=today-8
```

Consider the following information before deleting volume history information:

- Existing volume history files are *not* automatically updated with the DELETE VOLHISTORY command.
- Do not delete sequential volume history information until you no longer need that information. For example, do not delete storage volume reuse information, unless you have backed up the database at a later time than that specified for the delete operation.
- Do not delete the volume history information for database backup or export volumes that reside in automated libraries, unless you want to return the volumes to scratch status. When the DELETE VOLHISTORY command removes volume information for such volumes, they automatically return to scratch status. The volumes are then available for reuse by the server and the information stored on them may be overwritten.
- You cannot remove the most current database snapshot entry by performing a DELETE VOLHISTORY. This ensure that you will have a backup to recover from. Even if a more current standard database backup exists, the latest database snapshot is not deleted.

DRM: DRM expires database backup series and deletes the volume history entries.

Saving the device configuration file

The device configuration file contains information required to read backup data. It is critical that you make a copy of your device configuration file and save it. The file cannot be recreated.

Without the device configuration file, you cannot restore your database. The device configuration file includes the following definitions:

- Devices class definitions
- Library definitions
- Drive definitions
- Path definitions
- Server definitions
- The database manager backup node ID

This information is stored in the database, but during a database restore, it is not available from there. To perform a restore, therefore, the server must get the information from the device configuration file. When device information is updated in the database, it is also updated in the device configuration file. The device information must match the devices configured on the system where the restore will be performed. You may have to edit those commands in an existing file so that they match.

Only path definitions with SRCTYPE=SERVER are backed up to the device configuration file. Paths of SRCTYPE=DATAMOVER are not written out to the file.

To ensure the availability of the device configuration information, it is extremely important that you take one of the following steps:

- Store at least one backup copy of the device configuration file on a disk separate from the database
- Store your device configuration file offsite with your volume history file and database backups
- Store a printout of the information that is stored offsite
- Store a remote copy, for example, on an NFS-mounted file system

DRM: DRM saves a copy of the device configuration file in its disaster recovery plan file.

The DEVCONFIG server option lets you specify backup device configuration files (for details, see the *Administrator's Reference*). After the server is restarted, whenever the server updates device configuration information in the database, it also updates the same information in the backup files.

During a database restore operation, the server tries to open the first device configuration file in the order in which the files occur in the server options. If it cannot read that file, it searches for the next usable device configuration file. After the database has been restored, you might have to update the device configuration.

You can also back up the device configuration information at any time, by entering:

```
backup devconfig
```

If you do not specify file names, the device configuration file is backed up to *all* files specified with the DEVCONFIG server option.

In order to ensure updates are complete before the server is halted, the following actions are recommended:

- After issuing the BACKUP DEVCONFIG command, wait several minutes before halting the server.
- Specify multiple DEVCONFIG options in the server options file.
- Examine the device configuration file to see if the file is updated.

If you are using automated tape libraries, volume location information is also saved in the device configuration file. The file is updated whenever CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME, and AUDIT LIBRARY commands are issued, and the information is saved as comments (*/*...*/*). This information is used during restore or load operations to locate a volume in an automated library.

If a disaster occurs, you might have to restore Tivoli Storage Manager with devices that are not included in the device configuration file. See “Updating the device configuration file” on page 776 for more information.

Saving the server options and database and recovery log information

To restore the database, you need copies of the server options and of database and recovery log information.

Gather the following information:

- The server options file
- The output of the following database commands:

```
query db format=detailed
query dbspace
```
- The output of the following log command:

```
query log format=detailed
```

Running full and incremental backups

The first backup of your database must be a full backup.

To perform a full backup of your database to the TAPECLASS device class, enter:

```
backup db type=full devclass=tapeclass
```

In this example, the backup data is written to scratch volumes. You can also specify volumes by name. After a full backup, you can perform incremental backups, which copy all the changes to the database since the last full database backup.

To do an incremental backup of the database to the TAPECLASS device class, enter:

```
backup db type=incremental devclass=tapeclass
```

Running snapshot database backups

Snapshot database backups should be used in addition to full and incremental backups. When a snapshot database backup is performed, the recovery log keeps growing.

Snapshot database tapes can then be taken off-site for recovery purposes and therefore kept separate from the normal full and incremental backup tapes. Snapshot database backups enhance the protection of your server and its data while maintaining the full and incremental database backup series. Although snapshot database backups cannot restore a database to its most current state, you can use them to restore a database to a specific point-in-time.

Snapshot database backups:

- Copy the complete contents of a database, just like a full database backup.
- Create a new database backup series without interrupting the existing full and incremental backup series for the database.

Use the BACKUP DB command to perform a snapshot database backup. New volume history entries are created for the snapshot database volumes. Note that the most current snapshot database cannot be deleted with the DELETE VOLHISTORY command. This is to prevent the accidental loss of what could be the only way to recover the server.

To perform a snapshot database backup to the TAPECLASS device class, enter:

```
backup db type=dbsnapshot devclass=tapeclass
```

Snapshot database backups should be used in addition to full and incremental backups. When a snapshot database backup is performed, the recovery log keeps growing. When full and incremental backups are performed, the recovery log is restarted each time a full backup is performed.

Recovering the server using database and storage pool backups

To recover the server you need backups of the database and storage pools.

Figure 100 on page 773 shows the situation presented in the two scenarios in this section: an installation has lost its server, including the database and recovery log, and its onsite storage pools.

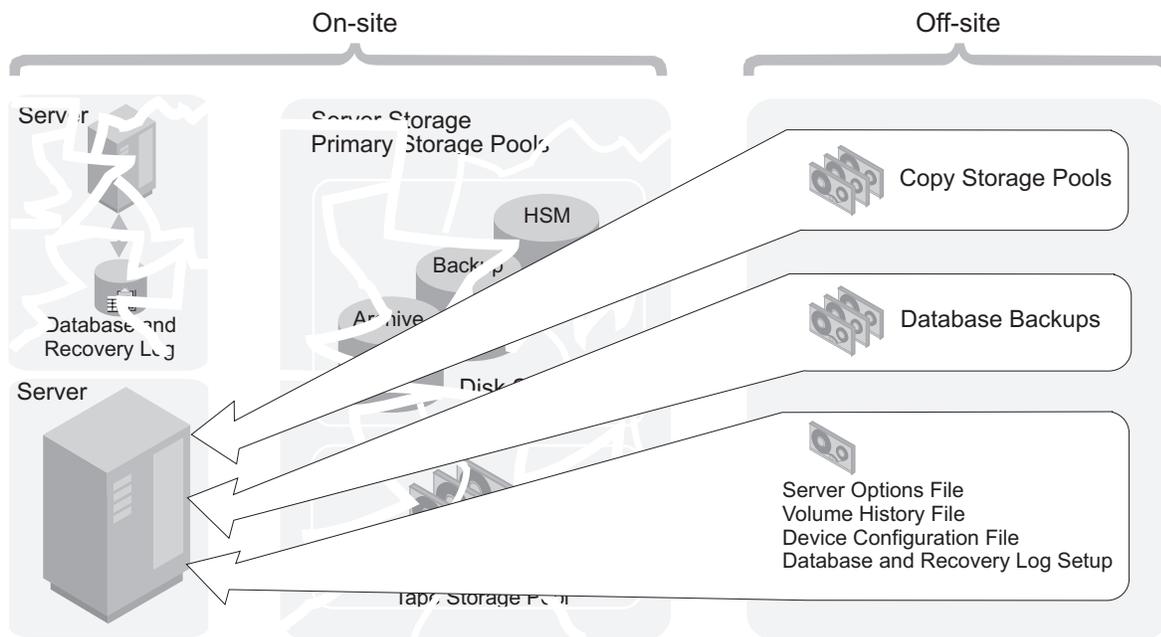


Figure 100. Recovery from a disaster

The following topics are included:

- Restoring to a point-in-time
- Restoring to the most current state

To perform a restore, you should have the following information, preferably stored offsite (see Figure 100):

- A full database backup
- The latest incremental database backup
- Copy storage pool volumes
- On tape or diskette, or as printouts:
 - Server options file
 - Volume history file
 - Device configuration file with the applicable device information (library, drive, and device class definitions)
 - Database and recovery log setup (the output from detailed queries of your database and recovery log)

DRM: DRM can query the server and generate a current, detailed disaster recovery plan for your installation.

Restoring a server database to a point in time

Point-in-time recovery is typically used for situations such as disaster recovery or to remove the effects of errors that can cause inconsistencies in the database.

You can use either full and incremental backups or snapshot database backups to restore a database to a point-in-time.

For a scenario of recovering to a point-in-time, see “Backup and recovery scenarios” on page 792.

To restore the database:

1. You must have a volume history file to restore the database. It cannot be recreated. Before restoring the database, copy the volume history file pointed to by the server options file. The backup copy will have a different name. You might need this backup copy if the restore fails and you need to retry. After the database is restored, any volume history information pointed to by the server options is lost. You will need this information to identify the volumes to be audited.
2. You must also have a device configuration file to restore the database. It cannot be recreated. Save a copy of the device configuration file. Do the same with the server options file. Have available your outputs from your detailed queries about your database and recovery log setup information.

You might need to modify the device configuration file based on the hardware available at the recovery site. For example, the recovery site might require a different device class, library, and drive definitions. For more information, see “Updating the device configuration file” on page 776

3. If the database or recovery log directories were lost, recreate the directories. For example:

```
mkdir /tsmdb001
mkdir /tsmdb002
mkdir /tsmdb003
mkdir /activelog
mkdir /archlog
mkdir /archfaillog
```

4. Issue the DSMSEV RESTORE DB utility. For example, to restore the database to a backup series that was created on April 19, 2009, enter:

```
dsmserv restore db todate=04/19/2009
```

The server does the following actions:

- a. Reads the volume history file to locate the last full backup that occurred on or before the specified date and time.
 - b. Using the device configuration file, requests a mount of the first volume, which should contain the beginning of the full backup.
 - c. Restores the backup data from the first volume.
 - d. Continues to request mounts and to restore data from the backup volumes that contain the full backup and any incremental backups that occurred on or before the date specified.
5. From the old volume history information (generated by the QUERY VOLHISTORY command) you need a list of all the volumes that were reused (STGREUSE), added (STGNEW), and deleted (STGDELETE) since the original backup. Use this list to perform the rest of this procedure. It might also be necessary to update the device configurations in the restored database. For details, see “Updating the device configuration file” on page 776.

6. Audit all disk volumes, all reused volumes, and any deleted volumes located by the AUDIT VOLUME command using the FIX=YES parameter.
This process identifies files recorded in the database that can no longer be found on the volume. If a copy of the file is in a copy storage pool or active-data pool, the file on the audited volume is marked as damaged. Otherwise, the file is deleted from the database and is lost.
7. If the audit detects any damaged files, issue the RESTORE STGPOOL command to restore those files after you have audited the volumes in the storage pool. Include the FIX=YES parameter on the AUDIT VOLUME command to delete database entries for files not found in the copy storage pool.
8. Mark as destroyed any volumes that cannot be located, and recover those volumes from copy storage pool backups. If no backups are available, delete the volumes from the database by using the DELETE VOLUME command with the DISCARDDATA=YES parameter.
9. Redefine any storage pool volumes that were added since the database backup.

You can take some additional measures to increase the protection of your data:

- Some files might be lost if they were moved after the backup (due to migration, reclamation, or move data requests) and the space occupied by those files has been reused. You can minimize this loss by using the REUSEDELAY parameter when defining or updating sequential access storage pools. This parameter delays volumes from being returned to scratch or being reused. See “Delaying reuse of volumes for recovery purposes” on page 766 for more information on the REUSEDELAY parameter.
- By backing up your storage pool and your database, you reduce the risk of losing data. To further minimize loss of data, you can:
 - Mark the backup volumes in the copy storage pool as OFFSITE and move them to an offsite location.
In this way the backup volumes are preserved and are not reused or mounted until they are brought onsite. Ensure that you mark the volumes as OFFSITE before you back up the database.
To avoid having to mark volumes as offsite or physically move volumes:
 - Specify a device class of DEVTYPE=SERVER in your database backup.
 - Back up a primary storage pool to a copy storage pool or associated with a device class of DEVTYPE=SERVER.
 - Back up the database immediately after you back up the storage pools.
 - Turn off migration and reclamation while you back up the database.
 - Do not perform any MOVE DATA operations while you back up the database.
 - Use the REUSEDELAY parameter's interval to prevent your copy storage pool volumes from being reused or deleted before they might be needed.
- If your old volume history file shows that any of the copy storage pool volumes needed to restore your storage pools have been reused (STGREUSE) or deleted (STGDELETE), you may not be able to restore all your files. You can avoid this problem by including the REUSEDELAY parameter when you define your copy storage pools.
- After a restore, the volume inventories for Tivoli Storage Manager and for your tape management system may be inconsistent. For example, after a database backup, a new volume is added to Tivoli Storage Manager. The tape management system inventory records the volume as belonging to Tivoli Storage

Manager. If the database is restored from the backup, Tivoli Storage Manager has no record of the added volume, but the tape management system does. You must synchronize these inventories.

Similarly, the volume inventories for Tivoli Storage Manager and for any automated libraries may also be inconsistent. If they are, issue the AUDIT LIBRARY command to synchronize these inventories.

Restoring a server database to its most current state

You can use full and incremental backups to restore a database to its most current state. Snapshot database backups are complete database copies of a point in time.

You can restore a database to its most current state if the last backup series that was created for the database is available. A backup series consists of a full backup, the latest incremental backup, and all active and archive logs for database changes since the last backup in the series was run.

To restore the database to its most current state, enter the DSMSEV RESTORE DB command. For example:

```
dsmserv restore db
```

If the original database and recovery log directories are available, issuing the DSMSEV RESTORE DB utility will restore the database. However, if they have been lost, first recreate them, then issue the DSMSEV RESTORE DB utility.

Note: Recover the database to its most current state is not possible if the active or archive logs are lost.

Updating the device configuration file

If a disaster occurs, you might have to restore Tivoli Storage Manager with devices that are not included in the device configuration file.

In such a case, you must update the device configuration files manually with information about the new devices. Whenever you define, update, or delete device information in the database, the device configuration file is automatically updated. This information includes definitions for device classes, libraries, drives, and servers.

Definitions for paths are included when SRCTYPE=SERVER.

Library volume location information is updated in the device configuration file whenever CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME, and AUDIT LIBRARY commands are issued for SCSI libraries.

If an automated tape library is used at the recovery site, volume location information in comments (*/*...*/*) in the device configuration file must be modified. First, manually place the physical database backup volumes in the automated library and note the element numbers where you place them. Then manually edit the device configuration file to identify the locations of the database backup volumes so that the server can find them to restore the database.

For virtual volumes, the device configuration file stores the password (in encrypted form) for connecting to the remote server. If you regressed the server to an earlier point-in-time, this password might not match what the remote server expects. In

this case, manually set the password in the device configuration file. Then ensure that the password on the remote server matches the password in the device configuration file.

Note: Set the password in clear text. After the server is operational again, you can issue a `BACKUP DEVCONFIG` command to store the password in encrypted form.

Restoring storage pools

You can recreate files in a primary storage pool by using duplicate copies in copy storage pools.

The files must have been copied to the copy storage pools by using the `BACKUP STGPOOL` command or during a simultaneous-write operation. You can also recreate active versions of client backup files in a primary storage pool by using duplicate copies in active-data pools. The files in active-data pools must have been copied to the pools by using the `COPY ACTIVEDATA` command or during a simultaneous-write operation.

Restoring 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. Active-data pools should not be considered for recovery of a primary pool unless the loss of inactive data is acceptable.

Task	Required Privilege Class
Restoring storage pools	System, unrestricted storage, or restricted storage

The `RESTORE STGPOOL` command restore specified primary storage pools that have files with the following problems:

- The primary copy of the file has been identified as having read errors during a previous operation. Files with read errors are marked as damaged.
- The primary copy of the file resides on a volume that has an access mode of `DESTROYED`. For how the access mode of a volume changes to the `DESTROYED` access mode, see “Storage pool restore processing” on page 757.

When you restore a storage pool, be prepared to provide the following information:

Primary storage pool

Specifies the name of the primary storage pool that is being restored.

Copy storage pool

Specifies the name of the copy storage pool from which the files are to be restored. This information is optional. If you do not specify a copy storage pool, the server restores the files from any copy storage pool where it can find them.

Active data only

Specifies that active versions of backup files are to be restored from active-data pools only. This information is optional. If it is not provided, files are restored from copy storage pools.

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.

Active-data pool

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 information is not provided, files are restored from any active-data pool in which active versions of backup files can be located

New storage pool

Specifies the name of the new primary storage pool to which to restore the files. This information is optional. If you do not specify a new storage pool, the server restores the files to the original primary storage pool.

Maximum number of processes

Specifies the maximum number of parallel processes that are used for restoring files.

Preview

Specifies whether you want to preview the restore operation without actually restoring data.

See “Fixing damaged files” on page 790 and “Backup and recovery scenarios” on page 792 for examples of using the RESTORE STGPOOL command.

Storage pool restoration

When you restore a storage pool, Tivoli Storage Manager determines which files are in that storage pool.

Using file copies from a copy storage pool, Tivoli Storage Manager restores the files that were in the storage pool to the same or a different storage pool. Using files from an active-data pool, Tivoli Storage Manager restores the active versions of client backup data to the same or a different storage pool. As part of the restore operation, inactive file versions are deleted from the server database if the server determines that an inactive file needs to be replaced but cannot find it in the active-data pool.

Note: Cached copies of files in a disk storage pool are never restored. References to any cached files that have been identified as having read errors or cached files that reside on a *destroyed* volume will be removed from the database during restore processing.

The RESTORE STGPOOL command with the PREVIEW=YES parameter can be used 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, noncached files. To identify the specific files that are damaged on these volumes, use the QUERY CONTENT command.

After the files are restored, the old references to these files in the primary storage pool are deleted from the database. This means that Tivoli Storage Manager now locates these files on the volumes to which they were restored, rather than on the volumes on which they were previously stored. If a destroyed volume becomes empty because all files have been restored to other locations, the destroyed volume is automatically deleted from the database.

The RESTORE STGPOOL command generates a background process that can be canceled with the CANCEL PROCESS command. If a RESTORE STGPOOL background process is canceled, some files may have already been restored prior to the cancellation. To display information about background processes, use the QUERY PROCESS command.

The RESTORE STGPOOL command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

Restoring files to a storage pool with collocation enabled

When restoring files to a primary storage pool that has collocation enabled, the server restores the files by collocation group, by client node, or by client file space.

This process preserves the collocation of client files. However, if the copy storage pool or active-data pool being used to restore files does not have collocation enabled, restore processing can be slow.

If you need to use a copy storage pool or an active-data pool that is not collocated to restore files to a primary storage pool that is collocated, you can improve performance by performing the following steps::

1. Restore the files first to a random access storage pool (on disk).
2. Allow or force the files to migrate to the target primary storage pool.
For the random access pool, set the target storage pool as the next storage pool. Adjust the migration threshold to control when migration occurs to the target storage pool.

Fixing an incomplete storage pool restoration

If the restoration of storage pool volumes is incomplete, you can get more information about the remaining files on those volumes.

The restoration may be incomplete for one or more of the following reasons:

- Either files were never backed up, or the backup copies were marked as damaged.
- A copy storage pool or active-data pool was specified on the RESTORE command, but files were backed up to a different copy storage pool or active-data pool. If you suspect this is a problem, use the RESTORE command again without specifying a copy storage pool or active-data pool from which to restore files. The PREVIEW option can be used on the second RESTORE command, if you do not actually want to restore files.
- Volumes in the copy storage pool or active-data 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 or active-data pools were moved or deleted by other processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool volumes or active-data pool volumes while restore processing is in progress:
 - MOVE DATA
 - DELETE VOLUME (DISCARDDATA=YES)
 - AUDIT VOLUME (FIX=YES)
 - MIGRATE STGPOOL
 - RECLAIM STGPOOL

- You can prevent reclamation processing for your copy storage pools and active-data pools by setting the RECLAIM parameter to 100 with the UPDATE STGPOOL command.

Restoring storage pool volumes

You can recreate files in primary storage pool volumes by using copies in a copy storage pool.

Attention: You can also recreate active versions of client backup files in storage pool volumes by using duplicate copies in active-data pools. However, active-data pools should not be considered for recovery of a volume unless the loss of inactive data is acceptable. Restoring 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.

Task	Required Privilege Class
Restore volumes in any storage pool for which they have authority	System, unrestricted storage, or restricted storage

Use the RESTORE VOLUME command to restore all files that are stored in the same primary storage pool and that were previously backed up to copy storage pools.

The RESTORE VOLUME command generates a background process that can be canceled with the CANCEL PROCESS command. If a RESTORE VOLUME background process is canceled, some files may have already been restored prior to the cancellation. To display information on background processes, use the QUERY PROCESS command.

The RESTORE VOLUME command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

When you use the RESTORE VOLUME command, be prepared to supply some or all of the following information:

Volume name

Specifies the name of the volume in the primary storage pool for which to restore files.

Tip: To restore more than one volume in the same primary storage pool, issue this command once and specify a list of volumes to be restored. When you specify more than one volume, Tivoli Storage Manager attempts to minimize volume mounts for the copy storage pool.

Copy storage pool name

Specifies the name of the copy pool from which the files are to be restored. This information is optional. If you do not specify a particular copy storage pool, the files are restored from any copy storage pool where it can find them.

Active data only

Specifies that active versions of backup files are to be restored from active-data pools only. This information is optional. If it is not provided, files are restored from copy storage pools.

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.

Active-data pool

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 information is not provided, files are restored from any active-data pool in which active versions of backup files can be located.

New storage pool

Specifies the name of the new primary storage pool to which to restore the files. This information is optional. If you do not specify a new storage pool, the files are restored to the original primary storage pool.

Maximum number of processes

Specifies the maximum number of parallel processes that are used for restoring files.

Preview

Specifies whether you want to preview the restore operation without actually restoring data.

See “Recovering a lost or damaged storage pool volume” on page 796 for an example of using the RESTORE VOLUME command.

Volume restoration

When you restore a volume, the server obtains a copy of each file that was on the volume from a copy storage pool or active-data pool, and then stores the files on a different volume.

Attention: Cached copies of files in a disk storage pool are never restored. References to any cached files that reside on a volume that is being restored are removed from the database during restore processing.

After files are restored, the old references to these files in the primary storage pool are deleted from the database. Tivoli Storage Manager now locates these files on the volumes to which they were restored, rather than on the volume on which they were previously stored.

The RESTORE VOLUME command changes the access mode of the volumes being restored to *destroyed*. When the restoration is complete (when all files on the volume are restored to other locations), the destroyed volume is empty and is then automatically deleted from the database.

Fixing an incomplete volume restoration

When the restoration of a volume might be incomplete, you can get more information on the remaining files on volumes for which restoration was incomplete.

The restoration might be incomplete for one or more of the following reasons:

- Files were either never backed up or the backup copies are marked as damaged.
- A copy storage pool or active-data pool was specified on the RESTORE command, but files were backed up to a different copy storage pool or a different active-data pool. If you suspect this is a problem, use the RESTORE

command again without specifying a copy storage pool or active-data pool from which to restore files. The PREVIEW option can be used on the second RESTORE command, if you do not actually want to restore files.

- Volumes in the copy storage pool or active-data 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 or active-data pools were moved or deleted by other processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool volumes or active-data pool volumes while restore processing is in progress:
 - MOVE DATA
 - DELETE VOLUME (DISCARDDATA=YES)
 - AUDIT VOLUME (FIX=YES)
 - MIGRATE STGPOOL
 - RECLAIM STGPOOL

You can prevent reclamation processing for your copy storage pools and active-data pools by setting the RECLAIM parameter to 100 with the UPDATE STGPOOL command.

Auditing storage pool volumes

If there are inconsistencies between the information in the database about files in a storage pool volume and the files themselves, you might not be able to access the files.

Use this section to help you audit storage pool volumes for data integrity.

Task	Required Privilege Class
Audit volumes in storage pools over which they have authority	Restricted storage privilege
Audit a volume in any storage pool	System privilege, unrestricted storage privilege

To ensure that all files are accessible on volumes in a storage pool, audit any volumes you suspect may have problems by using the AUDIT VOLUME command. You have the option of auditing multiple volumes using a time range criteria, or auditing all volumes in a storage pool.

You should audit a volume when the following conditions are true:

- The volume is damaged.
- The volume has not been accessed for a long period of time, for example, after six months
- A read or write error occurs while accessing the volume
- The database has been restored to an earlier point-in-time, and the volume is either a disk volume or a volume that was identified as being reused or deleted since the database backup

If a storage pool has data validation enabled, run an audit for the volumes in the storage pool to have the server validate the data.

To display the results of a volume audit after it has completed, use the QUERY ACTLOG command. See “Requesting information from the activity log” on page 624

Storage pool volume audit

When you audit a volume, a background process is started, and the results of the audit are stored in the activity log.

During the auditing process, the server performs the following actions:

- Sends informational messages about processing to the server console.
- Prevents new files from being written to the volume.
- Generates a cyclic redundancy check, if data validation is enabled for the storage pool.

You can specify whether you want the server to correct the database if inconsistencies are detected. Tivoli Storage Manager corrects the database by deleting database records that refer to files on the volume that cannot be accessed. The default is to report inconsistencies that are found (files that cannot be accessed), but to not correct the errors.

If files with read errors are detected, their handling depends on the following conditions:

- The type of storage pool to which the volume is assigned
- The FIX option of the AUDIT VOLUME command
- The location of file copies (whether a copy of the file exists in a copy storage pool)

Errors in an audit of a primary storage pool volume

When an volume in a primary storage pool is audited, the setting of the FIX parameter determines how errors are handled.

The FIX parameter on an AUDIT VOLUME command can have the following effects:

FIX=NO

The server reports, but does not delete, any database records that refer to files found with logical inconsistencies. If the AUDIT VOLUME command detects a read error in a file, the file is marked as *damaged* in the database. You can do one of the following actions:

- If a backup copy of the file is stored in a copy storage pool, you can restore the file by using the RESTORE VOLUME or RESTORE STGPOOL command.
- If the file is a cached copy, you can delete references to the file on this volume by using the AUDIT VOLUME command again. Specify FIX=YES.

If the AUDIT VOLUME command does not detect a read error in a damaged file, the file state is reset, and the file can be used. For example, if a dirty tape head caused some files to be marked damaged, you can clean the head and then audit the volume to make the files accessible again.

FIX=YES

Any inconsistencies are fixed as they are detected.

If the AUDIT VOLUME command detects a read error in a file:

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

If the AUDIT VOLUME command does not detect a read error in a damaged file, the file state is reset, and the file can be used. For example, if a dirty tape head caused some files to be marked damaged, you can clean the head and then audit the volume to make the files accessible again.

Errors in an audit of copy storage pool volumes

When an volume in a copy storage pool is audited, the setting of the FIX parameter determines how errors are handled.

The FIX parameter on an AUDIT VOLUME command can have the following effects:

FIX=NO

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

FIX=YES

The server deletes references to the file on the audited volume from the database.

Errors in an audit of active-data storage pool volumes

When an volume in a active-data storage pool is audited, the setting of the FIX parameter determines how errors are handled.

The FIX parameter on an AUDIT VOLUME command can have the following effects:

FIX=NO

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

FIX=YES

The server deletes references to the file on the audited volume from the database. The physical file is deleted from the active-data pool.

When auditing a volume in an active-data pool, the server skips inactive files in aggregates that have been removed by reclamation. These files are not reported as skipped or marked as damaged.

Data validation during audit volume processing

Data validation for storage pools allows the server to validate that data sent to a device during a write operation matches what the server later reads.

Data validation is helpful if you have introduced new hardware devices. The validation assures that the data is not corrupt as it moves through the hardware, and then is written to the volume in the storage pool. You can use the DEFINE STGPOOL or UPDATE STGPOOL commands to enable data validation for storage pools.

When you enable data validation for an existing storage pool, the server validates data that is written from that time forward. The server does not validate existing data which was written to the storage pool before data validation was enabled.

When data validation is enabled for storage pools, the server generates a cyclic redundancy check (CRC) value and stores it with the data when it is written to the storage pool. The server validates the data when it audits the volume, by generating a cyclic redundancy check and comparing this value with the CRC value stored with the data. If the CRC values do not match, then the server processes the volume in the same manner as a standard audit volume operation. This process can depend on the following conditions:

- The type of storage pool to which the volume is assigned
- The FIX option of the AUDIT VOLUME command
- The location of file copies (whether a copy of the file exists in a copy storage pool or an active-data pool)

See “Errors in an audit of a primary storage pool volume” on page 783, “Errors in an audit of copy storage pool volumes” on page 784, and “Errors in an audit of active-data storage pool volumes” on page 784 for details on how the server handles inconsistencies detected during an audit volume process. Check the activity log for details about the audit operation.

The server removes the CRC values before it returns the data to the client node.

Choosing when to enable data validation

Data validation is available for nodes and storage pools. The forms of validation are independent of each other.

Figure 101 on page 786 shows data validation:

- During a client session with the server **2**
- During a client session with the storage agent **1** (the storage agent reads the VALIDATEPROTOCOL setting for the client from the Tivoli Storage Manager server)
- During a storage agent session with the server **3**
- When a server (including a storage agent) sends data to the storage pool **4** or **5**

You can enable data validation for one or more nodes, storage agents, or storage pools. Figure 101 on page 786 illustrates data transfer that is eligible for data validation within a Tivoli Storage Manager environment. Your environment may contain some or all of these objects.

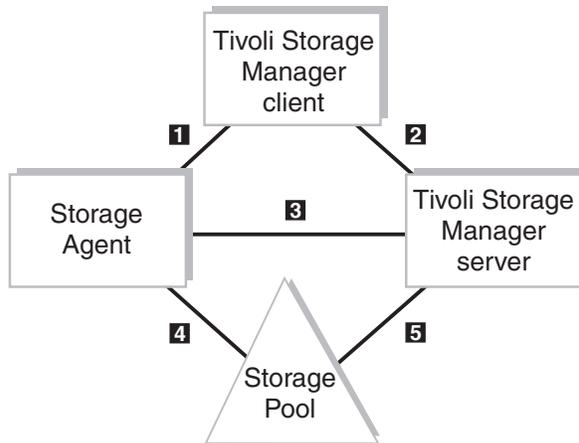


Figure 101. Data transfer eligible for data validation

Table 64 provides information that relates to Figure 101. This information explains the type of data being transferred and the appropriate command to issue.

Table 64. Setting data validation

Numbers in Figure 101	Where to Set Data Validation	Type of Data Transferred	Command	Command Parameter
1	Node definition	File Data and Metadata	See <i>Note</i>	See <i>Note</i>
2	Node definition	File Data and Metadata	REGISTER NODE UPDATE NODE	VALIDATEPROTOCOL= ALL or DATAONLY
3	Server definition (storage agent only)	Metadata	DEFINE SERVER UPDATE SERVER	VALIDATEPROTOCOL=ALL
4	Storage pool definition issued on the Tivoli Storage Manager server	File Data	DEFINE STGPOOL UPDATE STGPOOL	CRCDATA=YES ¹
5	Storage pool definition issued on the Tivoli Storage Manager server	File Data	DEFINE STGPOOL UPDATE STGPOOL	CRCDATA=YES ¹

¹ This parameter is not available for primary sequential access storage pools that use the following data formats: NETAPPDUMP, CELERRADUMP, NDMPDUMP.

Note: The storage agent reads the VALIDATEPROTOCOL setting for the client from the Tivoli Storage Manager server.

Figure 102 on page 787 is similar to the previous figure, however note that the top section encompassing **1**, **2**, and **3** is shaded. All three of these data validations are related to the VALIDATEPROTOCOL parameter. What is significant about this validation is that it is active only during the client session. After validation, the client and server discard the CRC values generated in the current

session. This is In contrast to storage pool validation, **4** and **5**, which is always active as long as the storage pool CRCDATA setting is equal to YES.

The validation of data transfer between the storage pool and the storage agent **4** is managed by the storage pool CRCDATA setting defined by the Tivoli Storage Manager server. Even though the flow of data is between the storage agent and the storage pool, data validation is determined by the storage pool definition. Therefore, if you always want your storage pool data validated, set your primary storage pool CRCDATA setting to YES.

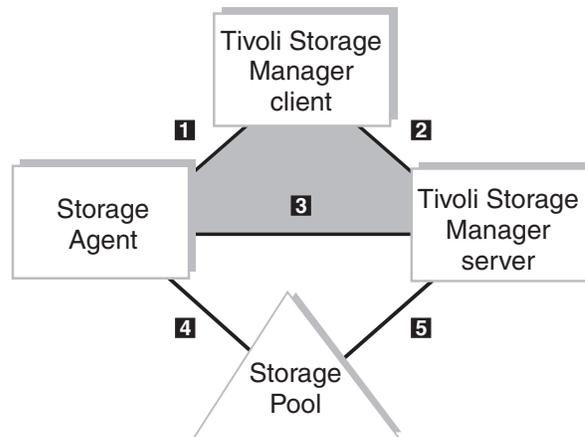


Figure 102. Protocol data validation versus storage pool data validation

If the network is unstable, you may decide to only enable data validation for nodes. Tivoli Storage Manager generates a cyclic redundancy check when the data is sent over the network to the server. Certain nodes may have more critical data than others and may require the assurance of data validation. When you identify the nodes that require data validation, you can choose to have only the user's data validated or all the data validated. Tivoli Storage Manager validates both the file data and the file metadata when you choose to validate all data. See "Validating a node's data during a client session" on page 496.

When you enable data validation for a server-to-server exchange or between a storage agent and server, the server must validate all data. You can enable data validation by using the DEFINE SERVER or UPDATE SERVER command. For a server-to-server exchange, see "Using virtual volumes to store data on another server" on page 712 for more information. For data that is exchanged between a storage agent and the server, refer to the *Storage Agent User's Guide* for the storage agent's operating system.

If the network is fairly stable but your site is perhaps using new hardware devices, you may decide to only enable data validation for storage pools. When the server sends data to the storage pool, the server generates cyclic redundancy checking, and stores the CRC value with the data. The server validates the CRC value when the server audits the volume. Later, you may decide that data validation for storage pools is no longer required after the devices prove to be stable. Refer to "Auditing storage pool volumes" on page 782 for more information on data validation for storage pools.

Performance considerations for data validation

Data validation affects performance because the server requires additional CPU overhead to calculate and compare CRC values.

Consider the impact on performance when you decide whether data validation is necessary for storage pools. This method of validation is independent of validating data during a client session with the server. When you choose to validate storage pool data, there is no performance impact on the client.

If you enable CRC for storage pools on devices that later prove to be stable, you can increase performance by updating the storage pool definition to disable data validation.

Performing storage pool data validation

The `AUDIT VOLUME` command allows you to specify an audit for data written to volumes within a range of days, or to run an audit for a given storage pool.

You can manage when the validation of data in storage pools occurs by scheduling the audit volume operation. You can choose a method suitable to your environment, for example:

- Select volumes at random to audit. A random selection does not require significant resources or cause much contention for server resources but can provide assurance that the data is valid.
- Schedule a daily audit of all volumes written in the last day. This method validates data written to a given storage pool on a daily basis.
- Audit volumes in storage pools only for client nodes that are considered to be critical users.

Auditing a disk storage pool volume

When you audit a disk storage pool volume, you can specify that only summary messages are sent to the activity log and server console.

To display the results of a volume audit after it has completed, you can issue the `QUERY ACTLOG` command.

To specify that only summary messages for `/dev/vol1` are sent to the activity log and server console, issue the following command:

```
audit volume /dev/vol1 quiet=yes
```

The audit volume process is run in the background and the server returns the following message:

```
ANR2313I Audit Volume NOFIX process started for volume /dev/vol1  
(process id 4).
```

To view the status of the audit volume process, issue the following command:
`query process 4`

Here is an example of the audit volume process report.

Process Number	Process Description	Status
4	Audit Volume (Inspect Only)	Storage Pool BACKUPPOOL, Volume /dev/vol1, Files Processed: 680, Irretrievable Files Found: 0, Partial Files Skipped: 0

Auditing multiple volumes in a sequential access storage pool

When you audit a sequential storage volume containing files that span multiple volumes, the server selects all associated volumes.

The server then begins the audit process with the first volume on which the first file resides. For example, Figure 103 shows five volumes defined to ENGBACK2. In this example, File A spans VOL1 and VOL2, and File D spans VOL2, VOL3, VOL4, and VOL5.

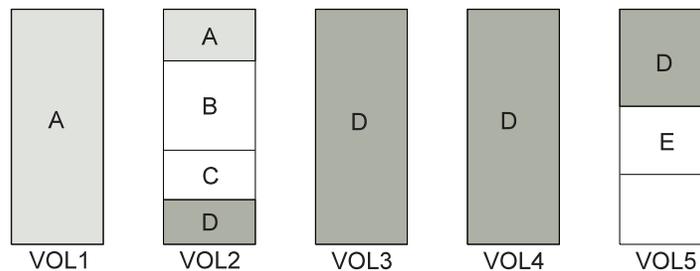


Figure 103. Tape volumes with files a, b, c, d, and e

If you request that the server audit volume VOL3, the server first accesses volume VOL2, because File D begins at VOL2. When volume VOL2 is accessed, the server *only* audits File D. It does not audit the other files on this volume.

Because File D spans multiple volumes, the server accesses volumes VOL2, VOL3, VOL4, and VOL5 to ensure that there are no inconsistencies between the database and the storage pool volumes.

For volumes that require manual mount and dismount operations, the audit process can require significant manual intervention.

Auditing a single volume in a sequential access storage pool

To audit a single volume in a sequential storage pool, request that the server skip any files that span multiple volumes.

This option is useful when the volume you want to audit contains part of a file, the rest of which resides on a different, damaged volume. For example, to audit only volume VOL5 in the example in Figure 103 and have the server fix any inconsistencies found between the database and the storage volume, enter:

```
audit volume vol5 fix=yes skippartial=yes
```

Auditing volumes by date written

You can limit the audit to volumes that were written in a certain time range.

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. When you include the STGPOOL parameter you limit the number of volumes that may include disk volumes.

Issue the AUDIT VOLUME command with the FROMDATE and TODATE parameters.

For example, to audit the volumes in storage pool BKPOOL1 for volumes written from March 20, 2002 to March 22, 2002.

```
audit volume stgpool=bkppool1 fromdate=03/20/2002 todate=03/22/2002
```

The server audits all volumes that were written to starting at 12:00:01 a.m. on March 20 and ending at 11:59:59 p.m. on March 22, 2002.

Auditing volumes in a specific storage pool

You can limit the audit to volumes in a specified storage pool.

For example, you can audit the volumes in storage pool BKPOOL1 by issuing the following command:

```
audit volume stgpool=bkppool1
```

Scheduling volume audits

You can schedule periodic volume audits.

For example, if your critical users store data in storage pool STPOOL3 and you want all volumes in the storage pool audited every 2 days at 9:00 p.m., issue the following command:

```
define schedule crcstg1 type=administrative  
cmd='audit volume stgpool=stgpool3' active=yes starttime=21:00 period=2
```

Fixing damaged files

If files are marked as damaged, there are some measure you can take to correct them.

A data error, which results in a file being unreadable, can be caused by such things as a tape deteriorating or being overwritten or by a drive needing cleaning. If a data error is detected when a client tries to restore, retrieve, or recall a file or during a volume audit, the file is marked as damaged. If the same file is stored in other copy storage pools or active-data pools, the status of those file copies is not changed.

If a client tries to access a damaged file and an undamaged copy is available on an onsite copy storage pool volume or active-data pool volume, the server sends the user the undamaged copy.

If files are marked as damaged, you can perform the following operations on them:

- Restore, retrieve, or recall the files
- Move the files by migration, reclamation, or the MOVE DATA command
- Back up during a BACKUP STGPOOL operation if the primary file is damaged

- Restore during a RESTORE STGPOOL or RESTORE VOLUME operation if the backup copy in a copy storage pool or active-data pool volume is damaged
- Migrate or reclaim during migration and reclamation

Ensuring the integrity of files

There are steps that you can take to ensure the data integrity of user files.

To maintain the data integrity of user files, you can perform the following steps:

1. Detect damaged files before the users do. The AUDIT VOLUME command marks a file as damaged if a read error is detected for the file. If an undamaged copy is in an onsite copy storage pool or an active-data pool volume, it is used to provide client access to the file. See “Data validation during audit volume processing” on page 784
2. Reset the damaged status of files if the error that caused the change to damaged status was temporary. You can use the AUDIT VOLUME command to correct situations when files are marked damaged due to a temporary hardware problem, such as a dirty tape head. The server resets the damaged status of files if the volume in which the files are stored is audited and no read errors are detected.
3. Correct files that are marked as damaged. If a primary file copy is marked as damaged and a usable copy exists in a copy storage pool or an active-data pool volume, the primary file can be corrected using the RESTORE VOLUME or RESTORE STGPOOL command. For an example, see “Restoring damaged files”
4. Regularly run commands to identify files that are marked as damaged:
 - The RESTORE STGPOOL command displays the name of each volume in the restored storage pool that contains one or more damaged primary files. Use this command with the preview option to identify primary volumes with damaged files without actually performing the restore operation.
 - The QUERY CONTENT command with the DAMAGED option lets you display damaged files on a specific volume.

For an example of how to use these commands, see “Restoring damaged files.”

Restoring damaged files

If you use copy storage pools, you can restore damaged client files. You can also check storage pools for damaged files and restore the files.

This section explains how to restore damaged files based on the scenario in “Scenario: scheduling a backup with one copy storage pool” on page 764.

If a client tries to access a file stored in TAPEPOOL and a read error occurs, the file in TAPEPOOL is automatically marked as damaged. Future accesses to the file automatically use the copy in COPYPOOL as long as the copy in TAPEPOOL is marked as damaged.

To restore any *damaged* files in TAPEPOOL, you can define a schedule that issues the following command periodically:

```
restore stgpool tapepool
```

You can check for and replace any files that develop data-integrity problems in TAPEPOOL or in COPYPOOL. For example, every three months, query the volumes in TAPEPOOL and COPYPOOL by entering the following commands:

```
query volume stgpool=tapepool
```

```
query volume stgpool=copypool
```

Then issue the following command for each volume in TAPEPOOL and COPYPOOL:

```
audit volume <volname> fix=yes
```

If a read error occurs on a file in TAPEPOOL, that file is marked *damaged* and an error message is produced. If a read error occurs on file in COPYPOOL, that file is deleted and a message is produced.

Restore *damaged* primary files by entering:

```
restore stgpool tapepool
```

Finally, create new copies in COPYPOOL by entering:

```
backup stgpool tapepool copypool
```

Backup and recovery scenarios

This section presents scenarios for protecting and recovering a Tivoli Storage Manager server. You can modify the procedures to meet your needs.

DRM: DRM can help you track your onsite and offsite primary and copy storage pool volumes. DRM can also query the server and generate a current, detailed disaster recovery plan for your installation.

These scenarios assume a storage hierarchy consisting of:

- The default random access storage pools (BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL)
- TAPEPOOL, a tape storage pool

To provide additional levels of protection for client data, the scenarios also specify an offsite copy storage pool and an onsite active-data pool.

Protecting the database and storage pools

This scenario describes the steps a company takes to protect against the permanent loss of data stored in the database and storage pools.

The company's standard procedures include the following activities:

- Weekly reclamation of its copy storage pool. Reclamation for the copy storage pools is turned off at other times

Note: In a copy storage pool definition, the REUSEDELAY parameter delays volumes from being returned to scratch or being reused. Set the value high enough to ensure that the database can be restored to an earlier point in time and that database references to files in the storage pool are valid. For example, to retain database backups for seven days and, therefore, sets REUSEDELAY to 7.

- Nightly back up of its primary storage pools to the copy storage pool. Every night, copy the active client backup data in the primary storage pools to the active-data pool
- A weekly full backup of the database and incremental backups on the other days
- Daily shipment of the database backup volumes and copy storage pool volumes to an offsite location

To protect client data, perform the following actions:

1. Create a copy storage pool named DISASTER-RECOVERY. Only scratch tapes are used, and the maximum number of scratch volumes is set to 100. The copy storage pool is defined by entering:

```
define stgpool disaster-recovery tapeclass pooltype=copy
maxscratch=100
```

2. Create an active-data pool named CLIENT-RESTORE and associates it with a sequential-access disk device class (FILE). Only scratch volumes are used, and the maximum number of scratch volumes is set to 50. The active-data pool is defined by entering:

```
define stgpool client-restore diskclass pooltype=activedata
maxscratch=50
```

3. Perform the first backup of the primary storage pools. The first backup of a primary storage pool is a full backup and, depending on the size of the storage pool, could take a long time.
4. Define schedules for the following daily operations:

- a. Run incremental backups of the primary storage pools each night. Issue the following commands:

```
backup stgpool backuppool disaster-recovery maxprocess=2
copy activedata backuppool client-restore maxprocess=2
backup stgpool archivepool disaster-recovery maxprocess=2
backup stgpool spacemgpool disaster-recovery maxprocess=2
backup stgpool tapepool disaster-recovery maxprocess=2
copy activedata tapepool client-restore maxprocess=2
```

The BACKUP STGPOOL commands use multiple, parallel processes to perform an incremental backup of each primary storage pool to the copy storage pool. The COPY ACTIVATEDATA commands use multiple, parallel processes to copy the active versions of client backup data to the active-data pool. Only those files for which a copy does not already exist in the copy pool or active-data pool are backed up.

Migration should be turned off during the rest of the day. You could add a schedule to migrate from disk to tape at this point. In this way, the backups are done while the files are still on disk.

- b. Change the access mode to OFFSITE for copy storage pool volumes that have read-write or read-only access, are onsite, and are at least partially filled. This is done by issuing the following command:

```
update volume * access=offsite location='vault site info'
wherestgpool=disaster-recovery whereaccess=readwrite,readonly
wherestatus=filling,full
```

- c. Back up the database. For example, issue the following command:

```
backup db type=incremental devclass=tapeclass scratch=yes
```

5. Perform the following operations nightly after the scheduled operations have completed:

- a. Back up the volume history and device configuration files. If they have changed, back up the server options files and the database and recovery log setup information.

- b. Move the copy storage pool volumes marked offsite, the database backup volumes, volume history files, device configuration files, server options files and the database and recovery log setup information to the offsite location.

- c. Identify offsite volumes that should be returned onsite. For example, issue the following command:

```
query volume stgpool=disaster-recovery access=offsite status=empty
```

These volumes, which have become empty through expiration, reclamation, and file space deletion, have waited the delay time specified by the REUSEDELAY parameter. The administrator periodically returns outdated backup database volumes. These volumes are displayed with the QUERY VOLHISTORY command and can be released for reuse with the DELETE VOLHISTORY command.

6. Bring the volumes identified in step 5c on page 793 onsite and update their access to read-write.

Recovering to a point-in-time from a disaster

In this scenario, an administrator restores the server to the point-in-time of the last backup.

The processor on which Tivoli Storage Manager resides, the database, and all onsite storage pool volumes are destroyed by fire. You can use either full and incremental backups or snapshot database backups to restore a database to a point-in-time.

DRM: DRM can help you perform these steps.

Do the following steps:

1. Install Tivoli Storage Manager on the replacement processor with the same server options and the same size database and recovery log as on the destroyed system.
2. Move the latest backup and all of the DISASTER-RECOVERY volumes onsite from the offsite location.

Note: Do not change the access mode of these volumes until after you have completed step 7.

3. If a current, undamaged volume history file exists, save it.
4. Restore the volume history and device configuration files, the server options, and the database and recovery log setup. For example, the recovery site might require different device class, library, and drive definitions. For more information, see “Updating the device configuration file” on page 776.
5. Restore the database from the latest backup level by issuing the DSMSEV RESTORE DB utility (see “Recovering the server using database and storage pool backups” on page 772).
6. Change the access mode of all the existing primary storage pool volumes in the damaged storage pools to DESTROYED. For example, issue the following commands:

```
update volume * access=destroyed wherestgpool=backuppool
update volume * access=destroyed wherestgpool=archivepool
update volume * access=destroyed wherestgpool=spacemgpool
update volume * access=destroyed wherestgpool=tapepool
```
7. Issue the QUERY VOLUME command to identify any volumes in the DISASTER-RECOVERY storage pool that were onsite at the time of the disaster. Any volumes that were onsite would have been destroyed in the disaster and could not be used for restore processing. Delete each of these volumes from the database by using the DELETE VOLUME command with the DISCARDATA option. Any files backed up to these volumes cannot be restored.
8. Change the access mode of the remaining volumes in the DISASTER-RECOVERY pool to READWRITE. Issue the following command:

```
update volume * access=readwrite wherestgpool=disaster-recovery
```

At this point, clients can access files. If a client tries to access a file that was stored on a destroyed volume, the retrieval request goes to the copy storage pool. In this way, clients can restore their files without waiting for the primary storage pool to be restored. When you update volumes brought from offsite to change their access, you greatly speed recovery time.

9. Define new volumes in the primary storage pool so the files on the damaged volumes can be restored to the new volumes. The new volumes also let clients backup, archive, or migrate files to the server. You do not need to perform this step if you use only scratch volumes in the storage pool.
10. Restore files in the primary storage pool from the copies located in the DISASTER-RECOVERY pool. To restore files from DISASTER-RECOVERY pool, issue the following commands:

```
restore stgpool backuppool maxprocess=2
restore stgpool tapepool maxprocess=2
restore stgpool archivepool maxprocess=2
restore stgpool spacemgpool maxprocess=2
```

These commands use multiple parallel processes to restore files to primary storage pools. After all the files have been restored for a destroyed volume, that volume is automatically deleted from the database. See “Fixing an incomplete storage pool restoration” on page 779 for what to do if one or more volumes cannot be fully restored.

11. To repopulate the active-data pool, copy active versions of backup data from a primary storage pool to an active-data pool. For example, issue the following commands:

```
copy activedata backuppool client-restore maxprocess=2
copy activedata tapepool client-restore maxprocess=2
```
12. To ensure against another loss of data, immediately back up all storage volumes and the database. Then resume normal activity, including weekly disaster backups and movement of data to the offsite location.

Restoring to a point-in-time in a shared library environment

A point-in-time restore for a library manager server or a library client server requires additional steps to ensure the consistency of the volume inventories of the affected servers.

This section describes the procedures for the two possible scenarios.

Restoring to a point-in-time a library manager server:

A point-in-time restore of a library manager server could create inconsistencies between the volume inventories of the library manager and library client servers. Steps must be taken to prevent this problem.

The restore removes all library client server transactions that occurred after the point in time from the volume inventory of the library manager server. The volume inventory of the library client server, however, still contains those transactions. New transactions could then be written to these volumes, resulting in a loss of client data. Do the following after the restore:

1. Halt further transactions on the library manager server: Disable all schedules, migration and reclamations on the library client and library manager servers.
2. Audit all libraries on all library client servers. The audits will re-enter those volume transactions that were removed by the restore on the library manager

server. You should audit the library clients from the oldest to the newest servers. Use the volume history file from the library client and library manager servers to resolve any conflicts.

3. Delete the volumes from the library clients that do not own the volumes.
4. Resume transactions by enabling all schedules, migration, and reclamations on the library client and library manager servers.

Restoring to a point-in-time a library client server:

A point-in-time restore of a library client server could cause volumes to be removed from the volume inventory of a library client server and later overwritten.

If a library client server acquired scratch volumes after the point-in-time to which the server is restored, these volumes would be set to private in the volume inventories of the library client and library manager servers. After the restore, the volume inventory of the library client server can be regressed to a point-in-time before the volumes were acquired, thus removing them from the inventory. These volumes would still exist in the volume inventory of the library manager server as private volumes owned by the client.

The restored volume inventory of the library client server and the volume inventory of the library manager server would be inconsistent. The volume inventory of the library client server must be synchronized with the volume inventory of the library manager server in order to return those volumes to scratch and enable them to be overwritten. To synchronize the inventories, do the following steps:

1. Audit the library on the library client server to synchronize the volume inventories of the library client and library manager servers.
2. To resolve any remaining volume ownership concerns, refer to the volume history and issue the UPDATE VOLUME command as needed.

Recovering a lost or damaged storage pool volume

If a company makes the necessary preparations, it can recover from a media loss.

These preparations are described in “Protecting the database and storage pools” on page 792. In the following scenario, an operator inadvertently destroys a tape volume (DSM087) belonging to the TAPEPOOL storage pool. An administrator performs the following actions to recover the data stored on the destroyed volume by using the offsite copy storage pool:

1. Determine the copy pool volumes that contain the backup copies of the files that were stored on the volume that was destroyed. Issue this command: by entering:

```
restore volume dsm087 preview=yes
```

This command produces a list of offsite volumes that contain the backed up copies of the files that were on tape volume DSM087.

2. Set the access mode of the copy volumes identified as UNAVAILABLE to prevent reclamation.

Note: This precaution prevents the movement of files stored on these volumes until volume DSM087 is restored.

3. Bring the identified volumes to the onsite location and set their access mode to READONLY to prevent accidental writes. If these offsite volumes are being used in an automated library, the volumes must be checked into the library when they are brought back onsite.
4. Restore the destroyed files. Issue this command:

```
restore volume dsm087
```

This command sets the access mode of DSM087 to DESTROYED and attempts to restore all the files that were stored on volume DSM087. The files are not actually restored to volume DSM087, but to another volume in the TAPEPOOL storage pool. All references to the files on DSM087 are deleted from the database and the volume itself is deleted from the database.
5. Set the access mode of the volumes used to restore DSM087 to OFFSITE using the UPDATE VOLUME command.
6. Set the access mode of the restored volumes, that are now onsite, to READWRITE.
7. Return the volumes to the offsite location. If the offsite volumes used for the restoration were checked into an automated library, these volumes must be checked out of the automated library when the restoration process is complete.

Restoring a library manager database

This scenario describes how a library manager's corrupted database can be restored.

In a Tivoli Storage Manager shared library environment, the server that manages and controls the shared library is known as the library manager. The library manager maintains a database of the volumes within the shared library. Perform the following steps to restore the corrupted database:

1. Rename and save a copy of the volume history file. After the database is restored, any volume history information that is pointed to by the server options is lost. You will need this information to identify the volumes to be audited.
2. Put the device configuration file and the server options file in the server working directory. You can no longer recreate the device configuration file; you must have a copy of the original.
3. Gather the outputs from your detailed queries about your database and recovery log setup information.
4. Determine whether the original database and recovery log are present. If the original database or recovery log directories were lost, recreate them using the operating system mkdir command.
5. Issue the DSMSERV RESTORE DB utility.
6. Start the library manager.
7. Issue an AUDIT LIBRARY command from each library client for each shared library.
8. Create a list from the old volume history information (generated by the QUERY VOLHISTORY command) that shows all of the volumes that were reused (STGREUSE), added (STGNEW), and deleted (STGDELETE) since the original backup. Use this list to perform the rest of this procedure.
9. Audit all disk volumes, all reused volumes, and any deleted volumes located by the AUDIT VOLUME command using the FIX=YES parameter.

10. Issue the RESTORE STGPOOL command to restore those files detected as damaged by the audit. Include the FIX=YES parameter on the AUDIT VOLUME command to delete database entries for files not found in the copy storage pool or active-data pool.
11. Mark as destroyed any volumes that cannot be located, and recover those volumes from copy storage pool backups. Recovery from active-data pool volumes is not recommended unless the loss of inactive data is acceptable. If no backups are available, delete the volumes from the database by using the DELETE VOLUME command with the DISCARDDATA=YES parameter.
12. Redefine any storage pool volumes that were added since the database backup.

Note: When a database is loaded or restored, the server-to-server communication verification token is changed. The verification token is an attribute of the database and is not stored in the database itself. Part of the token is the install date and time for the database. For servers that have been defined for server-to-server communications, issue an UPDATE SERVER command with FORCESYNC=YES.

Restoring a library client database

This scenario describes how a library client's corrupted database might be restored.

In a Tivoli Storage Manager shared library environment, the servers that share a library and rely on a library manager to coordinate and manage the library's usage are known as library clients. Each library client maintains a database of volume usage and volume history. If the library client's database becomes corrupted, it might be restored by following these steps:

1. Rename and save a copy of the volume history file. After the database is restored, any volume history information pointed to by the server options is lost. You will need this information to identify the volumes to be audited.
2. Put the device configuration file and the server options file in the server working directory. You can no longer recreate the device configuration file; you must have a copy of the original.
3. Gather the outputs from your detailed queries about your database and recovery log setup information.
4. Check to see if the original database and recovery log are present. If the original database or recovery log directories were lost, recreate them using the operating system mkdir command.
5. Issue the DSMSEV RESTORE DB utility.
6. Create a list from the old volume history information (generated by the QUERY VOLHISTORY command) that shows all of the volumes that were reused (STGREUSE), added (STGNEW), and deleted (STGDELETE) since the original backup. Use this list to perform the rest of this procedure.
7. Audit all disk volumes, all reused volumes, and any deleted volumes located by the AUDIT VOLUME command using the FIX=YES parameter.
8. Issue the RESTORE STGPOOL command to restore those files detected as damaged by the audit. Include the FIX=YES parameter on the AUDIT VOLUME command to delete database entries for files not found in the copy storage pool.
9. Mark as destroyed any volumes that cannot be located, and recover those volumes from copy storage pool backups. If no backups are available, delete the volumes from the database by using the DELETE VOLUME command with the DISCARDDATA=YES parameter.

10. Issue the `AUDIT LIBRARY` command for all shared libraries on this library client.
11. Redefine any storage pool volumes that were added since the database backup.

Note: When a database is loaded or restored, the server-to-server communication verification token is changed. The verification token is an attribute of the database and is not stored in the database itself. Part of the token is the install date and time for the database. For servers that have been defined for server-to-server communications, issue an `UPDATE SERVER` command with `FORCESYNC=YES`.

Chapter 26. Using disaster recovery manager

You can use the disaster recovery manager (DRM) function to prepare a plan that can help you to recover your applications if a disaster occurs.

You can recover at an alternate site, on replacement computer hardware, and with people who are not familiar with the applications. You can also manage your offsite recovery media, store client recovery information, and use the disaster recovery plan for audits to certify the recoverability of the server.

To recover from a disaster, you must know the location of your offsite recovery media. DRM helps you to determine which volumes to move offsite and back onsite and track the location of the volumes.

Tasks
"Querying defaults for the disaster recovery plan file"
"Specifying recovery instructions for your site" on page 807
"Specifying information about your server and client node machines" on page 808
"Specifying recovery media for client machines" on page 810
"Creating and storing the disaster recovery plan" on page 810
"Managing disaster recovery plan files stored on target servers" on page 812
"Moving backup media" on page 814
"Summary of disaster recovery manager daily tasks" on page 819
"Staying prepared for a disaster" on page 821
"Recovering from a disaster" on page 822
Disaster Recovery Reference:
"Disaster recovery manager checklist" on page 831
"The disaster recovery plan file" on page 836

Before using DRM, familiarize yourself with Chapter 25, "Protecting and recovering your server," on page 755.

Note: Unless otherwise noted, to perform the tasks described here requires system privilege class.

Querying defaults for the disaster recovery plan file

DRM provides default settings for the preparation of the recovery plan file and for the management of offsite recovery media.

To query the settings, issue the following command:

```
query drmstatus
```

The output will be similar to the following:

```

Recovery Plan Prefix: /u/recovery/plans/rpp
Plan Instructions Prefix: /u/recovery/plans/source/
Replacement Volume Postfix: @
Primary Storage Pools: PRIM1 PRIM2
Copy Storage Pools: COPY*
Active-data Storage Pools: ACTIVEPOOL
Not Mountable Location Name: Local
Courier Name: Joe's Courier Service
Vault Site Name: Ironvault, D. Lastname, 1-000-000-0000
DB Backup Series Expiration Days: 30 Day(s)
Recovery Plan File Expiration Days: 60 Day(s)
Check Label?: Yes
Process FILE Device Type?: No
Command File Name: /drm/orm/exec.cmds

```

Specifying defaults for the disaster recovery plan file

You can override the default settings in the recovery plan file.

The following table describes how to set defaults for the disaster recovery plan file.

Table 65. Defaults for the disaster recovery plan file

Process	Default
Primary storage pools to be processed	<p>When the recovery plan file is generated, you can limit processing to specified pools. The recovery plan file will not include recovery information and commands for storage pools with a data format of NETAPPDUMP.</p> <p>The default at installation: All primary storage pools.</p> <p>To change the default: SET DRMPRIMSTGPOOL</p> <p>For example, to specify that only the primary storage pools named PRIM1 and PRIM2 are to be processed, enter:</p> <pre>set drmprimstgpool prim1,prim2</pre> <p>Note: To remove all previously specified primary storage pool names and thus select all primary storage pools for processing, specify a null string ("") in SET DRMPRIMSTGPOOL.</p> <p>To override the default: Specify primary storage pool names in the PREPARE command</p>
Copy storage pools to be processed	<p>When the recovery plan file is generated, you can limit processing to specified pools.</p> <p>The default at installation: All copy storage pools.</p> <p>To change the default: SET DRMCOPYSTGPOOL</p> <p>For example, to specify that only the copy storage pools named COPY1 and COPY2 are to be processed, enter:</p> <pre>set drmcopystgpool copy1,copy2</pre> <p>To remove any specified copy storage pool names, and thus select all copy storage pools, specify a null string ("") in SET DRMCOPYSTGPOOL. If you specify both primary storage pools (using the SET DRMPRIMSTGPOOL command) and copy storage pools (using the SET DRMCOPYSTGPOOL command), the specified copy storage pools should be those used to back up the specified primary storage pools.</p> <p>To override the default: Specify copy storage pool names in the PREPARE command</p>

Table 65. Defaults for the disaster recovery plan file (continued)

Process	Default
<p>Active-data pools to be processed</p>	<p>When the recovery plan file is generated, you can limit processing to specified pools.</p> <p>The default at installation: None</p> <p>To specify the default: SET DRMACTIVEDATASTGPOOL</p> <p>For example, to specify that only the active-data pools named ACTIVEPOOL1 and ACTIVEPOOL2 are to be processed, enter:</p> <pre>set drmactivedatastgpool activepool1,activepool2</pre> <p>To remove any specified active-data pool names, specify a null string ("") in SET DRMACTIVEDATASTGPOOL.</p> <p>Active-data pool volumes in MOUNTABLE state are processed only if you specify the active-data pools using the SET DRMACTIVEDATASTGPOOL command or the ACTIVEPOOL parameter on the MOVE DRMEDIA, QUERY DRMEDIA, and PREPARE commands. Processing of active-data pool volumes in MOUNTABLE state is different than the processing of copy storage pool volumes in MOUNTABLE state. All MOUNTABLE copy storage pool volumes are processed regardless whether you specify copy storage pools with either the SET DRMCOPYSTGPOOL command or the COPYSTGPOOL parameter.</p> <p>If you do not issue the SET DRMACTIVEDATASTGPOOL command or if you use this command to remove the names of all active-data storage pools, the Tivoli Storage Manager server processes active-data pool volumes specified using the ACTIVEPOOL parameter:</p> <ul style="list-style-type: none"> • MOVE DRMEDIA and QUERY DRMEDIA: The server processes all active-data pool volumes except those in MOUNTABLE state. • PREPARE: The server processes only the active-data pool volumes that are marked onsite at the time the PREPARE command is run. These volumes are marked UNAVAILABLE. <p>To override the default: Specify active-data pool names using the MOVE DRMEDIA, QUERY DRMEDIA, or PREPARE command.</p>
<p>Identifier for replacement volume names</p>	<p>To restore a primary storage pool volume, mark the original volume <i>destroyed</i> and create a replacement volume having a unique name. You can specify a character to be appended to the name of the original volume in order to create a name for the replacement volume. This character can help you find the replacement volume names in the disaster recovery plan.</p> <p>The default identifier at installation: @</p> <p>To change the default: SET DRMPLANVPOSTFIX</p> <p>For example, to use the character r, enter:</p> <pre>set drmplanvpostfix r</pre>

Table 65. Defaults for the disaster recovery plan file (continued)

Process	Default
Recovery instructions prefix	<p>You can specify a prefix for the names of the recovery instructions source files in the recovery plan file.</p> <p>The default at installation: For a description of how DRM determines the default prefix, see the INSTRPREFIX parameter of the PREPARE command section in the <i>Administrator's Reference</i> or enter HELP PREPARE from administrative client command line.</p> <p>To set a default: SET DRMINSTRPREFIX</p> <p>For example, to specify the prefix, enter: <pre>set drminstrprefix /u/recovery/plans/rpp</pre></p> <p>The disaster recovery plan file will include, for example, the following file: <pre>/u/recovery/plans/rpp.RECOVERY.INSTRUCTIONS.GENERAL</pre></p> <p>To override the default: The INSTRPREFIX parameter with the PREPARE command</p>
Prefix for the recovery plan file	<p>You can specify a prefix to the path name of the recovery plan file. DRM uses this prefix to identify the location of the recovery plan file and to generate the macros and script file names included in the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE and RECOVERY.SCRIPT.NORMAL.MODE stanzas.</p> <p>The default at installation: For a description of how DRM determines the default prefix, see the PLANPREFIX parameter of the PREPARE command section in the <i>Administrator's Reference</i> or enter HELP PREPARE from administrative client command line.</p> <p>To change the default: SET DRMPPLANPREFIX</p> <p>For example, to specify the prefix, enter the following command: <pre>set drmpplanprefix /u/server/recoveryplans/</pre></p> <p>The disaster recovery plan file name created by PREPARE processing will be in the following format: <pre>/u/server/recoveryplans/20000603.013030</pre></p> <p>To override the default: The PLANPREFIX parameter with the PREPARE command</p>
The disaster recovery plan expiration period	<p>You can set the numbers of days after creation that a disaster recovery plan file stored on a target server expires. After the number of days has elapsed, all recovery plan files that meet both of the following conditions are eligible for expiration:</p> <ul style="list-style-type: none"> • The last recovery plan associated with the database series is older than the set number of days. • The recovery plan file is not associated with the most recent backup series. <p>The default at installation: 60 days</p> <p>To change the default: SET DRMRPFEXPIREDAYS</p> <p>For example, to change the time to 90 days, enter: <pre>set drmrpfexpiredays 90</pre></p>

Specifying defaults for offsite recovery media management

You can set defaults for offsite recovery media management.

Table 66. Defaults for offsite recovery media management

Process	Default
Copy storage pool volumes to be processed	<p>MOVE DRMEDIA and QUERY DRMEDIA can process copy storage pool volumes in the MOUNTABLE state. You can limit processing to specified copy storage pools.</p> <p>The default at installation: All copy storage pool volumes in the MOUNTABLE state</p> <p>To change the default: SET DRMCOPYSTGPOOL</p> <p>To override the default: COPYSTGPOOL parameter on MOVE DRMEDIA or QUERY DRMEDIA</p>
Active-data pool volumes to be processed	<p>MOVE DRMEDIA and QUERY DRMEDIA can process active-data pool volumes except those in the MOUNTABLE state. You can limit processing to specified active-data pools.</p> <p>The default at installation: None</p> <p>To specify the default: SET DRMACTIVEDATASTGPOOL</p> <p>To override the default: ACTIVEDATASTGPOOL parameter on MOVE DRMEDIA, QUERY DRMEDIA, or PREPARE</p>
Executable commands file name	<p>You can use MOVE DRMEDIA or QUERY DRMEDIA to generate executable commands and store them in a file.</p> <p>The default file name at installation: None</p> <p>To set a default: SET DRMCMDFILENAME. For example, enter: <pre>set drmcmdfilename /drm/orm/exec.cmds</pre></p> <p>To override the default: CMDFILENAME parameter on MOVE DRMEDIA or QUERY DRMEDIA</p>
Location name for volumes that move to the NOTMOUNTABLE state	<p>MOVE DRMEDIA generates a location name for volumes that move to the NOTMOUNTABLE state.</p> <p>The default at installation: NOTMOUNTABLE</p> <p>To change the default: SET DRMNOTMOUNTABLENAME</p> <p>For example, to specify a location named LOCAL, enter: <pre>set drmnotmountablename local</pre></p>
Location name for volumes that move to the COURIER or COURIERRETRIEVE state	<p>MOVE DRMEDIA generates a location name for volumes that are changing from NOTMOUNTABLE to COURIER or from VAULTRETRIEVE to COURIERRETRIEVE.</p> <p>The default at installation: COURIER</p> <p>To change the default: SET DRMCOURIERNAME</p> <p>For example, to specify a courier named Joe's Courier Service, enter: <pre>set drmcouriername "Joe's Courier Service"</pre></p>

Table 66. Defaults for offsite recovery media management (continued)

Process	Default
Reading labels of checked out volumes	<p>To determine whether DRM reads the sequential media labels of volumes that are checked out with MOVE DRMEDIA. Note: This command does not apply to 349X library types.</p> <p>The default at installation: DRM reads the volume labels.</p> <p>To change the default: SET DRMCHECKLABEL</p> <p>For example, to specify that DRM should not read the volume labels, enter: set drmchecklabel no</p>
Expiration period of a database backup series	<p>A database backup series (full plus incremental and snapshot) is eligible for expiration if all of these conditions are true:</p> <ul style="list-style-type: none"> • The volume state is VAULT or the volume is associated with a device type of SERVER (for virtual volumes). • It is not the most recent database backup series. • The last volume of the series exceeds the expiration value, number of days since the last backup in the series. <p>The default at installation: 60 days</p> <p>To change the default: SET DRMDBBACKUPEXPIREDAYS</p> <p>For example, to set the expiration value to 30 days, enter: set drmdbbackupexpiredays 30</p>
Whether to process copy storage pool and active-data pool volumes of the FILE device type	<p>At installation, MOVE DRMEDIA and QUERY DRMEDIA will not process copy storage pool or active-data pool volumes that are associated with a device type of FILE.</p> <p>The default at installation: Copy storage pool and active-data pool volumes of the FILE device type are not processed</p> <p>To change the default: SET DRMFILEPROCESS</p> <p>To allow processing, enter: set drmfileprocess yes</p>
Vault Name	<p>MOVE DRMEDIA uses the vault name to set the location of volumes that are moving from the COURIER state to the VAULT state</p> <p>The default at installation: The vault name is set to VAULT.</p> <p>To change the default: SET DRMVAULTNAME</p> <p>For example, to specify the vault name as IRONVAULT, the contact name as J. SMITH, and the telephone number as 1-555-000-0000, enter: set drmvaultname "Ironvault, J. Smith, 1-555-000-0000"</p>

Specifying recovery instructions for your site

The disaster recovery plan includes instructions that you create.

Enter your instructions in flat files that have the following names:

- *prefix*.RECOVERY.INSTRUCTIONS.GENERAL
- *prefix*.RECOVERY.INSTRUCTIONS.OFFSITE
- *prefix*.RECOVERY.INSTRUCTIONS.INSTALL
- *prefix*.RECOVERY.INSTRUCTIONS.DATABASE
- *prefix*.RECOVERY.INSTRUCTIONS.STGPOOL

Note: The files created for the recovery instructions must be physical sequential files.

RECOVERY.INSTRUCTIONS.GENERAL

Include information such as administrator names, telephone numbers, and location of passwords. For example:

```
Recovery Instructions for Tivoli Storage Manager Server ACMESRV on system ZEUS
Joe Smith (wk 002-000-1111 hm 002-003-0000): primary system programmer
Sally Doe (wk 002-000-1112 hm 002-005-0000): primary recovery administrator
Jane Smith (wk 002-000-1113 hm 002-004-0000): responsible manager
```

Security Considerations:

Joe Smith has the password for the Admin ID ACMEADM. If Joe is unavailable, you need to either issue SET AUTHENTICATION OFF or define a new administrative user ID at the replacement Tivoli Storage Manager server console.

RECOVERY.INSTRUCTIONS.OFFSITE

Include information such as the offsite vault location, courier name, and telephone numbers. For example:

```
Our offsite vault location is Ironvault, Safetown, AZ.
The phone number is 1-800-000-0008. You need to contact them directly
to authorize release of the tapes to the courier.
The name of the courier is Fred Harvey. You can contact him at 1-800-444-0000.
Since our vault is so far away, be sure to give the courier a list
of both the database backup, copy storage pool volumes, and active-data
storage pool volumes required. Fred is committed to returning these volumes
to us in less than 12 hours.
```

RECOVERY.INSTRUCTIONS.INSTALL

Include the following installation information:

Server installation and the location of installation volumes.

For example:

```
You will need to reinstall the Tivoli Storage Manager server and administrative
client after installing the Sun Solaris operating system.
The install volume for the Tivoli Storage Manager server is INS001. If that is
lost, you will need to contact Copy4You Software, at 1-800-000-0000, and obtain
a new copy. Another possibility is the local IBM Branch office at 555-7777.
```

RECOVERY.INSTRUCTIONS.DATABASE

Include information about how to recover the database and about how much hardware space requirements. For example:

```
You will need to find replacement disk space for the server database. We
have an agreement with Joe Replace that in the event of a disaster, he
will provide us with disk space.
```

RECOVERY.INSTRUCTIONS.STGPOOL

Include information on primary storage pool recovery instructions. For example:

```
Do not worry about the archive storage pools during this disaster recovery.  
Focus on migration and backup storage pools.  
The most important storage pool is XYZZZZ.
```

Specifying information about your server and client node machines

You need information about your server machine to rebuild its replacement. You also need information about client node machines to rebuild or restore them.

Follow this procedure to specify that information and store it in the server database:

1. Specify server machine information by issuing the DEFINE MACHINE command with ADSMSERVER=YES. For example, to define machine MACH22 in building 021, 2nd floor, in room 2929, with a priority of 1, enter the following command:

```
define machine tsm1 admsserver=yes priority=1
```

2. Specify the client node location and business priority by issuing the DEFINE MACHINE command. For example, to define machine MACH22 in building 021, 2nd floor, in room 2929, with a priority of 1, enter:

```
define machine mach22 building=021 floor=2 room=2929 priority=1
```

3. Associate one or more client nodes with a machine by issuing the DEFINE MACHNODEASSOCIATION command. Use this association information to identify client nodes on machines that were destroyed. You should restore the file spaces associated with these nodes. For example, to associate node CAMPBELL with machine MACH22, enter:

```
define machnodeassociation mach22 campbell
```

4. To query machine definitions, issue the QUERY MACHINE command. See the example, in “Client recovery scenario” on page 826.
5. To add machine characteristics and recovery instructions to the database, issue the INSERT MACHINE command. You must first query the operating system to identify the characteristics for your client machine.

You can add the information manually or use an awk script. A sample program is shipped with DRM.

- **Add information manually:**

The following partial output is from a query on an AIX client machine.

```
--1 Host Name: mach22 with 256 MB Memory Card  
--- 256 MB Memory Card  
---  
--4 Operating System: AIX Version 4 Release 3  
---  
--- Hardware Address: 10:00:5x:a8:6a:46
```

Specify characteristics and recovery instructions one line at a time with separate INSERT MACHINE commands:

- To save the first line (Host Name: mach22 with 256 MB Memory Card) as line 1 and to save the fourth line (Operating System: AIX Version 4 Release 3) as line 2 for machine MACH22, issue the following commands:

```
insert machine mach22 1 characteristics="Host Name: mach22 with
256 MB Memory Card"
```

```
insert machine mach22 2 characteristics="Operating System:
AIX Version 4 Release 3"
```

- To specify recovery instructions for your client machine, issue the following command:

```
insert machine mach22 1 -
recoveryinstructions="Recover this machine for accounts
receivable dept."
```

- **Add Information Using an Awk Script**

To help automate the adding of client machine information, a sample awk script named *machchar.awk.smp* is shipped with DRM. The following example shows how to use a local program to add machine characteristics or recovery instructions:

- a. The output from the AIX commands *lsdev*, *lsvg*, and *df* is written to the file *clientinfo.txt* on the AIX client machine that backed up data to the server. These commands list the devices, logical volumes by volume group, and file systems.
- b. The file, *clientinfo.txt*, is processed by the awk script, which builds a macro of INSERT MACHINE commands (one command for each line in the file).
- c. Run the macro to load the data into the database.
- d. From an AIX prompt, issue the following commands:

```
echo "devices" > clientinfo.txt
lsdev -C | sort -d -f >> clientinfo.txt
echo "logical volumes by volume group" >> clientinfo.txt
lsvg -o | lsvg -i -l >> clientinfo.txt
echo "file systems" >> clientinfo.txt
df >> clientinfo.txt
```

The following figure is an example procedure named *machchar* to add machine characteristics. The *machchar.awk.smp* script is shipped with DRM and is located in the */opt/tivoli/tsm/server/bin* directory.

```
# Read machine characteristics from a file and build Tivoli Storage
# Manager macro commands to insert the information into the machine
#characteristics table.
# Invoke with:
# awk -f machchar.awk -v machine=acctrcv filewithinfo
BEGIN {
    print "delete machine "machine" type=characteri"
    {
    print "insert machine "machine" "NR" characteri=\""$0"\"
    }
END
{
}
```

- e. The *machchar.awk* script is then run from an AIX prompt as follows:
`nawk -f machchar.awk -v machine=acctrcv clientinfo.txt > clientinfo.mac`
- f. To add the machine characteristics, start an administrative client and run the macro. For example:
`> dsmdmc -id=xxx -pw=xxx -se=xxx macro clientinfo.mac`

You can view your machine characteristics by issuing the QUERY MACHINE command with FORMAT=CHARACTERISTICS parameter.

- g. To specify recovery instructions for your client machine, use this same awk script process but with the RECOVERYINSTRUCTIONS parameter.

Specifying recovery media for client machines

Follow these steps to specify the bootable media needed to reinitialize or reinstall an operating system on a client machine and to associate machines with media. You can also associate non-executable media such as application user guides with client machines.

1. Define the bootable media. For example, define the media named TELLERWRKSTNIMAGE which is for AIX Version 4.3, contains the required volumes named AIX001, AIX002, and AIX003, and is located in Building 21.

```
define recoverymedia tellerwrkstnimage type=boot
  volumenames=aix001,aix002,aix003 product="AIX 4.3"
  location="Building 21"
```

You should define the recovery media after a client machine configuration changes. For example, after you have installed a new level of AIX on a client machine and created a bootable image using **mksysb**, issue the DEFINE RECOVERYMEDIA command to define the new **mksysb** volumes.

To query your recovery media definitions, issue the QUERY RECOVERYMEDIA command with the FORMAT=DETAILED parameter.

2. Associate one or more machines with recovery media. Use the association information to identify the boot media to use in the replacement machines. For example, to associate machine MACH255 with recovery media TELLERWRKSTNIMAGE, issue the following command:

```
define recmedmachassociation tellerwrkstnimage mach255
```

3. When the boot media is moved offsite, update its location. For example, to update the location of boot media TELLERWRKSTNIMAGE to the offsite location IRONVAULT, issue the following command:

```
update recoverymedia tellerwrkstnimage location=ironvault
```

You can define media that contain softcopy manuals that you would need during recovery. For example, to define a CD-ROM containing the AIX 5.1 manuals that are on volume CD0001, enter:

```
define recoverymedia aix51manuals type=other volumes=cd0001
  description="AIX 5.1 Bookshelf"
```

Creating and storing the disaster recovery plan

You can create a disaster recovery plan file and store the file locally or on another server.

The recovery plan contains the following information:

- The recovery procedure
- A list of required database volumes, copy storage pool volumes, and active-data pool volumes, devices to read those volumes, and database and recovery log space requirements
- Copies of the server options file, device configuration file, and volume history information file
- Commands for performing database recovery and primary storage pool recovery
- Commands for registering licenses

- Instructions that you define
- Machine and recovery media information that you define

For details about the recovery plan file, see “The disaster recovery plan file” on page 836.

DRM creates one copy of the disaster recovery plan file each time you issue the PREPARE command. You should create multiple copies of the plan for safekeeping. For example, keep copies in print, on CD, on disk space that is located offsite, or on a remote server.

Before creating a disaster recovery plan, back up your storage pools then backup the database. See “Backing up storage pools” on page 760 and “Backing up the server database” on page 767 for details about these procedures.

If you manually send backup media offsite, see “Moving copy storage pool and active-data pool volumes off-site” on page 816. If you use virtual volumes, see “Using virtual volumes to store data on another server” on page 712.

When your backups are both offsite and marked offsite, you can create a disaster recovery plan.

You can use the Tivoli Storage Manager scheduler to periodically run the PREPARE command (see Chapter 20, “Automating server operations,” on page 571).

Note: DRM creates a plan that assumes that the latest database full plus incremental series would be used to restore the database. However, you may want to use DBSNAPSHOT backups for disaster recovery and retain your full plus incremental backup series on site to recover from possible availability problems. In this case, you must specify the use of DBSNAPSHOT backups in the PREPARE command. For example:

```
prepare source=dbsnapshot
```

Storing the disaster recovery plan locally

When you create a recovery plan file but do not specify a device class, the file is stored locally in a file system. If you store the file locally, you can specify a storage location.

For example, to store the recovery plan file locally in the `/u/server/recoveryplans/` directory, enter:

```
prepare planprefix=/u/server/recoveryplans/
```

Recovery plan files that are stored locally are not automatically expired. You should periodically delete down-level recovery plan files manually. DRM appends to the file name the date and time (yyymmdd.hhmmss). For example:

```
/u/server/recoveryplans/20000925.120532
```

Storing the disaster recovery plan on a target server

When you create a recovery plan file and specify a device class, the file is stored on a target server.

Storing recovery plan files on a target server provides the following:

- A central repository on a target server for recovery plan files
- Automatic expiration of plan files
- Query capabilities that display information about recovery plan files and the ability to display the contents of a recovery plan file located on a target server
- Recovery plan file retrieval from a target server

Set up the source and target servers and define a device class a device type of SERVER (see “Setting up source and target servers for virtual volumes” on page 714 for details). For example, assume a device class named TARGETCLASS is defined on the source server where you create the recovery plan file. Then to create the plan file, enter:

```
prepare devclass=targetclass
```

The recovery plan file is written as an object on the target server, and a volume history record is created on the source server. For more about recovery plan files that are stored on target servers, see “Displaying information about recovery plan files.”

Managing disaster recovery plan files stored on target servers

The following sections describe how to view information about disaster recovery plans stored on a target server and view their contents. It also describes how to direct the contents of a disaster recovery plan file to another file and how to delete volume history records of the recovery plan files.

Displaying information about recovery plan files

You can display information about recovery plan files from the server that created the files (the source server) or from the server on which the files are stored (the target server).

1. **From the source server:** Issue QUERY RPFIL command with the DEVCLASS parameter that was used on the PREPARE command. Specify the type of database backups that were assumed when the plan was created (either full plus incremental or snapshot). For example, to display a list of all recovery plan files that have been saved for the source server on any target servers and created assuming snapshot database backups, enter:

```
query rpfil devclass=* source=dbsnapshot
```

2. You can also issue the QUERY VOLHISTORY command to display a list of recovery plan files for the source server. Specify recovery plan files that were created assuming either full plus incremental database backups (TYPE=RPFIL) or database snapshot backups (TYPE=RPFSNAPSHOT). For example:

```
query volhistory type=rpfil
```

3. **From the target server:** Issue a QUERY RPFIL command that specifies the node name associated with the server or servers that prepared the plan. For example, to display a list of all recovery plan files that have been saved in the target server, enter:

```
query rpfil nodename=*
```

Displaying the contents of a recovery plan file

From the server that created the recovery plan file (the source server) or from the server on which the file is stored (the target server), you can display the contents of that file that was saved as an object on the target server.

For an example of the contents of a recovery plan file, see “The disaster recovery plan file” on page 836. You cannot issue the commands shown below from a server console. An output delay can occur if the plan file is located on tape.

- **From the source server:** Issue the following command for a recovery plan file created on September 1, 2000 at 4:39 a.m. with the device class TARGETCLASS:

```
query rpfcontent marketing.20000901.043900 devclass=targetclass
```
- **From the target server:** Issue the following command for a recovery plan file created on August 31, 2000 at 4:50 a.m. on a source server named MARKETING whose node name is BRANCH8:

```
query rpfcontent marketing.20000831.045000 nodename=branch8
```

Restoring a recovery plan file

To restore a recovery plan file, use the QUERY RPFCONTENT command and direct the output to a file. You can issue the command from the server that created the files (the source server) or from the server on which the files are stored (the target server). To see a list of recovery plan file names, issue the QUERY RPFFILE command.

For example, a recovery plan file named *marketing.20000831.045000* was created using the device class of TARGETCLASS and on a source server whose node name at the target server is BRANCH8. You want to restore the file and direct the output to *rpf.out*:

- **From the source server:** Issue the following command:

```
query rpfcontent marketing.20000831.045000  
devclass=targetclass > rpf.out
```
- **From the target server:** Issue the following command:

```
query rpfcontent marketing.20000831.045000  
nodename=branch8 > rpf.out
```

To display a list of recovery plan files, use the QUERY RPFFILE command. See “Displaying information about recovery plan files” on page 812 for more information.

Expiring recovery plan files automatically

You can set DRM to expire recovery plan files stored on a target server a certain number of days after they are created.

All recovery plan files that meet the criteria are eligible for expiration if both of the following conditions exist:

- The last recovery plan file of the series is over 90 days old.
- The recovery plan file is not associated with the most recent backup series. A backup series consists of a full database backup and all incremental backups that apply to that full backup. Another series begins with the next full backup of the database.

Expiration applies to plan files based on both full plus incremental and snapshot database backups. Note, however, that expiration does not apply to plan files stored locally. See “Storing the disaster recovery plan locally” on page 811.

To set up expiration, issue the SET DRMRPFEXPIREDAYS command. The default value is 60 days. For example, to change the time to 90 days, enter:

```
set drmrpfexpiredays 90
```

Deleting recovery plan files manually

You can delete volume history records containing information about recovery plan file objects.

When the records are deleted from the source server and the grace period is reached, the objects are deleted from the target server. The record for the latest recovery plan file is not deleted.

To delete recovery plan files, issue the DELETE VOLHISTORY command. For example, to delete records for recovery plan files that were created on or before 08/30/2000 and assuming full plus incremental database backup series, enter the following command:

```
delete volhistory type=rpfile todate=08/30/2000
```

To limit the operation to recovery plan files that were created assuming database snapshot backups, specify TYPE=RPFSNAPSHOT.

Moving backup media

To recover from a disaster you need database backup volumes, copy storage pool volumes, and, optionally, active-data pool volumes. To stay prepared for a disaster, you need to perform a number of daily tasks.

Task	Required Privilege Class
Send copy storage pool and active-data pool volumes offsite and back onsite	Unrestricted storage or operator

1. Move new backup media offsite and update the database with their locations. See "Moving copy storage pool and active-data pool volumes off-site" on page 816 for details.
2. Return expired or reclaimed backup media onsite and update the database with their locations. See "Moving copy storage pool and active-data pool volumes on-site" on page 817 for details.
3. Offsite recovery media management does not process virtual volumes. To display all virtual copy storage pool, active-data pool, and database backup volumes that have their backup objects on the remote target server, issue the QUERY DRMEDIA command. For example, enter the following command.

```
query drmedia * wherestate=remote
```

The disaster recovery plan includes the location of copy storage pool volumes and active-data pool volumes. The plan can provide a list of offsite volumes required to restore a server.

The following diagram shows the typical life cycle of the recovery media:

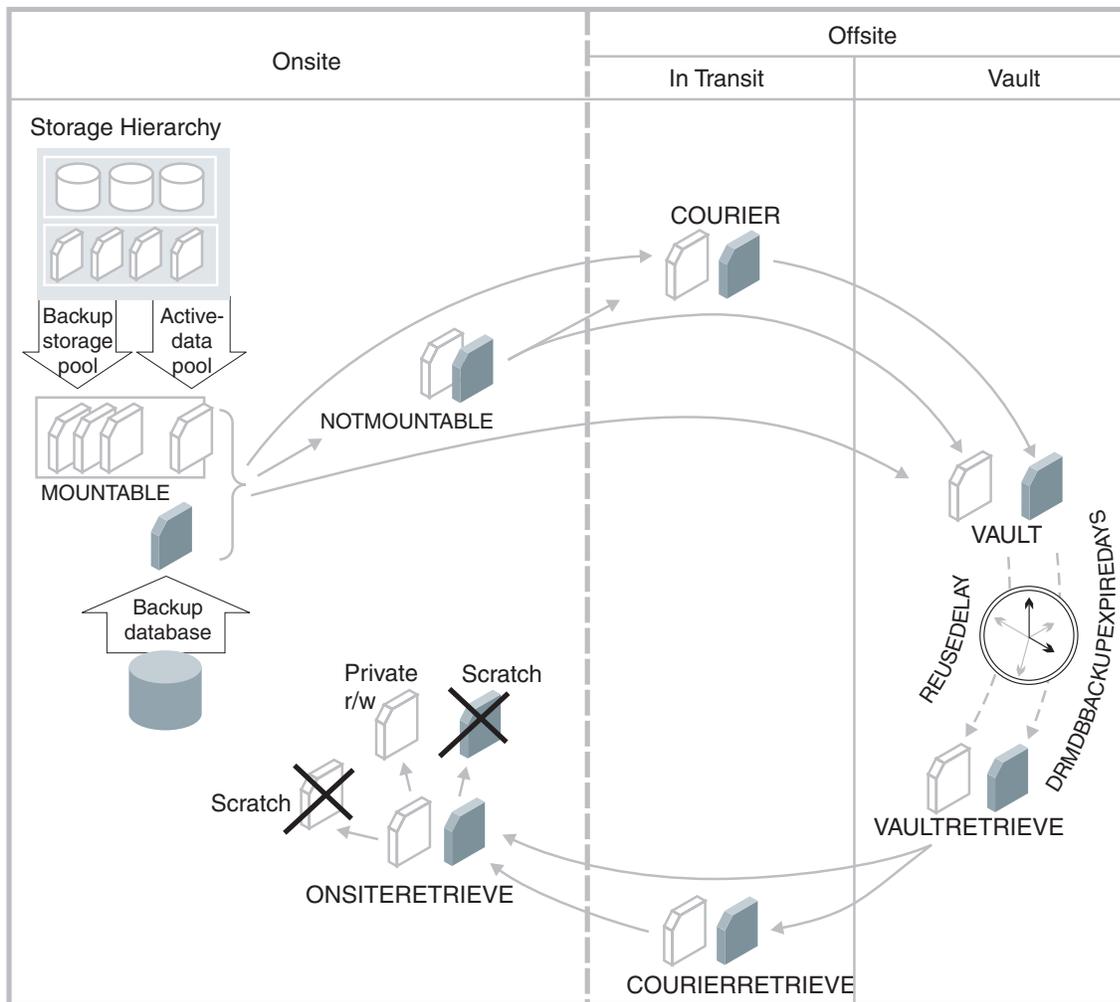


Figure 104. Recovery media life cycle

DRM assigns the following states to volumes. The location of a volume is known at each state.

MOUNTABLE

The volume contains valid data, and Tivoli Storage Manager can access it.

NOTMOUNTABLE

The volume contains valid data and is onsite, but Tivoli Storage Manager cannot access it.

COURIER

The volume contains valid data and is in transit to the vault.

VAULT

The volume contains valid data and is at the vault.

VAULTRETRIEVE

The volume, which is located at the offsite vault, no longer contains valid data and is to be returned to the site. For more information about reclamation of offsite copy storage pool volumes and active-data pool volumes, see “Reclamation of off-site volumes” on page 326. For information on expiration of database backup volumes, see step 1 on page 817.

COURIERRETRIEVE

The volume no longer contains valid data and is in the process of being returned by the courier.

ONSITERETRIEVE

The volume no longer contains valid data and has been moved back to the onsite location. The volume records of database backup, scratch copy storage pool volumes, and scratch active-data pool volumes are deleted from the database. For private copy storage pool volumes and active-data pool volumes, the access mode is updated to READWRITE.

Moving copy storage pool and active-data pool volumes off-site

After you have created the backup copies of your primary storage pools and database, you can send your backup media off-site. To send media off-site, mark the volumes as unavailable to Tivoli Storage Manager and give them to the courier. Do the following actions to identify the database backup, copy storage pool, and active-data pool volumes and move them off-site:

1. Identify the copy storage pool, active-data pool, and database backup volumes to be moved off-site. For example, issue the following command:

```
query drmedia * wherestate=mountable
```

DRM displays information similar to the following:

Volume Name	State	Last Update Date/Time	Automated LibName
TPBK05	Mountable	01/01/2000 12:00:31	LIBRARY
TPBK99	Mountable	01/01/2000 12:00:32	LIBRARY
TPBK06	Mountable	01/01/2000 12:01:03	LIBRARY

2. Indicate the movement of volumes whose current state is MOUNTABLE. For example, issue the following command: by issuing the following command:

```
move drmedia * wherestate=mountable
```

For all volumes in the MOUNTABLE state, DRM does the following:

- Updates the volume state to NOTMOUNTABLE and the volume location according to the SET DRMNOTMOUNTABLENAME. If this command has not been issued, the default location is NOTMOUNTABLE.
- For a copy storage pool volume or active-data pool volume, updates the access mode to unavailable.
- For a volume in an automated library, checks the volume out of the library.
 - a. During checkout processing, SCSI libraries request operator intervention. To bypass these requests and eject the cartridges from the library, first issue the following command:

```
move drmedia * wherestate=mountable remove=no
```

- b. Access a list of the volumes by issuing the following command:

```
query drmedia wherestate=notmountable
```

From this list identify and remove the cartridges (volumes) from the library.

- c. For the 349X library type, if the number of cartridges to be checked out of the library is greater than the number of slots in the I/O station, you can define a high capacity area in your library. Then use the following command to eject the cartridges to the high capacity area, rather than to the I/O station:

```
move drmedia * wherestate=mountable remove=bulk
```

- Send the volumes to the off-site vault. Issue the following command to have DRM select volumes in the NOTMOUNTABLE state:

```
move drmedia * wherestate=notmountable
```

For all volumes in the NOTMOUNTABLE state, DRM updates the volume state to COURIER and the volume location according to the SET DRMCOURIERNAME. If the SET command has not yet been issued, the default location is COURIER. For more information, see “Specifying defaults for offsite recovery media management” on page 805

- When the vault location confirms receipt of the volumes, issue the MOVE DRMEDIA command in the COURIER state. For example:

```
move drmedia * wherestate=courier
```

For all volumes in the COURIER state, DRM updates the volume state to VAULT and the volume location according to the SET DRMVAULTNAME command. If the SET command has not yet been issued, the default location is VAULT. For more information, see “Specifying defaults for offsite recovery media management” on page 805.

- Display a list of volumes that contain valid data at the vault. Issue the following command:

```
query drmedia wherestate=vault
```

DRM displays information similar to the following:

Volume Name	State	Last Update Date/Time	Automated LibName
TAPE0P	Vault	01/05/2000 10:53:20	
TAPE1P	Vault	01/05/2000 10:53:20	
DBT02	Vault	01/05/2000 10:53:20	
TAPE3S	Vault	01/05/2000 10:53:20	

- If you do not want to step through all the states, you can use the TOSTATE parameter on the MOVE DRMEDIA command to specify the destination state. For example, to transition the volumes from NOTMOUNTABLE state to VAULT state, issue the following command:

```
move drmedia * wherestate=notmountable tostate=vault
```

For all volumes in the NOTMOUNTABLE state, DRM updates the volume state to VAULT and the volume location according to the SET DRMVAULTNAME command. If the SET command has not yet been issued, the default location is VAULT.

See “Staying prepared for a disaster” on page 821 for an example that demonstrates sending server backup volumes off-site using MOVE DRMEDIA and QUERY DRMEDIA commands.

Moving copy storage pool and active-data pool volumes on-site

Use the following procedure to expire the non-virtual database backup volumes and return the volumes back on-site for reuse or disposal.

- To specify the number of days before a database backup series is expired, issue the SET DRMDBBACKUPEXPIREDAYS command. The following example sets the number of days to 30.

```
set drmdbbackupexpiredays 30
```

To ensure that the database can be returned to an earlier level and database references to files in the copy storage pool or active-data pool are still valid, specify the same value for the REUSEDELAY parameter in your copy storage pool and active-data pool definitions. If copy storage pools or active-data pools managed by DRM have different REUSEDELAY values, set the DRMDBBACKUPEXPIREDAYS value to the highest REUSEDELAY value.

A database backup volume is considered eligible for expiration if all of the following conditions are true:

- The age of the last volume of the series has exceeded the expiration value. This value is the number of days since the last backup in the series. At installation, the expiration value is 60 days. To override this value, issue the SET DRMDBBACKUPEXPIREDAYS command.
- For volumes that are not virtual volumes, all volumes in the series are in the VAULT state.
- The volume is not part of the most recent database backup series.

Database backup volumes that are virtual volumes are removed during expiration processing. This processing is started manually by issuing the EXPIRE INVENTORY command or automatically through the EXPINTERVAL option setting specified in the server options file.

2. Move a copy storage pool volume or an active-data pool volume on-site for reuse or disposal. A copy storage pool volume or an active-data pool volume can be moved on-site 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 can be moved on-site if the database backup series is EXPIRED according to the rules outlined in step 1 on page 817. To determine which volumes to retrieve, issue the following command:

```
query drmedia * wherestate=vaultretrieve
```

The server dynamically determines which volumes can be moved back on-site. When you issue QUERY DRMEDIA WHERESTATE=VAULTRETRIEVE, the field **Last Update Date/Time** in the output will contain the data and time that the state of the volume was moved to VAULT, not VAULTRETRIEVE. Because the server makes the VAULTRETRIEVE determination dynamically, issue QUERY DRMEDIA WHERESTATE=VAULTRETRIEVE without the BEGINDATE, ENDDATE, BEGINTIME or ENDTIME parameters. Doing so will ensure that you identify all volumes that are in the VAULTRETRIEVE state.

3. After the vault location acknowledges that the volumes have been given to the courier, issue the MOVE DRMEDIA command.

```
move drmedia * wherestate=vaultretrieve
```

The server does the following for all volumes in the VAULTRETRIEVE state:

- Change the volume state to COURIERRETRIEVE.
 - Update the location of the volume according to what is specified in the SET DRMCOURIERNAME command. For more information, see “Specifying defaults for offsite recovery media management” on page 805.
4. When the courier delivers the volumes, acknowledge that the courier has returned the volumes on-site. Issue the following command; by issuing:

```
move drmedia * wherestate=courierretrieve
```

The server does the following for all volumes in the COURIERRETRIEVE state:

- Moves the volumes on-site where they can be reused or disposed of.
- Deletes the database backup volumes from the volume history table.

- For scratch copy storage pool volumes or active-data pool volumes, deletes the record in the database. For private copy storage pool volumes or active-data pool volumes, updates the access to read/write.
5. If you do not want to step through all the states, you can use the TOSTATE parameter on the MOVE DRMEDIA command to specify the destination state. For example, to move the volumes from VAULTRETRIEVE state to ONSITERETRIEVE state, issue the following command:
- ```
move drmedia * wherestate=vaultretrieve toststate=onsiteretrieve
```

The server does the following for all volumes with in the VAULTRETRIEVE state:

- Moves the volumes on-site where they can be can be reused or disposed of.
- Deletes the database backup volumes from the volume history table.
- For scratch copy storage pool volumes or active-data pool volumes, deletes the record in the database. For private copy storage pool volumes or active-data pool volumes, updates the access to read/write.

---

## Summary of disaster recovery manager daily tasks

This section summarizes the use of DRM during routine operations and during disaster recovery.

### Setup

1. License DRM.
2. Ensure the device configuration and volume history files exist.
3. Back up the storage pools.
4. Copy active data to active-data pools.
5. Do a full backup the database (for example, a database snapshot backup).
6. Define site-specific server recovery instructions.
7. Describe priority client machines.
8. Generate the disaster recovery plan.

### Daily Preparation Operations

#### Day 1

1. Back up client files.
2. Back up the primary storage pools to copy storage pools.
3. Copy active data from primary storage pools to active-data pools.
4. Back up the database (for example, a database snapshot backup).
5. Mark the backup volumes as unavailable to Tivoli Storage Manager.
6. Send the backup volumes and disaster recovery plan file to the vault.
7. Generate the disaster recovery plan.

#### Day 2

1. Back up client files
2. Back up active and inactive data that is in the primary storage pools to copy storage pools. Copy the active data that is in primary storage pools to active-data pools.
3. Back up the database (for example, a database snapshot backup).
4. Mark the backup volumes as unavailable to Tivoli Storage Manager.
5. Send the backup volumes and disaster recovery plan file to the vault.

6. Generate the disaster recovery plan.

#### **Day 3**

1. Automatic storage pool reclamation processing occurs.
2. Back up client files.
3. Back up the active and inactive data that is in primary storage pools to copy storage pools. Copy the active data that is in primary storage pools to active-data pools.
4. Back up the database (for example, a database snapshot backup).
5. Send the backup volumes and a list of expired volumes to be reclaimed to the vault.
6. The vault acknowledges receipt of the volumes sent on the previous day.
7. Generate the disaster recovery plan.

### **Disaster and Recovery**

#### **Day 4**

The server and the client machines are destroyed.

1. Restore the server using the latest recovery plan.
2. Identify the top priority client nodes at the disaster site.
3. Restore urgently needed client-system files from active-data pools. Restore other, less urgently needed client-system files from copy storage pools.
4. Restore the primary storage pools from copy storage pools.  
**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.
5. Move database backup, copy storage pool, and active-data pool volumes to the vault.

### **Daily Operations**

#### **Day 5**

1. Back up client files.
2. Back up the active and inactive data that is in primary storage pools to copy storage pools. Copy the active data that is in primary storage pools to active-data pools.
3. Back up the database (for example, a database snapshot backup).
4. Send the backup volumes and a list of expired volumes to be reclaimed to the vault.
5. Generate the disaster recovery plan.

---

## Staying prepared for a disaster

This topic provides an overview and a scenario of the tasks required to stay prepared for a disaster. The steps are performed by the onsite Tivoli Storage Manager administrator unless otherwise indicated.

1. Record the following information in the RECOVERY.INSTRUCTIONS stanza source files:
  - Software license numbers
  - Sources of replacement hardware
  - Any recovery steps specific to your installation
2. Store the following information in the database:
  - Server and client node machine information (DEFINE MACHINE, DEFINE MACHINENODE ASSOCIATION, and INSERT MACHINE)
  - The location of the boot recovery media (DEFINE RECOVERYMEDIA)
3. Schedule automatic nightly backups to occur in the following order:
  - Primary Storage Pools
    - Backup active and inactive data to copy storage pools.
    - Copy active data to active-data pools.
  - Database
4. Daily, create a list of the previous night's database, copy storage pool, and active-data pool volumes to be sent offsite:

```
query drmedia * wherestate=mountable
```

  - a. Check the volumes out of the library:

```
move drmedia * wherestate=mountable
```
  - b. Send the volumes offsite and record that the volumes were given to the courier:

```
move drmedia * wherestate=notmountable
```
5. Create a new recovery plan:

```
prepare
```
6. Give a copy the recovery plan file to the courier.
7. Create a list of tapes that contain data that is no longer valid and that should be returned to the site:

```
query drmedia * wherestate=vaultretrieve
```
8. Give the courier the database backup tapes, storage pool backup tapes, active-data pool tapes, the recovery plan file, and the list of volumes to be returned from the vault.
9. The courier gives you any tapes that were on the previous day's return from the vault list.

Update the state of these tapes and check them into the library:

```
move drmedia * wherestate=courierretrieve cmdf=/drm/checkin.libvol
cmd="checkin libvol libauto &vol status=scratch"
```

The volume records for the tapes that were in the COURIERRETRIEVE state are deleted from the database. The MOVE DRMEDIA command also generates the CHECKIN LIBVOL command for each tape processed in the file */drm/checkin.libvol*. For example:

```
checkin libvol libauto tape01 status=scratch
checkin libvol libauto tape02 status=scratch
...
```

**Note:** An administrator can run the MACRO command by specifying /drm/checkin.libvol.

```
> dsmadm -id=xxxxx -pa=yyyyyy -se=zzzz MACRO /drm/checkin.libvol
```

10. The courier takes the database backup tapes, storage pool backup tapes, active-data pool tapes, the recovery plan, and the list of volumes to return from the vault.
11. Call the vault and verify that the backup tapes arrived and are secure, and that the tapes to be returned to the site have been given to the courier.
12. Set the location of the volumes sent to the vault:  

```
move drmedia * wherestate=courier
```
13. Set the location of the volumes given to the courier by the vault:  

```
move drmedia * wherestate=vaultretrieve
```

---

## Recovering from a disaster

This section provides an overview of the tasks involved in recovering the server and clients. It also presents scenarios of both procedures.

**Recovering the Server:** Here are guidelines for recovering your server:

1. Obtain the latest disaster recovery plan file.
2. Break out the file to view, update, print, or run as macros or scripts (for example, batch programs or batch files).
3. Obtain the copy storage pool volumes and active-data pool volumes from the vault.
4. Locate a suitable replacement machine.
5. Restore the operating system and Tivoli Storage Manager to your replacement machine.
6. Review the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE RECOVERY.SCRIPT.NORMAL.MODE scripts because they are important for restoring the server to a point where clients can be recovered (see “Disaster recovery mode stanza” on page 844).

**Recovering the Clients:** To recover clients, do the following:

1. Get the following information by querying the recovered database:
  - Client machines that have been defined to Tivoli Storage Manager, along with their location and restore priority value
  - The location of the boot recovery media
  - Specific recovery instructions for the machine
  - Hardware requirements for the machine
2. With this information restore the client machines. As a first priority, check into the library the volumes that belong to active-data pools. When a client initiates a restore attempt, storage volumes within an active-data pool have a higher restore priority than standard primarysequential storage pool volumes or copy storage pool volumes.

## Server recovery scenario

Here is the procedure for a complete recovery of the server after a disaster has destroyed it. In this example virtual volumes are not used. The steps are performed by the onsite administrator unless otherwise indicated.

1. Review the recovery steps described in the RECOVERY.INSTRUCTIONS.GENERAL stanza of the plan.
2. Request the server backup tapes from the offsite vault.
3. Break out the recovery plan file stanzas into multiple files (see “Breaking out a disaster recovery plan file” on page 836.) These files can be viewed, updated, printed, or run as Tivoli Storage Manager macros or scripts.
4. Print the RECOVERY.VOLUMES.REQUIRED file. Give the printout to the courier to retrieve the copy storage pool volumes and active-data pool volumes.
5. Find a replacement server. The RECOVERY.DEVICES.REQUIRED stanza specifies the device type that is needed to read the backups. The SERVER.REQUIREMENTS stanza specifies the disk space required.
6. The recovery media names and their locations are specified in the RECOVERY.INSTRUCTIONS.INSTALL stanza and the MACHINE.RECOVERY.MEDIA.REQUIRED stanza. Ensure that the environment is the same as when the disaster recovery plan file was created. The environment includes:
  - The directory structure of the Tivoli Storage Manager server executable and disk formatting utility
  - The directory structure for Tivoli Storage Manager server configuration files (disk log, volume history file, device configuration file, and server options file)
  - The directory structure and the files created when the disaster recovery plan file was split into multiple files
7. Restore the operating system and the Tivoli Storage Manager server software to the replacement server in one of the following ways:
  - Use a system restore utility that is appropriate for your platform.
  - Build a new replacement server instead of restoring the environment from a backup:
    - a. Install the Tivoli Storage Manager server software
    - b. Create the database instance user ID and group as in the original.
    - c. Create the database directories, the active directories, and the archive directories as in the original.
    - d. Run the dsmicfgx utility to configure the replacement instance. This step configures the API for the DSMSERV RESTORE DB utility.
      - 1) Specify the instance userid and password.
      - 2) Specify the database directories, the active directories, and the archive directories.
    - e. Remove the database instance that was created by the dsmicfgx utility. For example

```
dsmserve removedb TSMDB1
```
    - f. Restore the original dsmserve.opt, volume history, and device configuration files to the instance directory (as done by the recovery script in plan file)
    - g. Run the DSMSERV RESTORE DB (as done by the recovery script in the plan file.

8. Review the Tivoli Storage Manager macros contained in the recovery plan:
  - If, at the time of the disaster, the courier had not picked up the previous night's database and storage pool incremental backup volumes but they were not destroyed, remove the entry for the storage pool backup volumes from the `COPYSTGPOOL.VOLUMES.DESTROYED` file.
  - If, at the time of the disaster, the courier had not picked up the previous night's database and active-data pool volumes but they were not destroyed, remove the entry for the active-data pool volumes from the `ACTIVEDATASTGPOOL.VOLUMES.DESTROYED` file.
9. If some required storage pool backup volumes could not be retrieved from the vault, remove the volume entries from the `COPYSTGPOOL.VOLUMES.AVAILABLE` file.  
 If some required active-data pool volumes could not be retrieved from the vault, remove the volume entries from the `ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE` file.
10. If all primary volumes were destroyed, no changes are required to the `PRIMARY.VOLUMES` script and Tivoli Storage Manager macro files.
11. Review the device configuration file to ensure that the hardware configuration at the recovery site is the same as the original site. Any differences must be updated in the device configuration file. Examples of configuration changes that require updates to the configuration information are:
  - Different device names
  - Use of a manual library instead of an automated library
  - For automated libraries, the requirement of manually placing the database backup volumes in the automated library and updating the configuration information to identify the element within the library. This allows the server to locate the required database backup volumes.

For information about updating the device configuration file, see "Updating the device configuration file" on page 776.
12. To restore the database to a point where clients can be recovered, ensure that the Tivoli Storage Manager server is halted and then invoke the `RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE` script file. Enter the script file name at the command prompt. As an alternative, you can use the recovery script as a guide and manually issue the steps.  
 The following steps are an example recovery script:
  - a. Copy the Tivoli Storage Manager server options file from the `dsmserv.opt` file to its original location.
  - b. Copy the volume history file required by database restore processing from the `VOLUME.HISTORY.FILE` file to its original location. Use this copy of the volume history file unless you have a more recent copy (after the disaster occurred).
  - c. Copy the device configuration file required by database restore processing from the `DEVICE.CONFIGURATION.FILE` file to its original location.
  - d. Issue the `DSMSERV RESTORE DB` command.
  - e. Start the server.
  - f. Register Tivoli Storage Manager server licenses.
  - g. Mark copy storage pool volumes and active-data pool volumes retrieved from the vault as available.
  - h. Mark copy storage pool volumes and active-data pool volumes that cannot be obtained as unavailable.

- i. Mark primary storage pool volumes as *destroyed*.

Due to changes in hardware configuration during recovery, you might have to update the device configuration file located in the restored Tivoli Storage Manager database (see “Updating the device configuration file” on page 776)

You can mount copy storage pool volumes and active-data pool volumes upon request, check in the volumes in advance, or manually place the volumes in the library and ensure consistency by issuing the AUDIT LIBRARY command. Use the AUDIT LIBRARY command to ensure that the restored Tivoli Storage Manager database is consistent with the automated library volumes.

13. If client workstations are not damaged, invoke the RECOVERY.SCRIPT.NORMAL.MODE script file to restore the server primary storage pools. If client workstations are damaged, you may want to delay this action until after all clients are recovered.

This action is optional because Tivoli Storage Manager can access the copy storage pool volumes and active-data pool volumes directly to restore client data. Using this feature, you can minimize client recovery time because server primary storage pools do not have to be restored first. However, in this scenario, the client workstations were not damaged, so the focus of the administrator is to restore full Tivoli Storage Manager server operation.

As an alternative, you can use the recovery script as a guide and manually run each step. The steps run in this script are:

- a. Create replacement primary volumes.
- b. Define the replacement primary volumes to Tivoli Storage Manager.
- c. Restore the primary storage pools from copy storage pools.

**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.

14. Collect the database backup, copy storage pool volumes, and active-data pool volumes used in the recovery for return to the vault. For these backup volumes to be returned to the vault using the routine MOVE DRMEDIA process, issue the following commands:

```
update volhist TPBK50 devcl=lib8mm ormstate=mountable
update volhist TPBK51 devcl=lib8mm ormstate=mountable
```

The copy storage pool volumes and active-data pool volumes used in the recovery already have the correct ORMSTATE.

15. Issue the BACKUP DB command to back up the newly restored database.
16. Issue the following command to check the volumes out of the library:

```
move drmedia * wherestate=mountable
```

17. Create a list of the volumes to be given to the courier:

```
query drmedia * wherestate=notmountable
```

18. Give the volumes to the courier and issue the following command:

```
move drmedia * wherestate=notmountable
```

19. Issue the PREPARE command.

## Client recovery scenario

The following scenario demonstrates the recovery of clients.

1. To view a list of client machines that were lost in building 21 and their restore priority, issue the following command:

```
query machine building=021 format=detailed
```

DRM displays information similar to the following:

```
Machine Name: POLARIS
Machine Priority: 1
Building: 21
Floor: 2
Room: 1
Server?: No
Description: Payroll
Node Name: POLARIS
Recovery Media Name: MKSYSB1
Characteristics?: Yes
Recovery Instructions?: Yes
```

Identify which client machines have the highest priority so that restores can begin using active-data pool volumes.

2. For *each* machine, issue the following commands:
  - a. Determine the location of the boot media. For example:

```
query recoverymedia mksysb1
```

The server displays the following information:

| Recovery Media Name | Volume Names      | Location  | Machine Name |
|---------------------|-------------------|-----------|--------------|
| MKSYSB1             | vol1 vol2<br>vol3 | IRONVAULT | POLARIS      |

- b. Determine the machine-specific recovery instructions. For example:

```
query machine polaris format=recoveryinstructions
```

The server displays the following:

```
Recovery Instructions for Polaris.
Primary Contact:
Jane Smith (wk 520-000-0000 hm 520-001-0001)
Secondary Contact:
John Adams (wk 520-000-0001 hm 520-002-0002)
```

- c. Determine the machine hardware requirements.

```
query machine polaris format=characteristics
```

The server displays information similar to the following:

```

devices
aio0 Defined Asynchronous I/O
bus0 Available 00-00 Microchannel Bus
fd0 Available 00-00-0D-00 Diskette Drive
fda0 Available 00-00-0D Standard I/O Diskette Adapter
fpa0 Available 00-00 Floating Point Processor
gda0 Available 00-04 Color Graphics Display Adapter
hd1 Defined Logical volume
hd2 Defined Logical volume
hd3 Defined Logical volume
hdisk0 Available 00-01-00-00 400 MB SCSI Disk Drive
hdisk1 Available 00-01-00-40 Other SCSI Disk Drive
hft0 Available High Function Terminal Subsystem
inet0 Available Internet Network Extension
ioplanar0 Available 00-00 I/O Planar
kbd0 Defined 00-00-0K-00 United States keyboard
lb0 Available 00-02-00-20 TIVSM Library
lo0 Available Loopback Network Interface
loglv00 Defined Logical volume
lp0 Available 00-00-0P-00 IBM 4201 Model 3 Proprinter III
lv03 Defined Logical volume
lv04 Defined Logical volume
lvdd Available N/A
mem0 Available 00-0B 8 MB Memory Card
mem1 Available 00-0C 16 MB Memory Card
mous0 Defined 00-00-0M-00 3 button mouse
mt0 Available 00-02-00-40 TIVSM Tape Drive
ppa0 Available 00-00-0P Standard I/O Parallel Port Adapter
pty0 Available Asynchronous Pseudo-Terminal
rootvg Defined Volume group
sa0 Available 00-00-S1 Standard I/O Serial Port 1
sa1 Available 00-00-S2 Standard I/O Serial Port 2
scsi0 Available 00-01 SCSI I/O Controller
scsil Available 00-02 SCSI I/O Controller
sio0 Available 00-00 Standard I/O Planar
siokb0 Available 00-00-0K Keyboard Adapter
sioms0 Available 00-00-0M Mouse Adapter
siotb0 Available 00-00-0T Tablet Adapter
sys0 Available 00-00 System Object
sysplanar0 Available 00-00 CPU Planar
sysunit0 Available 00-00 System Unit
tok0 Available 00-03 Token-Ring High-Performance Adapter
tr0 Available Token Ring Network Interface
tty0 Available 00-00-S1-00 Asynchronous Terminal
tty1 Available 00-00-S2-00 Asynchronous Terminal
usrvice Defined Logical volume
veggie2 Defined Volume group
logical volumes by volume group
veggie2:
LV NAME TYPE LPs PPs PVs LV STATE MOUNT POINT
hd2 jfs 103 103 1 open/syncd /usr
hd1 jfs 1 1 1 open/syncd /home
hd3 jfs 3 3 1 open/syncd /tmp
hd9var jfs 1 1 1 open/syncd /var
file systems
Filesystem Total KB free %used iused %iused Mounted on
/dev/hd4 8192 420 94% 909 44% /
/dev/hd9var 4096 2972 27% 87 8% /var
/dev/hd2 421888 10964 97% 17435 16% /usr
/dev/hd3 12288 11588 5% 49 1% /tmp
/dev/hd1 4096 3896 4% 26 2% /home

```

### 3. With the information obtained, restore each client machine.

Once the high-priority clients have begun restoring their data from active-data pools, the other, lower-priority clients can begin restoring directly from copy storage pools. Restoration from copy storage pools can run concurrently with the restoration from active-data pools. High-priority clients do not attempt to access the copy storage pool volumes because active-data pools have a higher restore priority than copy storage pools.

---

## Recovering with different hardware at the recovery site

You may have to recover your system using hardware that is different from that used when you backed up your database and created disaster recovery plan file. Before restoring the database, update the device configuration file included in the recovery plan file. After restoring the database, update the device configuration on the database.

This section describes a number of such situations in detail. If the hardware environment is different at the recovery site, you must update the device configuration file. Tivoli Storage Manager uses the device configuration file to access the devices that are needed to read the database backup volumes. The RECOVERY.VOLUMES.REQUIRED stanza in the plan file identifies the volumes that are needed to restore the database.

### Automated SCSI library at the original and recovery sites

Manually place the database backup volumes in the automated library and note the element numbers where you place them. Then update the comments in the device configuration file to identify the locations of those volumes.

**Note:** You may also need to audit the library after the database is restored in order to update the server inventory of the library volumes.

Here is an example of an original device configuration file, which describes an automated tape library:

```
/* Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
 mountlimit=2 mountwait=60 mountretention=60
 prefix=tsm library=auto8mmlib

define library auto8mmlib libtype=scsi

define drive auto8mmlib 8mm_tape0 element=82 online=yes

define drive auto8mmlib 8mm_tape1 element=83 online=yes

define path server1 auto8mmlib srctype=server desttype=library
 device=/dev/rmt/41b

define path server1 8mm_tape0 srctype=server desttype=drive
 library=auto8mmlib device=/dev/rmt/1mt

define path server1 8mm_tape1 srctype=server desttype=drive
 library=auto8mmlib device=/dev/rmt/1mt

/* LIBRARYINVENTORY SCSI AUTO8MMLIB KEV004 1 101*/
/* LIBRARYINVENTORY SCSI AUTO8MMLIB KEV005 3 101*/
```

Here is an example of the updated device configuration file when an automated library is used at the recovery site to read a database volume DBBK01:

```
/* Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
 mountlimit=2 mountwait=60 mountretention=60
 prefix=tsm library=auto8mmlib

define library auto8mmlib libtype=scsi
```

```

define drive auto8mmlib 8mm_tape0 element=82 online=yes

define drive auto8mmlib 8mm_tape1 element=83 online=yes

define path server1 auto8mmlib srctype=server desttype=library
device=/dev/rmt/41b

define path server1 8mm_tape0 srctype=server desttype=drive
library=auto8mmlib device=/dev/rmt/1mt

define path server1 8mm_tape1 srctype=server desttype=drive
library=auto8mmlib device=/dev/rmt/1mt

/* LIBRARYINVENTORY SCSI AUTO8MMLIB DBBK01 1 101*/

```

In this example, database backup volume DBBK01 was placed in element 1 of the automated library. Then a comment is added to the device configuration file to identify the location of the volume. Tivoli Storage Manager needs this information to restore the database restore. Comments that no longer apply at the recovery site are removed.

## Automated SCSI library at the original site and a manual scsi library at the recovery site

Ensure that the DEFINE DRIVE and DEFINE LIBRARY commands in the device configuration file are valid for the new hardware configuration.

For example, if an automated tape library was used originally and cannot be used at the recovery site, update the device configuration file. Include the DEFINE LIBRARY and DEFINE DRIVE commands that are needed to define the manual drive to be used. In this case, you must manually mount the backup volumes.

**Note:** If you are using an automated library, you may also need to update the device configuration file to specify the location of the database backup volume.

Here is an example of an original device configuration file, which describes an automated tape library:

```

/* Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
mountlimit=2 mountwait=60 mountretention=60
prefix=tsm library=auto8mmlib

define library auto8mmlib libtype=scsi

define drive auto8mmlib 8mm_tape0 element=82 online=yes

define drive auto8mmlib 8mm_tape1 element=83 online=yes

define path server1 auto8mmlib srctype=server desttype=library
device=/dev/rmt/41b

define path server1 8mm_tape0 srctype=server desttype=drive
library=auto8mmlib device=/dev/rmt/1mt

define path server1 8mm_tape1 srctype=server desttype=drive
library=auto8mmlib device=/dev/rmt/1mt

/* LIBRARYINVENTORY SCSI AUTO8MMLIB KEV004 1 101*/
/* LIBRARYINVENTORY SCSI AUTO8MMLIB KEV005 3 101*/

```

Here is an example of the updated device configuration file when a manual library is used at the recovery site:

```
/* Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
 mountlimit=1 mountwait=60 mountretention=60 prefix=tsm
 library=manual8mm

define library manual8mm libtype=manual

define drive manual8mm 8mm_tape0

define path server1 8mm_tape0 srctype=server desttype=drive
 library=manual8mm device=/dev/rmt/1mt
```

The following changes were made:

- In the device class definition, the library name was changed from AUTO8MMLIB to MANUAL8MM. The device class name remains the same because it is associated with the database backup volumes in the volume history file.
- The manual library, MANUAL8MM, was defined.
- A new drive, 8MM\_TAPE0, was defined for the manual library.
- The comments that named the location of volumes in the automated library were removed.

After you restore the database, modify the device configuration file in the database. After starting the server, define, update, and delete your library and drive definitions to match your new configuration.

**Note:** If you are using an automated library, you may need to use the AUDIT LIBRARY command to update the server inventory of the library volumes.

## Managing copy storage pool volumes and active-data pool volumes at the recovery site

The RECOVERY.VOLUMES.REQUIRED stanza in the recovery plan file identifies the required copy storage pool volumes and active-data pool volumes. All volumes must be available to the restored server.

The restored server uses copy storage pool volumes to satisfy requests (for example, from backup/archive clients) and to restore primary storage pool volumes that were destroyed. If they are available, the server uses active-data pools to restore critical client data.

**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.

After the database is restored, you can handle copy storage pool volumes and active-data pool volumes at the recovery site in three ways:

- Mount each volume as requested by Tivoli Storage Manager. If an automated library is used at the recovery site, check the volumes into the library.
- Check the volumes into an automated library before Tivoli Storage Manager requests them.

- Manually place the volumes in an automated library and audit the library to update the server inventory.

If you are using an automated library, you may also need to audit the library after the database is restored in order to update the Tivoli Storage Manager inventory of the volumes in the library.

## Disaster recovery manager checklist

The following checklist can help you set up disaster recovery manager.

Table 67. Checklist

| Activity                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Start Date | End Date | Status | Responsible Person | Backup Person |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|----------|--------|--------------------|---------------|
| <b>Plan for DRM</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |            |          |        |                    |               |
| <b>Evaluate your disaster recovery requirements</b> <ul style="list-style-type: none"> <li>• What are the business priorities for recovering your clients?</li> <li>• Where is the recovery site?</li> <li>• Is the recovery site hot, warm, or cold?</li> <li>• Do the clients have connectivity to recovery server?</li> <li>• Who are the system and Tivoli Storage Manager administrators?</li> <li>• Will you need to return to the original site?</li> <li>• Where are the offsite backups stored?</li> <li>• How does the vault handle the backup media?</li> <li>• How are the backups packaged or processed?</li> <li>• Who provides the courier service?</li> </ul> |            |          |        |                    |               |
| <b>Evaluate the current storage pool backup implementation</b> <ul style="list-style-type: none"> <li>• What primary storage pools are being backed up?</li> <li>• When are the backups performed?</li> <li>• Will the backups remain onsite or be sent offsite?</li> <li>• Naming conventions for replacement volumes for primary storage pools</li> </ul>                                                                                                                                                                                                                                                                                                                   |            |          |        |                    |               |

Table 67. Checklist (continued)

| Activity                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Start Date | End Date | Status | Responsible Person | Backup Person |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|----------|--------|--------------------|---------------|
| <p><b>Evaluate the current database backup implementation</b></p> <ul style="list-style-type: none"> <li>• When are the backups performed?</li> <li>• Backup purpose: offsite or onsite</li> <li>• Will you use snapshot database backups or full plus incremental database backups?</li> <li>• How long do you want to keep backup series? Verify that the values for copy storage pool and active-data pool REUSEDELAY and DRMDBBACKUPEXPIREDAYS are the same. If copy storage pools or active-data pools managed by DRM have different REUSEDELAY values, set the DRMDBBACKUPEXPIREDAYS value to the highest REUSEDELAY value.</li> </ul>                                           |            |          |        |                    |               |
| <p><b>Determine which primary storage pools are to be managed by DRM</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |            |          |        |                    |               |
| <p><b>Determine which copy storage pools are to be managed by DRM</b></p> <ul style="list-style-type: none"> <li>• Offsite copy storage pools</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |            |          |        |                    |               |
| <p><b>Determine which active-data pools are to be managed by DRM</b></p> <ul style="list-style-type: none"> <li>• Offsite active-data pools</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |            |          |        |                    |               |
| <p><b>Where to Save the Recovery Plan File</b></p> <p><b>Locally:</b></p> <ul style="list-style-type: none"> <li>• What is the recovery plan file pathname prefix?</li> <li>• How will recovery plan files be made available at the recovery site? <ul style="list-style-type: none"> <li>– Print and store offsite</li> <li>– Copy stored offsite</li> <li>– Copy sent/NFS to recovery site</li> </ul> </li> </ul> <p><b>On Another Server:</b></p> <ul style="list-style-type: none"> <li>• What server is to be used as the target server?</li> <li>• What is the name of the target server's device class?</li> <li>• How long do you want to keep recovery plan files?</li> </ul> |            |          |        |                    |               |
| <p><b>Determine where you want to create the user-specified recovery instructions</b></p> <p>What is the prefix of the instructions pathname?</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |          |        |                    |               |

Table 67. Checklist (continued)

| Activity                                                                                                                                                                                                                                                                                  | Start Date | End Date | Status | Responsible Person | Backup Person |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|----------|--------|--------------------|---------------|
| <p><b>Analyze the sequence of steps related to the PREPARE command backup movement</b></p> <p>Document the flow of activities and timings</p> <ul style="list-style-type: none"> <li>• Sending of volumes offsite</li> <li>• Return of empty volumes</li> <li>• PREPARE timing</li> </ul> |            |          |        |                    |               |
| <p><b>Installation</b></p>                                                                                                                                                                                                                                                                |            |          |        |                    |               |
| <p><b>Receive and Install the Tivoli Storage Manager code</b></p>                                                                                                                                                                                                                         |            |          |        |                    |               |
| <p><b>License DRM</b></p> <ul style="list-style-type: none"> <li>• REGISTER LICENSE or</li> <li>• Update the server options</li> </ul>                                                                                                                                                    |            |          |        |                    |               |

Table 67. Checklist (continued)

| Activity                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Start Date | End Date | Status | Responsible Person | Backup Person |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|----------|--------|--------------------|---------------|
| <p><b>Set DRM defaults</b></p> <p>Issue:</p> <ul style="list-style-type: none"> <li>• SET DRMDBBACKUPEXPIREDAYS to define the database backup expiration</li> <li>• SET DRMPRIMSTGPOOL to specify the DRM-managed primary storage pools</li> <li>• SET DRMCOPYSTGPOOL to specify the DRM-managed copy storage pools</li> <li>• SET DRMACTIVEDATASTGPOOL to specify the DRM-managed active-data pools</li> <li>• SET DRMPLANVPOSTFIX to specify a character to be appended to new storage pools</li> <li>• SET DRMPLANPREFIX to specify the RPF prefix</li> <li>• SET DRMINSTRPREFIX to specify the user instruction file prefix</li> <li>• SET DRMNOTMOUNTABLENAME to specify the default location for media to be sent offsite</li> <li>• SET DRMCOURIERNAME to specify the default courier</li> <li>• SET DRMVAULTNAME to specify the default vault</li> <li>• SET DRMCMDFILENAME to specify the default file name to contain the commands specified with the CMD parameter on MOVE and QUERY DRMEDIA</li> <li>• SET DRMCHECKLABEL to specify whether volume labels are verified when checked out by the MOVE DRMEDIA command</li> <li>• SET DRMRPFEXPIREDAYS to specify a value for the frequency of RPF expiration (when plan files are stored on another server)</li> </ul> |            |          |        |                    |               |

Table 67. Checklist (continued)

| Activity                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Start Date | End Date | Status | Responsible Person | Backup Person |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|----------|--------|--------------------|---------------|
| <p><b>Define the site-specific recovery instructions</b></p> <p>Identify:</p> <ul style="list-style-type: none"> <li>• Target disaster recovery server location</li> <li>• Target server software requirements</li> <li>• Target server hardware requirements (storage devices)</li> <li>• Tivoli Storage Manager administrator contact</li> <li>• Courier name and telephone number</li> <li>• Vault location and contact person</li> </ul> <p>Create:</p> <ul style="list-style-type: none"> <li>• Enter the site-specific recovery instructions data into files created in the same path/HLQ as specified by SET DRMINSTRPREFIX</li> </ul> |            |          |        |                    |               |
| <p><b>Test disaster recovery manager</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |          |        |                    |               |
| <p><b>Test the installation and customization</b></p> <ul style="list-style-type: none"> <li>• QUERY DRMSTATUS to display the DRM setup</li> <li>• Back up the active and inactive data that is in primary storage pools to copy storage pools. Copy the active data that is in primary storage pools to active-data pools.</li> <li>• Back up the Tivoli Storage Manager database</li> <li>• QUERY DRMEDIA to list the copy storage pool and active-data pool volumes</li> <li>• MOVE DRMEDIA to move offsite</li> <li>• PREPARE to create the recovery plan file</li> </ul>                                                                 |            |          |        |                    |               |
| <p><b>Examine the recovery plan file created</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |            |          |        |                    |               |
| <p><b>Test the recovery plan file break out</b></p> <ul style="list-style-type: none"> <li>• awk script planexpl.awk</li> <li>• Locally written procedure</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |            |          |        |                    |               |
| <p><b>Set up the schedules for automated functions</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |          |        |                    |               |

---

## The disaster recovery plan file

The disaster recovery plan file contains the information required to recover a Tivoli Storage Manager server to the point in time represented by the last database backup operation that is completed before the plan is created. The plan is organized into stanzas, which you can break out into multiple files.

### Breaking out a disaster recovery plan file

You can break out the stanzas of the disaster recovery plan file into individual files.

You can use an awk script or an editor to break out the stanzas into individual files. A sample procedure, *planexpl.awk.smp*, is shipped with DRM and is located in */opt/tivoli/tsm/server/bin* or wherever the server resides. You can modify this procedure for your installation. Store a copy of the procedure offsite for recovery.

### Structure of the disaster recovery plan file

The disaster recovery plan is divided into the following types of stanzas:

#### Command stanzas

Consist of scripts (for example, batch programs or batch files) and Tivoli Storage Manager macros. You can view, print, and update these stanzas, and run them during recovery.

**Note:** The `RECOVERY.SCRIPT.NORMAL.MODE` and `RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE` stanzas contain the commands that invoke the scripts and macros contained in the other stanzas.

#### Instruction stanzas

Consist of recovery instructions specific to your site. You can view, print, and update these stanzas, and use them during recovery.

#### Server requirements stanzas

Include the database and recovery log requirements, device and volume requirements, and license information. You can view and print these stanzas, and use them during recovery.

#### Configuration file stanzas

Consist of the volume history, device configuration, and server options files.

#### Machine and recovery media stanzas

Consist of machine recovery instructions and information about machine hardware, software, and recovery media. You can print and update these stanzas, and use them during server recovery.

Table 68 on page 837 lists the recovery plan file stanzas, and indicates what type of administrative action is required during set up or periodic updates, routine processing, and disaster recovery. The table also indicates whether the stanza contains a macro, a script, or a configuration file.

**Note:** For tasks identified as **During setup or periodic updates**, DRM automatically collects this information for the plan.

Table 68. Administrative tasks associated with the disaster recovery plan file

| Stanza Name                                    | Tasks                                                                                                          |
|------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| PLANFILE.DESCRPTION                            | None                                                                                                           |
| PLANFILE.TABLE.OF.CONTENTS                     | None                                                                                                           |
| SERVER.REQUIREMENTS                            | None                                                                                                           |
| RECOVERY.INSTRUCTIONS.GENERAL                  | <b>During setup or periodic updates:</b> Edit the source file associated with the stanza (optional)            |
| RECOVERY.INSTRUCTIONS.OFFSITE                  | <b>During setup or periodic updates:</b> Edit the source file associated with the stanza (optional)            |
| RECOVERY.INSTRUCTIONS.INSTALL                  | <b>During setup or periodic updates:</b> Edit the source file associated with the stanza (optional)            |
| RECOVERY.INSTRUCTIONS.DATABASE                 | <b>During setup or periodic updates:</b> Edit the source file associated with the stanza (optional)            |
| RECOVERY.INSTRUCTIONS.STGPOOL                  | <b>During setup or periodic updates:</b> Edit the source file associated with the stanza (optional)            |
| RECOVERY.VOLUMES.REQUIRED                      | <b>During routine processing:</b> MOVE DRMEDIA                                                                 |
| RECOVERY.DEVICES.REQUIRED                      | None                                                                                                           |
| RECOVERY.SCRIPT. DISASTER.RECOVERY.MODE script | <b>During disaster recovery:</b> Edit and run (optional)                                                       |
| RECOVERY.SCRIPT. NORMAL.MODE script            | <b>During disaster recovery:</b> Edit and run (optional)                                                       |
| DB.STORAGEPATHS                                | <b>During disaster recovery:</b> Edit (optional)                                                               |
| LICENSE.REGISTRATION macro                     | <b>During disaster recovery:</b> Edit and run (optional)                                                       |
| ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE            | <b>During routine processing:</b> MOVE DRMEDIA<br><br><b>During disaster recovery:</b> Edit and run (optional) |
| ACTIVEDATASTGPOOL.VOLUMES.DESTROYED            | <b>During routine processing:</b> MOVE DRMEDIA<br><br><b>During disaster recovery:</b> Edit and run (optional) |
| COPYSTGPOOL.VOLUMES.AVAILABLE macro            | <b>During routine processing:</b> MOVE DRMEDIA<br><br><b>During disaster recovery:</b> Edit and run (optional) |
| COPYSTGPOOL.VOLUMES.DESTROYED macro            | <b>During routine processing:</b> MOVE DRMEDIA<br><br><b>During disaster recovery:</b> Edit and run (optional) |
| PRIMARY.VOLUMES.DESTROYED macro                | <b>During disaster recovery:</b> Edit and run (optional)                                                       |

Table 68. Administrative tasks associated with the disaster recovery plan file (continued)

| Stanza Name                                  | Tasks                                                                                                           |
|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| PRIMARY.VOLUMES.REPLACEMENT macro            | <b>During disaster recovery:</b> Edit and run (optional)                                                        |
| STGPOOLS.RESTORE macro                       | <b>During disaster recovery:</b> Edit and run (optional)                                                        |
| VOLUME.HISTORY.FILE configuration file       | <b>During disaster recovery:</b> Copy (optional)                                                                |
| DEVICE.CONFIGURATION.FILE configuration file | <b>During disaster recovery:</b> Edit and copy (optional)                                                       |
| DSMSERV.OPT.FILE configuration file          | <b>During disaster recovery:</b> Edit and copy (optional)                                                       |
| LICENSE.INFORMATION                          | None                                                                                                            |
| MACHINE.GENERAL.INFORMATION                  | <b>During setup or periodic updates:</b> Issue DEFINE MACHINE ADSMSERVER=YES (optional)                         |
| MACHINE.RECOVERY.INSTRUCTIONS                | <b>During setup or periodic updates:</b> Issue INSERT MACHINE RECOVERYINSTRUCTIONS (optional)                   |
| MACHINE.RECOVERY.CHARACTERISTICS             | <b>During setup or periodic updates:</b> Issue INSERT MACHINE CHARACTERISTICS (optional)                        |
| MACHINE.RECOVERY.MEDIA                       | <b>During setup or periodic updates:</b> Issue DEFINE RECOVERYMEDIA and DEFINE RECMEDMACHASSOCIATION (optional) |

## Example disaster recovery plan file

This section contains an example of a disaster recovery plan file and information about each stanza. The disaster recovery plan file has been divided into separate figures that correlate to the descriptions of specific stanzas within each figure.

### Description and table of contents stanzas

These stanzas identify the server for the recovery plan and the date and time the plan is created, and also list all the stanzas in the plan.

#### PLANFILE.DESCRPTION

```
begin PLANFILE.DESCRPTION
Recovery Plan for Server COUPE
Created by DRM PREPARE on 09/26/2008 13:46:24
DRM PLANPREFIX D:\TSM\SERVER1\PLANPRE
Storage Management Server for Windows - Version 6, Release 1, Level 0.0
end PLANFILE.DESCRPTION
```

#### PLANFILE.TABLE.OF.CONTENTS

```

begin PLANFILE.TABLE.OF.CONTENTS

PLANFILE.DESCRPTION
PLANFILE.TABLE.OF.CONTENTS

Server Recovery Stanzas:
SERVER.REQUIREMENTS
RECOVERY.VOLUMES.REQUIRED
RECOVERY.DEVICES.REQUIRED
RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script
RECOVERY.SCRIPT.NORMAL.MODE script
DB.STORAGEPATHS
LICENSE.REGISTRATION macro
ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE macro
ACTIVEDATASTGPOOL.VOLUMES.DESTROYED macro
COPYSTGPOOL.VOLUMES.AVAILABLE macro
COPYSTGPOOL.VOLUMES.DESTROYED macro
PRIMARY.VOLUMES.DESTROYED macro
PRIMARY.VOLUMES.REPLACEMENT macro
STGPOOLS.RESTORE macro
VOLUME.HISTORY.FILE
DEVICE.CONFIGURATION.FILE
DSMSERV.OPT.FILE
LICENSE.INFORMATION

end PLANFILE.TABLE.OF.CONTENTS

```

### Server requirements stanza

The `SERVER.REQUIREMENTS` stanza identifies the database and recovery log storage requirements for the server.

The replacement server must have enough disk space to install the database and recovery log.

This stanza also identifies the directory where the server executable file resided when the server was started. If the server executable file is in a different directory on the replacement server, edit the plan file to account for this change.

If you use links to the server executable file, you must create the links on the replacement machine or modify the following plan file stanzas:

```
RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE
```

```

begin SERVER.REQUIREMENTS

Database Requirements Summary:

 Database Name: tsm_serv
 Total Space(MB): 285,985
 Used Space(MB): 384
 Free Space(MB): 285,527
 Page Size(Bytes): 16,384
 Total Pages: 26,627
 Usable Pages: 26,491
 Used Pages: 22,003
 Free Pages: 4,488
 Full Device Class Name: VTL

Location: E:\tsmdata\DBSpace
Total Space(MB): 285,985
Used Space(MB): 457
Free Space(MB): 285,527

Recovery Log Requirements Summary:

 Assigned Capacity (MB): 8,192
 Used Space(MB): 0
 Free Space(MB): 8,159
 Active Log Directory: H:\tsmdata\Alog
 Mirror Log Directory:
 Archive Failover Log Directory: c:\tsmdata\Flog
 Archive Log Directory: H:\tsmdata\archive

Server Installation Directory: D:\tsm\

end SERVER.REQUIREMENTS

-

begin RECOVERY.VOLUMES.REQUIRED

Volumes required for data base restore

Location = dkvault
Device Class = VTL
Volume Name =
003902L4

Volumes required for storage pool restore

Location = dkvault
Copy Storage Pool = COPYPOOL
Device Class = VTL
Volume Name =
003900L4

Volumes required for active-data storage pool(s)

Location = dkvault
Active-data Storage Pool = ADP1
Device Class = VTL
Volume Name =
003901L4

end RECOVERY.VOLUMES.REQUIRED

```

## Recovery instructions stanzas

The administrator enters recovery instructions into source files that the PREPARE command includes in the plan files.

See “Specifying recovery instructions for your site” on page 807 for details. In the following descriptions, *prefix* represents the prefix portion of the file name. See “Specifying defaults for the disaster recovery plan file” on page 802 for details.

### RECOVERY.INSTRUCTIONS.GENERAL

Identifies site-specific instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.GENERAL. The instructions should include the recovery strategy, key contact names, an overview of key applications backed up by this server, and other relevant recovery instructions.

```
begin RECOVERY.INSTRUCTIONS.GENERAL

This server contains the backup and archive data for FileRight Company
accounts receivable system. It also is used by various end users in the
finance and materials distribution organizations.
The storage administrator in charge of this server is Jane Doe 004-001-0006.
If a disaster is declared, here is the outline of steps that must be completed.
1. Determine the recovery site. Our alternate recovery site vendor is IBM
 BRS in Tampa, FL, USA 213-000-0007.
2. Get the list of required recovery volumes from this recovery plan file
 and contact our offsite vault so that they can start pulling the
 volumes for transfer to the recovery site.
3. etc...

end RECOVERY.INSTRUCTIONS.GENERAL
```

### RECOVERY.INSTRUCTIONS.OFFSITE

Contains instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.OFFSITE. The instructions should include the name and location of the offsite vault, and how to contact the vault (for example, a name and phone number).

```
begin RECOVERY.INSTRUCTIONS.OFFSITE

Our offsite vaulting vendor is OffsiteVault Inc.
Their telephone number is 514-555-2341. Our account rep is Joe Smith.
Our account number is 1239992. Their address is ...
Here is a map to their warehouse ...
Our courier is ...

end RECOVERY.INSTRUCTIONS.OFFSITE
```

### RECOVERY.INSTRUCTIONS.INSTALL

Contains instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.INSTALL. The instructions should include how to rebuild the base server machine and the location of the system image backup copies.

```
begin RECOVERY.INSTRUCTIONS.INSTALL
```

The base server system is Windows Server 2008: Standard running on an IBM PC-350. The Windows Server 2008 operating system and product installation media is stored at the vault. There is also a copy in bldg 24 room 4 cabinet a. The system administrator responsible for the Windows Server 2008 and server installation is Fred Myers. Following are the instructions for installation of Windows Server 2008 and the server:

```
end RECOVERY.INSTRUCTIONS.INSTALL
```

## RECOVERY.INSTRUCTIONS.DATABASE

Contains instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.DATABASE. The instructions should include how to prepare for the database recovery. For example, you may enter instructions on how to initialize or load the backup volumes for an automated library. No sample of this stanza is provided.

## RECOVERY.INSTRUCTIONS.STGPOOL

Contains instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.STGPOOL. The instructions should include the names of your software applications and the copy storage pool names containing the backup of these applications. No sample of this stanza is provided.

## Volume and device requirements stanzas

These stanzas provide a list of volumes required to recover the server and details about the devices needed to read those volumes.

## RECOVERY.VOLUMES.REQUIRED

Provides a list of the database backup, copy storage-pool volumes, and active-data pool volumes required to recover the server. This list can include both virtual volumes and nonvirtual volumes. A database backup volume is included if it is part of the most recent database backup series. A copy storage pool volume or an active-data pool volume is included if it is not empty and not marked *destroyed*.

If you are using a nonvirtual volume environment and issuing the MOVE DRMEDIA command, a blank location field means that the volumes are onsite and available to the server. This volume list can be used in periodic audits of the volume inventory of the courier and vault. You can use the list to collect the required volumes before recovering the server.

For virtual volumes, the location field contains the target server name.

```

begin RECOVERY.VOLUMES.REQUIRED

Volumes required for data base restore

Location = dkvault
Device Class = VTL
Volume Name =
003902L4

Volumes required for storage pool restore

Location = dkvault
Copy Storage Pool = COPYPOOL
Device Class = VTL
Volume Name =
003900L4

Volumes required for active-data storage pool(s)

Location = dkvault
Active-data Storage Pool = ADP1
Device Class = VTL
Volume Name =
003901L4

end RECOVERY.VOLUMES.REQUIRED

```

## RECOVERY.DEVICES.REQUIRED

Provides details about the devices needed to read the backup volumes.

```

begin RECOVERY.DEVICES.REQUIRED

Purpose: Description of the devices required to read the
 volumes listed in the recovery volumes required stanza.

 Device Class Name: VTL
 Device Access Strategy: Sequential
 Storage Pool Count: 2
 Device Type: LTO
 Format: DRIVE
 Est/Max Capacity (MB):
 Mount Limit: 2
 Mount Wait (min): 5
 Mount Retention (min): 1
 Label Prefix: ADSM
 Drive Letter:
 Library: VTL
 Directory:
 Server Name:
 Retry Period:
 Retry Interval:
 Twosided:
 Shared:
 WORM: No
 Drive Encryption: Allow
 Scaled Capacity:
 Last Update by (administrator): ADMIN
 Last Update Date/Time: 09/26/2008 12:11:50

end RECOVERY.DEVICES.REQUIRED

```

## Disaster recovery mode stanza

The RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE stanza contains a script with the commands needed to recover the server.

### RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE

You can use the script as a guide and run the commands from a command line. Or you can copy it to a file, modify it and the files it refers to, and run the script. You may need to modify the script because of differences between the original and the replacement systems. At the completion of these steps, client requests for file restores are satisfied directly from copy storage pool volumes and active-data pool volumes.

The disaster recovery plan issues commands using the administrative client. The disaster recovery plan file issues commands using the administrative client. Ensure that the path to the administrative client is established before running the script. For example, set the shell variable PATH or update the scripts with the path specification for the administrative client.

The commands in the script do the following:

- Restore the server options, volume history, and device configuration information files.
- Invoke the macros contained in the following stanzas:
  - LICENSE.REGISTRATION
  - COPYSTGPOOL.VOLUMES.AVAILABLE
  - COPYSTGPOOL.VOLUMES.DESTROYED
  - ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE
  - ACTIVEDATASTGPOOL.VOLUMES.DESTROYED
  - PRIMARY.VOLUMES.DESTROYED.

To help understand the operations being performed in this script, see “Backup and recovery scenarios” on page 792.

To invoke this script, specify the following positional parameters:

- \$1 (the administrator ID)
- \$2 (the administrator password)
- \$3 (the server ID as specified in the dsm.sys file)

**Note:** The default location for dsm.sys is `/opt/tivoli/tsm/client/admin/bin`.

For example, to invoke this script using an administrator ID of *don*, password of *mox*, server name of *prodtsm*, enter the following command:

```
planprefix/RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE don mox prodtsm
```

For more information, see the entry for the recovery plan prefix in Table 65 on page 802.

```

begin RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script
@echo off

rem Purpose: This script contains the steps required to recover the server
rem to the point where client restore requests can be satisfied
rem directly from available copy storage pool volumes.
rem Note: This script assumes that all volumes necessary for the restore have
rem been retrieved from the vault and are available. This script assumes
rem the recovery environment is compatible (essentially the same) as the
rem original. Any deviations require modification to this script and the
rem macros and scripts it runs. Alternatively, you can use this script
rem as a guide, and manually execute each step.

if not %1.==. if not %2.==. goto start
echo Specify the following positional parameters:
echo administrative client ID and password.
echo Script stopped.
goto end
:start

rem Set the server working directory.
pushd "D:\tsm\server1\"

rem Restore server options, volume history, device configuration files.
copy "D:\TSM\SERVER1\PLANPRE.DSMSERV.OPT.FILE" "D:\TSM\SERVER1\DSMSERV.OPT"
copy "D:\TSM\SERVER1\PLANPRE.VOLUME.HISTORY.FILE" "D:\TSM\SERVER1\VOLHIST.OUT"
copy "D:\TSM\SERVER1\PLANPRE.DEVICE.CONFIGURATION.FILE" "D:\TSM\SERVER1\DEVCFG.OUT"

rem Make sure db storage paths exist.
mkdir "E:\tsmdata\DBSpace"

rem Restore the server database to latest version backed up per the
rem volume history file.
"D:\TSM\SERVER1\DSMSERV" -k "Server1" restore db todate=09/26/2008 totime=13:28:52 +
source=dbb

rem "D:\TSM\SERVER1\DSMSERV" -k "Server1" restore db todate=09/26/2008 totime=13:28:52 +
source=dbb on="D:\TSM\SERVER1\PLANPRE.DB.STORAGEPATHS" activelogdir="H:\tsmdata\Alog"

rem Start the server.
start "Server1" "D:\TSM\SERVER1\DSMSERV" -k "Server1"
echo Wait for the server to start. Ensure that the Administrative command
echo line client option file is set up to communicate with this server, then
echo press enter to continue recovery script execution.
pause

rem Set the administrative command line client directory.
pushd "D:\tsm\Server\tsmdiag"
set DSM_DIR=D:\tsm\Server\tsmdiag

rem Register the Server Licenses.
dsmadm -id=%1 -pass=%2 -ITEMCOMMIT +
-OUTFILE="D:\TSM\SERVER1\PLANPRE.LICENSE.REGISTRATION.LOG" macro +
"D:\TSM\SERVER1\PLANPRE.LICENSE.REGISTRATION.MAC"

```

```

rem Tell the server these active-data pool volumes are available for use.
rem Recovery Administrator: Remove from macro any volumes not obtained from vault.
dsmadm -id=%1 -pass=%2 -ITEMCOMMIT +
-OUTFILE="D:\TSM\SERVER1\PLANPRE.ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE.LOG" +
macro "D:\TSM\SERVER1\PLANPRE.ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE.MAC"

rem Active-data pool volumes in this macro were not marked as 'offsite' at the time
rem PREPARE ran. They were likely destroyed in the disaster.
rem Recovery Administrator: Remove from macro any volumes not destroyed.
dsmadm -id=%1 -pass=%2 -ITEMCOMMIT +
-OUTFILE="D:\TSM\SERVER1\PLANPRE.ACTIVEDATASTGPOOL.VOLUMES.DESTROYED.LOG" +
macro "D:\TSM\SERVER1\PLANPRE.ACTIVEDATASTGPOOL.VOLUMES.DESTROYED.MAC"

rem Tell the server these copy storage pool volumes are available for use.
rem Recovery Administrator: Remove from macro any volumes not obtained from vault.
dsmadm -id=%1 -pass=%2 -ITEMCOMMIT +
-OUTFILE="D:\TSM\SERVER1\PLANPRE.COPYSTGPOOL.VOLUMES.AVAILABLE.LOG" +
macro "D:\TSM\SERVER1\PLANPRE.COPYSTGPOOL.VOLUMES.AVAILABLE.MAC"

rem Copy storage pool volumes in this macro were not marked as 'offsite' at the time
rem PREPARE ran. They were likely destroyed in the disaster.
rem Recovery Administrator: Remove from macro any volumes not destroyed.
dsmadm -id=%1 -pass=%2 -ITEMCOMMIT +
-OUTFILE="D:\TSM\SERVER1\PLANPRE.COPYSTGPOOL.VOLUMES.DESTROYED.LOG" +
macro "D:\TSM\SERVER1\PLANPRE.COPYSTGPOOL.VOLUMES.DESTROYED.MAC"

rem Mark primary storage pool volumes as ACCESS=DESTROYED.
rem Recovery administrator: Remove from macro any volumes not destroyed.
dsmadm -id=%1 -pass=%2 -ITEMCOMMIT +
-OUTFILE="D:\TSM\SERVER1\PLANPRE.PRIMARY.VOLUMES.DESTROYED.LOG" +
macro "D:\TSM\SERVER1\PLANPRE.PRIMARY.VOLUMES.DESTROYED.MAC"

rem Restore the previous working directory.
popd

rem Restore the previous working directory.
popd

:end
end RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script

```

## Normal mode stanza

The RECOVERY.SCRIPT.NORMAL.MODE stanza contains a script with the commands needed to restore the server primary storage pools.

### RECOVERY.SCRIPT.NORMAL.MODE

You can use the script as a guide and run the commands from a command line. Or you can copy it to a file, modify it and the files it refers to, and run the script. You may need to modify the script because of differences between the original and the replacement systems.

The disaster recovery plan issues commands using the administrative client.

**Note:** Ensure that the path to the administrative client is established before running the script. For example, set the shell variable PATH or update the scripts with the path specification for the administrative client.

At the completion of these steps, client requests for file restores are satisfied from primary storage pool volumes. Clients should also be able to resume file backup, archive, and migration functions.

This script invokes the macros contained in the following stanzas:

PRIMARY.VOLUMES.REPLACEMENT

## STGPOOLS.RESTORE

To help understand the operations being performed in this script, see “Backup and recovery scenarios” on page 792.

To invoke this script, the following positional parameters must be specified:

- \$1 (the administrator ID)
- \$2 (the administrator password)
- \$3 (the server ID as specified in the dsm.sys file)

For example, to invoke this script using an administrator ID of *don*, password of *mox*, server name of *prodtsm*, enter the following command:

```
planprefix/RECOVERY.SCRIPT.NORMAL.MODE don mox prodtsm
```

For more information, see the entry for the recovery plan prefix in Table 65 on page 802.

```
begin RECOVERY.SCRIPT.NORMAL.MODE script
@echo off

rem Purpose: This script contains the steps required to recover the server
rem primary storage pools. This mode allows you to return the
rem copy storage pool volumes to the vault and to run the
rem server as normal.
rem Note: This script assumes that all volumes necessary for the restore
rem have been retrieved from the vault and are available. This script
rem assumes the recovery environment is compatible (essentially the
rem same) as the original. Any deviations require modification to this
rem this script and the macros and scripts it runs. Alternatively, you
rem can use this script as a guide, and manually execute each step.

if not %1.==. if not %2.==. goto start
echo Specify the following positional parameters:
echo administrative client ID and password.
echo Script stopped.
goto end
:start

rem Set the administrative command line client directory.
pushd "D:\tsm\Server\tsmdiag"
set DSM_DIR=D:\tsm\Server\tsmdiag

rem Define replacement volumes in the primary storage pools. Must
rem have different name than original.
rem Recovery administrator: Edit macro for your replacement volumes.
dsmadmc -id=%1 -pass=%2 -ITEMCOMMIT +
-OUTFILE="D:\TSM\SERVER1\PLANPRE.PRIMARY.VOLUMES.REPLACEMENT.LOG" +
macro "D:\TSM\SERVER1\PLANPRE.PRIMARY.VOLUMES.REPLACEMENT.MAC"

rem Restore the primary storage pools from the copy storage pools.
dsmadmc -id=%1 -pass=%2 -ITEMCOMMIT +
-OUTFILE="D:\TSM\SERVER1\PLANPRE.STGPOOLS.RESTORE.LOG" +
macro "D:\TSM\SERVER1\PLANPRE.STGPOOLS.RESTORE.MAC"

rem Restore the previous working directory.
popd

:end
end RECOVERY.SCRIPT.NORMAL.MODE script
```

## Database directories stanza

The DB.STORAGEPATHS stanza identifies the directories for the Tivoli Storage Manager database.

This stanza is referred to by the alternate DSMSERV RESTORE DB command in the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script. It is filled out with the "locations" from QUERY DBSPACE . If you need to restore the database to an alternate location, update this file with the new directories. You must also update the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script to invoke the alternate DSMSERV RESTORE DB command instead of the default one.

```
begin DB.STORAGEPATHS
E:\tsmdata\DBSpace
end DB.STORAGEPATHS
```

## License registration stanza

The LICENSE.REGISTRATION stanza contains a macro to register your server licenses.

### LICENSE.REGISTRATION

This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

```
begin LICENSE.REGISTRATION macro

/* Purpose: Register the server licenses by specifying the names */
/* of the enrollment certificate files necessary to re-create the */
/* licenses that existed in the server. */
/* Recovery Administrator: Review licenses and add or delete licenses */
/* as necessary. */

register license file(dataret.lic)
register license file(tsmbasic.lic)
register license file(tsmee.lic)

end LICENSE.REGISTRATION macro
```

## Copy storage pool volumes stanzas

The copy storage pool volumes stanzas contain macros to mark copy storage pool volumes as available or unavailable.

### COPYSTGPOOL.VOLUMES.AVAILABLE

Contains a macro to mark copy storage pool volumes that were moved offsite and then moved back onsite. This stanza does not include copy storage pool virtual volumes. You can use the information as a guide and issue the administrative commands, or you can copy it to a file, modify it, and run it. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

After a disaster, compare the copy storage pool volumes listed in this stanza with the volumes that were moved back onsite. You should remove entries from this stanza for any missing volumes.

```

begin COPYSTGPOOL.VOLUMES.AVAILABLE macro

/* Purpose: Mark copy storage pool volumes as available for use in recovery. */
/* Recovery Administrator: Remove any volumes that have not been obtained */
/* from the vault or are not available for any reason. */
/* Note: It is possible to use the mass update capability of the server */
/* UPDATE command instead of issuing an update for each volume. However, */
/* the 'update by volume' technique used here allows you to select */
/* a subset of volumes to be processed. */

upd vol "003900L4" acc=READO wherestg=COPYPOOL

end COPYSTGPOOL.VOLUMES.AVAILABLE macro

```

## COPYSTGPOOL.VOLUMES.DESTROYED

Contains a macro to mark copy storage pool volumes as unavailable if the volumes were onsite at the time of the disaster. This stanza does not include copy storage pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster. You can use the information as a guide and issue the administrative commands from a command line, or you can copy it to a file, modify it, and run it. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

After a disaster, compare the copy storage pool volumes listed in this stanza with the volumes that were left onsite. If you have any of the volumes and they are usable, you should remove their entries from this stanza.

```

begin COPYSTGPOOL.VOLUMES.DESTROYED macro

/* Purpose: Mark destroyed copy storage pool volumes as unavailable. */
/* Volumes in this macro were not marked as 'offsite' at the time the */
/* PREPARE ran. They were likely destroyed in the disaster. */
/* Recovery Administrator: Remove any volumes that were not destroyed. */

end COPYSTGPOOL.VOLUMES.DESTROYED macro

```

## Active-data storage pool volumes stanzas

The active-data storage pool volumes stanzas contain macros to mark active-data storage pool volumes as available or unavailable.

### ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE

Contains a macro to mark active-data pool volumes that were moved offsite and then moved back onsite. This stanza does not include active-data pool virtual volumes. You can use the information as a guide and issue the administrative commands, or you can copy it to a file, modify it, and run it. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

After a disaster, compare the active-data pool volumes listed in this stanza with the volumes that were moved back onsite. You should remove entries from this stanza for any missing volumes.

```

begin ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE macro

/* Purpose: Mark active-data storage pool volumes as available for use in recovery. */
/* Recovery Administrator: Remove any volumes that have not been obtained */
/* from the vault or are not available for any reason. */
/* Note: It is possible to use the mass update capability of the server */
/* UPDATE command instead of issuing an update for each volume. However, */
/* the 'update by volume' technique used here allows you to select */
/* a subset of volumes to be processed. */

upd vol "003901L4" acc=READ0 wherestg=ADP1

end ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE macro

```

## ACTIVEDATASTGPOOL.VOLUMES.DESTROYED

Contains a macro to mark active-data pool volumes as unavailable if the volumes were onsite at the time of the disaster. This stanza does not include active-data pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster. You can use the information as a guide and issue the administrative commands from a command line, or you can copy it to a file, modify it, and run it. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

After a disaster, compare the active-data pool volumes listed in this stanza with the volumes that were left onsite. If you have any of the volumes and they are usable, you should remove their entries from this stanza.

```

begin ACTIVEDATASTGPOOL.VOLUMES.DESTROYED macro

/* Purpose: Mark destroyed active-data storage pool volumes as unavailable. */
/* Volumes in this macro were not marked as 'offsite' at the time the */
/* PREPARE ran. They were likely destroyed in the disaster. */
/* Recovery Administrator: Remove any volumes that were not destroyed. */

end ACTIVEDATASTGPOOL.VOLUMES.DESTROYED macro

```

## Primary storage pool volumes stanzas

These stanzas contain a macro to mark primary storage pool volumes as *destroyed*.

### PRIMARY.VOLUMES.DESTROYED

Contains a macro to mark primary storage pool volumes as *destroyed* if the volumes were onsite at the time of disaster. You can use the information as a guide and run the administrative commands from a command line, or you can copy it to a file, modify it, and run it. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

During recovery, compare the primary storage pool volumes listed in this stanza with the volumes that were onsite. If you have any of the volumes and they are usable, remove their entries from the stanza.

This stanza does not include primary storage pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster.

```

begin PRIMARY.VOLUMES.DESTROYED macro

/* Purpose: Mark primary storage pool volumes as ACCESS=DESTROYED. */
/* Recovery administrator: Delete any volumes listed here */
/* that you do not want to recover. */
/* Note: It is possible to use the mass update capability of the server */
/* UPDATE command instead of issuing an update for each volume. However */
/* the 'update by volume' technique used here allows you to select */
/* a subset of volumes to be marked as destroyed. */

vary offline "D:\DISK.DSM" wait=yes
upd vol "D:\DISK.DSM" acc=DESTROYED wherestg=PRIMP00L1

end PRIMARY.VOLUMES.DESTROYED macro

```

## PRIMARY.VOLUMES.REPLACEMENT

Contains a macro to define primary storage pool volumes to the server. You can use the macro as a guide and run the administrative commands from a command line, or you can copy it to a file, modify it, and execute it. This macro is invoked by the RECOVERY.SCRIPT.NORMAL.MODE script.

Primary storage pool volumes with entries in this stanza have at least one of the following three characteristics:

- Original volume in a storage pool whose device class was DISK.
- Original volume in a storage pool with MAXSCRATCH=0.
- Original volume in a storage pool and volume scratch attribute=no.

The SET DRMPLANVPOSTFIX command adds a character to the end of the names of the original volumes listed in this stanza. This character does the following:

- Improves the retrievability of volume names that must be renamed in the stanzas. Before using the volume names, change these names to new names that are valid for the device class on the replacement system.
- Generates a new name that can be used by the replacement server. Your naming convention must take into account the appended character.

### Note:

1. Replacement primary volume names must be different from any other original volume name or replacement name.
2. The RESTORE STGPOOL command restores storage pools on a logical basis. There is no one-to-one relationship between an original volume and its replacement.
3. There could be entries for the same volume in PRIMARY.VOLUMES.REPLACEMENT if the volume has a device class of DISK.

This stanza does not include primary storage pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster.

## Primary storage volumes replacement stanza

```
begin PRIMARY.VOLUMES.REPLACEMENT macro

/* Purpose: Define replacement primary storage pool volumes for either: */
/* 1. Original volume in a storage pool whose device class was DISK. */
/* 2. Original volume in a storage pool with MAXSCRATCH=0. */
/* 3. Original volume in a storage pool and volume scratch=no. */
/* Recovery administrator: Edit this section for your replacement */
/* volume names. New name must be unique, i.e. different from any */
/* original or other new name. */

/* Replace D:\DISK.DSM DISK 8,096.0M in PRIMPOOL1 */
def vol PRIMPOOL1 "D:\DISK.DSMX" acc=READW f=8,096 wait=yes

end PRIMARY.VOLUMES.REPLACEMENT macro
```

## Storage pools restore stanza

This stanza contains a macro to restore the primary storage pools.

### STGPOOLS.RESTORE

You can use the stanza as a guide and execute the administrative commands from a command line. You can also copy it to a file, modify it, and execute it. This macro is invoked by the RECOVERY.SCRIPT.NORMAL.MODE script.

This stanza does not include primary storage pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster.

```
egin STGPOOLS.RESTORE macro

/* Purpose: Restore the primary storage pools from copy storage pool(s). */
/* Recovery Administrator: Delete entries for any primary storage pools */
/* that you do not want to restore. */

restore stgp PRIMPOOL1

end STGPOOLS.RESTORE macro
```

## Configuration stanzas

These stanzas contain copies of the following information: volume history, device configuration, and server options.

### VOLUME.HISTORY.FILE

Contains a copy of the volume history information when the recovery plan was created. The DSMSEV RESTORE DB command uses the volume history file to determine what volumes are needed to restore the database. It is used by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

The following rules determine where to place the volume history file at restore time:

- If the server option file contains VOLUMEHISTORY options, the server uses the fully qualified file name associated with the first entry. If the file name does not begin with a directory specification, the server uses the prefix *vollhprefix*.
- If the server option file does not contain VOLUMEHISTORY options, the server uses the default name *vollhprefix* followed by *drmvollh.txt*. The directory where the server is started from is used as the *vollhprefix*.

If a fully qualified file name was not specified in the server options file for the VOLUMEHISTORY option, the server adds it to the DSMSERV.OPT.FILE stanza.

### DEVICE.CONFIGURATION.FILE

Contains a copy of the server device configuration information when the recovery plan was created. The DSMSERV RESTORE DB command uses the device configuration file to read the database backup volumes. It is used by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

At recovery time, you may need to modify this stanza. You must update the device configuration information if the hardware configuration at the recovery site has changed. Examples of changes requiring updates to the configuration information are:

- Different device names
- Use of a manual library instead of an automated library
- For automated libraries, the requirement to manually place the database backup volumes in the automated library and update the configuration information to identify the element within the library. This allows the server to locate the required database backup volumes.

For details, see “Updating the device configuration file” on page 776.

The following rules determine where the device configuration file is placed at restore time:

- If the server options file contains DEVCONFIG entries, the server uses the fully qualified file name associated with the first entry. If the specified file name does not begin with a directory specification, the server adds the prefix *devcprefix*.
- If the server options file does not contain DEVCONFIG entries, the server uses the default name *devcprefix* followed by *drmdevc.txt*.

For example, if *devcprefix* is /opt/tivoli/tsm/server/bin, the file name used by PREPARE is /opt/tivoli/tsm/server/bin/drmdevc.txt.

**Note:** The *devcprefix* is set based on the following:

- If the environmental variable DSMSERV\_DIR has been defined, it is used as the *devcprefix*.
- If the environmental variable DSMSERV\_DIR has not been defined, the directory where the server is started from is used as the *devcprefix*.

If a fully qualified file name was not specified for the DEVCONFIG option in the server options file, the server adds it to the stanza DSMSERV.OPT.FILE.

### DSMSERV.OPT.FILE

Contains a copy of the server options file. This stanza is used by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

**Note:** The following figure contains text strings that are too long to display in hardcopy or softcopy publications. The long text strings have a plus symbol (+) at the end of the string to indicate that they continue on the next line.

The disaster recovery plan file adds the DISABLESCHEDULES option to the server options file and sets it to YES. This option disables administrative and client

schedules while the server is being recovered. After the server is recovered, you can enable scheduling by deleting the option or setting it to NO and then restarting the server.

### License information stanza

The license information stanza contains a copy of the latest license audit results and the server license terms.

#### LICENSE.INFORMATION

```
begin LICENSE.INFORMATION
 Last License Audit: 09/26/2008 12:02:35
 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
end LICENSE.INFORMATION
```

### Machine files stanza

These stanzas contain information needed to rebuild the server machine.

#### MACHINE.GENERAL.INFORMATION

Provides information for the server machine (for example, machine location). This stanza is included in the plan file if the machine information is saved in the database using the DEFINE MACHINE with ADSMSERVER=YES.

```
begin MACHINE.GENERAL.INFORMATION
Purpose: General information for machine DSMSRV1.
 This is the machine that contains DSM server DSM.
 Machine Name: DSMSRV1
 Machine Priority: 1
 Building: 21
 Floor: 2
 Room: 2749
 Description: DSM Server for Branch 51
 Recovery Media Name: DSMSRVIMAGE
end MACHINE.GENERAL.INFORMATION
```

#### MACHINE.RECOVERY.INSTRUCTIONS

Provides the recovery instructions for the server machine. This stanza is included in the plan file if the machine recovery instructions are saved in the database.

```
begin MACHINE.RECOVERY.INSTRUCTIONS
Purpose: Recovery instructions for machine DSMSRV1.

Primary Contact:
 Jane Smith (wk 520-000-0000 hm 520-001-0001)
Secondary Contact:
 John Adams (wk 520-000-0001 hm 520-002-0002)
end MACHINE.RECOVERY.INSTRUCTIONS
```

#### MACHINE.RECOVERY.CHARACTERISTICS

Provides the hardware and software characteristics for the server machine. This stanza is included in the plan file if the machine characteristics are saved in the database.

```
begin MACHINE.CHARACTERISTICS
Purpose: Hardware and software characteristics of machine DSMSRV1.
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/ida/c0d0p8 1032056 953908 25724 98% /
/dev/ida/c0d0p1 518000 49500 442188 11% /boot
/dev/ida/c0d0p3 5670432 3840664 1541720 72% /devel
/dev/ida/c0d0p6 9573932 1341124 7746476 15% /drivers
/dev/ida/c0d0p2 3096288 2751896 187108 94% /home
/dev/ida/c0d0p5 2064160 698048 1261260 36% /pkgs
none 191664 0 191664 0% /dev/shm
/dev/ida/c0d0p7 4104144 1369168 2526492 36% /usr
/dev/ida/c0d0p10 518000 44280 447408 10% /var
/dev/ida/c0d0p11 5835056 20 5538628 1% /webend MACHINE.CHARACTERISTICS
```

## MACHINE.RECOVERY.MEDIA

Provides information about the media (for example, boot media) needed for rebuilding the machine that contains the server. This stanza is included in the plan file if recovery media information is saved in the database and it has been associated with the machine that contains the server.

```
begin MACHINE.RECOVERY.MEDIA.REQUIRED
Purpose: Recovery media for machine DSMSRV1.
Recovery Media Name: DSMSRV
Type: Other
Volume Names:
Location: IRONMNT
Description: Server Installation CD
Product:
Product Information:

Recovery Media Name: LINUXOS
Type: Other
Volume Names:
Location: IRONMNT
Description: Linux/xxxx Installation CD
Product:
Product Information:
end MACHINE.RECOVERY.MEDIA.REQUIRED
```



---

## Part 6. Appendixes



---

## Appendix A. External media management interface description

The programming interface that IBM Tivoli Storage Manager provides to external media-management programs consists of request description strings that IBM Tivoli Storage Manager sends and response strings that the external program sends.

To use the interface, you must first define an EXTERNAL-type Tivoli Storage Manager library that represents the media manager. You do not define drives, label volumes, or check in media. Refer to your media manager's documentation for that product's setup information and instructions for operational usage.

See "Configuration for libraries controlled by media manager programs" on page 120

The details of the request types and the required processing are described in the sections that follow. The request types are:

- Initialization of the external program
- Begin Batch
- End Batch
- Volume Query
- Volume Eject
- Volume Release
- Volume Mount
- Volume Dismount

The responses can be right-padded with any number of white-space characters.

The libraryname passed in a request must be returned in the response. The volume specified in an eject request or a query request must be returned in the response. The volume specified in a mount request (except for 'SCRATCH') must be returned in the response. When 'SCRATCH' is specified in a mount request, the actual volume mounted must be returned.

---

### CreateProcess call

The server creates two anonymous unidirectional pipes and maps them to the stdin and stdout streams during the CreateProcess call. When a standard handle is redirected to refer to a file or a pipe, the handle can only be used by the ReadFile and WriteFile functions.

This precludes normal C functions such as gets or printf. Since the server will never terminate the external program process, it is imperative that the external program recognize a read or write failure on the pipes and exit the process. In addition, the external program should exit the process if it reads an unrecognized command.

The external program may obtain values for the read and write handles using the following calls:

```
readPipe=GetStdHandle(STD_INPUT_HANDLE)

and

writePipe=GetStdHandle(STD_OUTPUT_HANDLE)
```

---

## Processing during server initialization

Ensure that the external media management program works with the server during the server's initialization.

For each external library defined to the server, the following must occur during server initialization:

1. The server loads the external program (CreateProcess) in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string, in text form, into the standard input (stdin) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string, in text form, into its standard output (stdout) stream.
4. The server closes the pipes.
5. When the agent detects that the pipes are closed, it performs any necessary cleanup and calls the `stdlib` exit routine.

---

## Processing for mount requests

To process the mount request, the server has to complete certain steps.

The server completes the following steps to process mount requests:

1. The server loads the external program in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string (in text form) into the standard input (stdin) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string (in text form) into its standard output (stdout) stream.
4. The server sends the MOUNT request (stdin).
5. The agent sends the MOUNT response (stdout).
6. The agent waits.
7. The server sends the DISMOUNT request (stdin).
8. The agent sends the DISMOUNT response (stdout), performs any necessary cleanup, and calls the `stdlib` exit routine.

---

## Processing for release requests

To process release requests, the server has to complete certain steps.

The server completes the following steps to process release requests:

1. The server loads the external program in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string (in text form) into the standard input (stdin) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string (in text form) into its standard output (stdout) stream.
4. The server sends the RELEASE request (stdin).
5. The agent sends the RELEASE response (stdout), performs any necessary cleanup, and calls the `stdlib` exit routine.

---

## Processing for batch requests

Batch processing is done during MOVE MEDIA, MOVE DRMEDIA, and QUERY MEDIA command processing when performed on volumes in external libraries.

The move commands cause a QUERY to be issued for a volume. If the QUERY indicates that the volume is in the library, a subsequent EJECT for that volume is issued. Because the move commands can match any number of volumes, a QUERY and an EJECT request is issued for each matching volume.

The QUERY MEDIA command results in QUERY requests being sent to the agent. During certain types of processing, Tivoli Storage Manager might need to know if a volume is present in a library. The external agent should verify that the volume is physically present in the library.

1. The server loads the external program in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string (in text form) into the standard input (stdin) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string (in text form) into its standard output (stdout) stream.
4. The server sends the BEGIN BATCH request (stdin).
5. The agent sends the BEGIN BATCH response (stdout).
6. The server sends 1 to n volume requests ( $n > 1$ ). These can be any number of QUERY or EJECT requests. For each request, the agent will send the applicable QUERY response or EJECT response.
7. The server sends the END BATCH request (stdin).
8. The agent sends the END BATCH response (stdout), performs any necessary cleanup, and calls the `stdlib` exit routine.

---

## Error handling

If the server encounters an error during processing, it closes the stdin and stdout streams to the agent exit. The agent detects this when it attempts to read from stdin or write to stdout. If this occurs, the agent performs any necessary cleanup and calls the `stdlib` exit routine.

If the code for any response (except for EJECT and QUERY) is not equal to SUCCESS, Tivoli Storage Manager does not proceed with the subsequent steps. After the agent sends a non-SUCCESS return code for any response, the agent will perform any necessary cleanup and call the `stdlib` exit routine.

However, even if the code for EJECT or QUERY requests is not equal to SUCCESS, the agent will continue to send these requests.

If the server gets an error while trying to write to the agent, it will close the pipes, perform any necessary cleanup, and terminate the current request.

---

## Begin batch request

The programming interface includes a begin-batch description string and an external-program response string.

The format of the begin batch request is:

```
BEGIN BATCH
```

The format of the external program response is:

```
BEGIN BATCH COMPLETE, RESULT=resultCode
```

where:

*resultCode*

One of the following:

- SUCCESS
- INTERNAL\_ERROR

---

## End batch request

The end batch request is sent by Tivoli Storage Manager to indicate that no more requests are to be sent by the external library manager for the current process. The external agent must send the end batch response and end by using the `stdlib` exit routine.

The format of the end batch request is:

```
END BATCH
```

The format of the external program response is:

```
END BATCH COMPLETE, RESULT=resultCode
```

where *resultCode* is SUCCESS or INTERNAL\_ERROR.

---

## Volume query request

The programming interface includes a volume-query-request description string and an external-program response string.

The format of the volume query request is:

```
QUERY libraryname volume
```

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volume*

Specifies the volume name to be queried.

The format of the external program response is:

```
QUERY libraryname volume COMPLETE, STATUS=statusValue, RESULT=resultCode
```

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volume*

Specifies the volume name queried.

*resultCode*

One of the following:

- SUCCESS
- LIBRARY\_ERROR
- VOLUME\_UNKNOWN
- VOLUME\_UNAVAILABLE
- CANCELLED
- TIMED\_OUT
- INTERNAL\_ERROR

If *resultCode* is not SUCCESS, the exit must return *statusValue* set to UNDEFINED.

If *resultCode* is SUCCESS, STATUS must be one of the following values:

- IN\_LIBRARY
- NOT\_IN\_LIBRARY

IN\_LIBRARY means that the volume is currently in the library and available to be mounted.

NOT\_IN\_LIBRARY means that the volume is not currently in the library.

---

## Initialization requests

When the server is started, it sends an initialization request to the external media management program for each EXTERNAL library. The external program processes this request to ensure that the external program is present, functional, and ready to process requests.

If the initialization request is successful, Tivoli Storage Manager informs its operators that the external program reported its readiness for operations. Otherwise, Tivoli Storage Manager reports a failure to its operators.

Tivoli Storage Manager does not attempt any other type of operation with that library until an initialization request has succeeded. The server sends an initialization request first. If the initialization is successful, the request is sent. If the initialization is not successful, the request fails. The external media management program can detect whether the initialization request is being sent by itself or with another request by detecting end-of-file on the stdin stream. When end-of-file is detected, the external program must end by using the `stdlib` exit routine (not the `return` call).

When a valid response is sent by the external program, the external program must end by using the exit routine.

### Format of the request:

```
INITIALIZE libraryname
```

where *libraryname* is the name of the EXTERNAL library as defined to Tivoli Storage Manager.

### Format of the external program response:

```
INITIALIZE libraryname COMPLETE, RESULT=resultcode
```

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*resultcode*

One of the following:

- SUCCESS
- NOT\_READY
- INTERNAL\_ERROR

---

## Volume eject request

The Programming Interface includes a volume-eject-request description string and an external-program response string.

The format of the volume eject request is:

```
EJECT libraryname volume 'location info'
```

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volume*

Specifies the volume to be ejected.

*'location info'*

Specifies the location information associated with the volume from the Tivoli Storage Manager inventory. It is delimited with single quotation marks. This information is passed without any modification from the Tivoli Storage Manager inventory. The customer is responsible for setting its contents with the appropriate UPDATE MEDIA or UPDATE VOLUME command before the move command is invoked. Set this field to some target location value that will assist in placing the volume after it is ejected from the library. It is suggested that the external agent post the value of this field to the operator.

The format of the external program response is:

```
EJECT libraryname volume COMPLETE, RESULT=resultCode
```

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volume*

Specifies the ejected volume.

*resultCode*

One of the following:

- SUCCESS
- LIBRARY\_ERROR
- VOLUME\_UNKNOWN
- VOLUME\_UNAVAILABLE
- CANCELLED
- TIMED\_OUT
- INTERNAL\_ERROR

---

## Volume release request

When the server returns a volume to scratch status, the server starts the external media management program, issues a request to initialize, then issues a request to release a volume.

The external program must send a response to the release request. No matter what response is received from the external program, Tivoli Storage Manager returns the volume to scratch. For this reason, Tivoli Storage Manager and the external program can have conflicting information on which volumes are scratch. If an error occurs, the external program should log the failure so that the external library inventory can be synchronized later with Tivoli Storage Manager. The synchronization can be a manual operation.

The format of the request is:

```
RELEASE libraryname volname
```

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volname*

Specifies the name of the volume to be returned to scratch (released).

The format of the external program response is:

```
RELEASE libraryname volname COMPLETE, RESULT=resultcode
```

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volname*

Specifies the name of the volume returned to scratch (released).

*resultcode*

One of the following:

- SUCCESS
- VOLUME\_UNKNOWN
- VOLUME\_UNAVAILABLE
- INTERNAL\_ERROR

---

## Volume mount request

When the server requires a volume mount, the server starts the external media management program, issues a request to initialize, then issues a request to mount a volume. The external program is responsible for verifying that this request is coming from Tivoli Storage Manager and not from an unauthorized system.

The volume mounted by the external media management program must be a tape with a standard IBM label that matches the external volume label. When the external program completes the mount request, the program must send a response. If the mount was successful, the external program must remain active. If the mount failed, the external program must end immediately by using the `stdlib` exit routine.

The format of the request is:

```
MOUNT libraryname volname accessmode devicetypes timelimit userid
volumenumber 'location'
```

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volname*

Specifies the actual volume name if the request is for an existing volume. If a scratch mount is requested, the *volname* is set to `SCRATCH`.

*accessmode*

Specifies the access mode required for the volume. Possible values are `READONLY` and `READWRITE`.

*devicetypes*

Specifies a list of device types that can be used to satisfy the request for the volume and the FORMAT specified in the device class. The most preferred device type is first in the list. Items are separated by commas, with no intervening spaces. Possible values are:

- 3590
- 3590E
- 3590H
- 3592
- 4MM\_DDS1
- 4MM\_DDS1C
- 4MM\_DDS2
- 4MM\_DDS2C
- 4MM\_DDS3
- 4MM\_DDS3C
- 4MM\_HP\_DDS4
- 4MM\_DDS4
- 4MM\_HP\_DDS4C
- 4MM\_DDS4C
- 4MM\_DDS5C
- 4MM\_DDS6C
- 8MM\_SAIT
- 8MM\_VXA2
- 8MM\_VXA3
- 8MM\_AITC
- 8MM\_AIT
- 8MM\_8200
- 8MM\_ELIANT
- 8MM\_8500
- 8MM\_8500C
- 8MM\_8205
- 8MM\_8900
- 8MM\_M2
- DLT\_2000
- DLT\_4000
- DLT\_7000
- DLT\_8000
- SDLT
- SDLT320
- DLT1
- DLT2
- SDLT600
- LTO\_ULTRIUM
- LTO\_ULTRIUM2
- LTO\_ULTRIUM3
- OPT\_UDO\_60GB

- QIC\_525
- QIC\_IBM1000
- IBM\_QIC4GBC
- QIC\_12GBC
- QIC\_5010C
- QIC\_20GBC
- QIC\_25GBC
- QIC\_30GBC
- QIC\_50GBC
- QIC\_70GBC
- REMOVABLEFILE
- M8100
- STK\_SD3
- STK\_9840
- STK\_T9840C
- STK\_9940
- STK\_9940B
- STK\_9840\_VOLSAFE
- STK\_T9840C\_VOLSAFE
- STK\_9940\_VOLSAFE
- STK\_9940B\_VOLSAFE
- STK\_T10000A

*timelimit*

Specifies the maximum number of minutes that the server waits for the volume to be mounted. If the mount request is not completed within this time, the external manager responds with the result code TIMED\_OUT.

*userid*

Specifies the user ID of the process that needs access to the drive.

*volumenumber*

The *volumenumber* is 1.

*'location'*

Specifies the value of the location field from the Tivoli Storage Manager inventory (for example, 'Room 617 Floor 2'). One blank character is inserted between the volume number and the left single quotation mark in the location information. If no location information is associated with a volume, nothing is passed to the exit. If no volume information exists, the single quotation marks are not passed. Also, if volume information is passed, then probably the volume has been ejected from the library and needs to be returned to the library before the mount operation can proceed. The location information should be posted by the agent so that the operator can obtain the volume and return it to the library.

The format of the external program response is:

MOUNT *libraryname volname* COMPLETE ON *specialfile*, RESULT=*resultcode*

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volname*

Specifies the name of the volume mounted for the request.

*specialfile*

The fully qualified path name of the device special file for the drive in which the volume was mounted. If the mount request fails, the value should be set to /dev/null.

The external program must ensure that the special file is closed before the response is returned to the server.

*resultcode*

One of the following:

- SUCCESS
- DRIVE\_ERROR
- LIBRARY\_ERROR
- VOLUME\_UNKNOWN
- VOLUME\_UNAVAILABLE
- CANCELLED
- TIMED\_OUT
- INTERNAL\_ERROR

---

## Volume dismount request

When a successful mount operation completes, the external process must wait for a request to dismount the volume. When the dismount operation completes, the external program must send a response to the server.

After the dismount response is sent, the external process ends immediately by using the `stdlib` exit routine.

### Format of the request:

`DISMOUNT libraryname volname`

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volname*

Specifies the name of the volume to be dismounted.

### Format of the external program response:

`DISMOUNT libraryname volname COMPLETE, RESULT=resultcode`

where:

*libraryname*

Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

*volname*

Specifies the name of the volume dismounted.

*resultcode*

One of the following:

- SUCCESS
- DRIVE\_ERROR
- LIBRARY\_ERROR
- INTERNAL\_ERROR

---

## Appendix B. User exit and file exit receivers

The data structure of the user exit receivers applies to the file exit receivers. To use one of these exits with Tivoli Storage Manager, you must specify the corresponding server option (FILEEXIT, FILETEXTEXIT, or USEREXIT) in the server options file.

The samples for the C, H, and make files are shipped with the server code in the /opt/tivoli/tsm/server/bin directory.

### Attention:

1. Use caution in modifying these exits. A user exit abend will bring down the server.
2. The file specified in the file exit option will continue to grow unless you prune it.

You can also use Tivoli Storage Manager commands to control event logging. For details, see “Logging IBM Tivoli Storage Manager events to receivers” on page 626 and *Administrator’s Reference*.

---

## Sample user exit declarations

userExitSample.h contains declarations for a user-exit program.

The environment is:

Linux/i386

Figure 105. Sample user exit declarations

```
/*
 * Name: userExitSample.h
 * Description: Declarations for a user exit
 */
#define _H_USEREXITSAMPLE

#include <stdio.h>
#include <sys/types.h>

/***** Do not modify below this line. *****/

#define BASE_YEAR 1900

typedef short int16;
typedef int int32;

/* uchar is usually defined in <sys/types.h> */
/* DateTime Structure Definitions - TSM representation of a timestamp*/

typedef struct
{
 uchar year; /* Years since BASE_YEAR (0-255) */
 uchar mon; /* Month (1 - 12) */
 uchar day; /* Day (1 - 31) */
 uchar hour; /* Hour (0 - 23) */
}
```

```

 uchar min; /* Minutes (0 - 59) */
 uchar sec; /* Seconds (0 - 59) */
} DateTime;

/*****
 * Some field size definitions (in bytes) *
 *****/

#define MAX_SERVERNAME_LENGTH 64
#define MAX_NODE_LENGTH 64
#define MAX_COMMNAME_LENGTH 16
#define MAX_OWNER_LENGTH 64
#define MAX_HL_ADDRESS 64
#define MAX_LL_ADDRESS 32
#define MAX_SCHED_LENGTH 30
#define MAX_DOMAIN_LENGTH 30
#define MAX_MSGTEXT_LENGTH 1600

/*****
 * Event Types (in e1EventRecvData.eventType) *
 *****/

#define TSM_SERVER_EVENT 0x03 /* Server Events */
#define TSM_CLIENT_EVENT 0x05 /* Client Events */

/*****
 * Application Types (in e1EventRecvData.applType) *
 *****/

#define TSM_APPL_BACKARCH 1 /* Backup or Archive client */
#define TSM_APPL_HSM 2 /* Space manage client */
#define TSM_APPL_API 3 /* API client */
#define TSM_APPL_SERVER 4 /* Server (ie. server to server) */

/*****
 * Event Severity Codes (in e1EventRecvData.sevCode) *
 *****/

#define TSM_SEV_INFO 0x02 /* Informational message. */
#define TSM_SEV_WARNING 0x03 /* Warning message. */
/*
#define TSM_SEV_ERROR 0x04 /* Error message. */
#define TSM_SEV_SEVERE 0x05 /* Severe error message. */
#define TSM_SEV_DIAGNOSTIC 0x06 /* Diagnostic message. */
#define TSM_SEV_TEXT 0x07 /* Text message. */

/*****
 * Data Structure of Event that is passed to the User-Exit. *
 * This data structure is the same for a file generated using *
 * the FILEEXIT option on the server. *
 *****/

typedef struct evRdata
{
 int32 eventNum; /* the event number. */
 int16 sevCode; /* event severity. */
 int16 applType; /* application type (hsm, api, etc)*/
 int32 sessId; /* session number */
 int32 version; /* Version number of this structure (1)*/
 int32 eventType; /* event type
 * (TSM_CLIENT_EVENT, TSM_SERVER_EVENT)*/
 DateTime timeStamp; /* timestamp for event data. */
 uchar serverName[MAX_SERVERNAME_LENGTH+1]; /* server name */
 uchar nodeName[MAX_NODE_LENGTH+1]; /* Node name for session */
 uchar commMethod[MAX_COMMNAME_LENGTH+1]; /* communication method */
 uchar ownerName[MAX_OWNER_LENGTH+1]; /* owner */
 uchar hlAddress[MAX_HL_ADDRESS+1]; /* high-level address */

```

```

 uchar llAddress[MAX_LL_ADDRESS+1]; /* low-level address */
 uchar schedName[MAX_SCHED_LENGTH+1]; /* schedule name if applicable*/
 uchar domainName[MAX_DOMAIN_LENGTH+1]; /* domain name for node */
 uchar event[MAX_MSGTEXT_LENGTH]; /* event text */
} e1EventRecvData;

/*****
 * Size of the Event data structure *
 *****/

#define ELEVENTRECVDATA_SIZE sizeof(e1EventRecvData)

/*****
 * User Exit EventNumber for Exiting *
 *****/

#define USEREXIT_END_EVENTNUM 1822 /* Only user-exit receiver to exit*/
#define END_ALL_RECEIVER_EVENTNUM 1823 /* All receivers told to exit */

/*****
 *** Do not modify above this line. ***
 *****/

/***** Additional Declarations *****/

#endif

```

---

## Sample user-exit program

userExitSample.c is a sample user-exit program invoked by the server.

Figure 106. Sample user exit program

```

/*****
 * Name: userExitSample.c
 * Description: Example user-exit program invoked by the server
 * Environment: Linux/i386
 *****/

#include <stdio.h>
#include "userExitSample.h"

/*****
 *** Do not modify below this line. ***
 *****/

extern void adsmV3UserExit(void *anEvent);

/*****
 *** Main ***
 *****/

int main(int argc, char *argv[])
{
/* Do nothing, main() is never invoked, but stub is needed */

exit(0); /* For picky compilers */

} /* End of main() */

/*****
 * Procedure: adsmV3UserExit
 * If the user-exit is specified on the server, a valid and
 * appropriate event causes an e1EventRecvData structure (see

```

```

* userExitSample.h) to be passed to adsmV3UserExit that returns a void.
* INPUT : A (void *) to the elEventRecvData structure
* RETURNS: Nothing
*****/

void adsmV3UserExit(void *anEvent)
{
/* Typecast the event data passed */
elEventRecvData *eventData = (elEventRecvData *)anEvent;

/*****
*** Do not modify above this line. ***
*****/

if((eventData->eventNum == USEREXIT_END_EVENTNUM) ||
 (eventData->eventNum == END_ALL_RECEIVER_EVENTNUM))
{
/* Server says to end this user-exit. Perform any cleanup, *
 * but do NOT exit() !!! */
return;
}

/* Field Access: eventData->... */
/* Your code here ... */

/* Be aware that certain function calls are process-wide and can cause
 * synchronization of all threads running under the TSM Server process!
 * Among these is the system() function call. Use of this call can
 * cause the server process to hang and otherwise affect performance.
 * Also avoid any functions that are not thread-safe. Consult your
 * system's programming reference material for more information.
 */

return; /* For picky compilers */
} /* End of adsmV3UserExit() */

```

---

## Readable text file exit (FILETEXTEXIT) format

If you specify the readable text file exit (FILETEXTEXIT), each logged event is written to a fixed-size, readable line.

The following table presents the format of the output. Fields are separated by blank spaces.

*Table 69. Readable text file exit (FILETEXTEXIT) format*

| Column                 | Description                            |
|------------------------|----------------------------------------|
| 0001-0006              | Event number (with leading zeros)      |
| 0008-0010              | Severity code number                   |
| 0012-0013              | Application type number                |
| 0015-0023              | Session ID number                      |
| 0025-0027              | Event structure version number         |
| 0029-0031              | Event type number                      |
| 0033-0046              | Date/Time (YYYYMMDDHHmmSS)             |
| 0048-0111              | Server name (right padded with spaces) |
| 0113-0176 <sup>1</sup> | Node name                              |
| 0178-0193 <sup>1</sup> | Communications method name             |
| 0195-0258 <sup>1</sup> | Owner name                             |

Table 69. Readable text file exit (FILETEXEXIT) format (continued)

| Column                 | Description                                  |
|------------------------|----------------------------------------------|
| 0260-0323 <sup>1</sup> | High-level internet address (n.n.n.n)        |
| 0325-0356 <sup>1</sup> | Port number from high-level internet address |
| 0358-0387 <sup>1</sup> | Schedule name                                |
| 0389-0418 <sup>1</sup> | Domain name                                  |
| 0420-2019              | Event text                                   |
| 2020-2499              | Unused spaces                                |
| 2500                   | New line character                           |

<sup>1</sup> Columns 113 - 418 contain data only for events that originate in a client or in another Tivoli Storage Manager server. Otherwise, columns 113 - 418 contain blanks.



---

## Appendix C. Accessibility features for Tivoli Storage Manager

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

### Accessibility features

The following list includes the major accessibility features in Tivoli Storage Manager:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers
- Keys that are discernible by touch but do not activate just by touching them
- Industry-standard devices for ports and connectors
- The attachment of alternative input and output devices
- User documentation provided in HTML and PDF format. Descriptive text is provided for all documentation images.

The Tivoli Storage Manager Information Center, and its related publications, are accessibility-enabled.

### Keyboard navigation

Tivoli Storage Manager follows Linux operating-system conventions for keyboard navigation and access.

### Vendor software

Tivoli Storage Manager includes certain vendor software that is not covered under the IBM license agreement. IBM makes no representation about the accessibility features of these products. Contact the vendor for the accessibility information about its products.

### Related accessibility information

You can view the publications for Tivoli Storage Manager in Adobe® Portable Document Format (PDF) using the Adobe Acrobat Reader. You can access these or any of the other documentation PDFs at the IBM Publications Center at <http://www.ibm.com/shop/publications/order/>.

### IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility: <http://www.ibm.com/able>.



---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

*IBM Corporation  
2Z4A/101  
11400 Burnet Road  
Austin, TX 78758  
U.S.A.*

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

IBM, the IBM logo, and `ibm.com`<sup>®</sup> 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 <http://www.ibm.com/legal/copytrade.shtml>.

Adobe is either a registered trademark or trademark of Adobe Systems Incorporated in the United States, other countries, or both.

Intel<sup>®</sup> and Itanium<sup>®</sup> are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.



**Java**<sup>™</sup>

**COMPATIBLE** Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.



---

# Glossary

This glossary includes terms and definitions for IBM Tivoli Storage Manager.

To view glossaries for other IBM products, go to <http://www.ibm.com/software/globalization/terminology/>.

The following cross-references are used in this glossary:

- *See* refers the reader from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- *See also* refers the reader to a related or contrasting term.

## A

### **absolute mode**

In storage management, a backup copy-group mode that specifies that a file is considered for incremental backup even if the file has not changed since the last backup. See also *modified mode*.

### **access control list (ACL)**

In computer security, a list associated with an object that identifies all the subjects that can access the object and their access rights. For example, an access control list is associated with a file that identifies the users who can access that file and their access rights.

### **access mode**

An attribute of a storage pool or a storage volume that specifies whether the server can write to or read from the storage pool or storage volume. The access mode can be read/write, read-only, or unavailable. Volumes in primary storage pools can also have an access mode of destroyed. Volumes in copy storage pools can also have an access mode of offsite.

### **acknowledgment**

The transmission of acknowledgment characters as a positive response to a data transmission.

**ACL** See *access control list*.

### **activate**

To validate the contents of a policy set and then make it the active policy set.

### **active-data pool**

A named set of storage pool volumes that contain only active versions of client backup data.

### **active file system**

A file system to which space management has been added. With space management, tasks for an active file system include automatic migration, reconciliation, selective migration, and recall. Contrast with *inactive file system*.

### **active policy set**

The activated policy set that contains the policy rules in use by all client nodes that are assigned to the policy domain. See also *policy domain* and *policy set*.

### **active version**

The most recent backup copy of a file stored. The active version of a file cannot be deleted until a backup process detects that the user has either replaced the file with a newer version or has deleted the file from the file server or workstation. Contrast with *inactive version*.

### **activity log**

A log that records normal activity messages that are generated by the server. These messages include information about server and client operations, such as the start time of sessions or device I/O errors.

### **adaptive subfile backup**

A type of backup that sends only changed portions of a file to the server, instead of sending the entire file. Adaptive subfile backup reduces network traffic and increases the speed of the backup.

### **administrative client**

A program that runs on a file server, workstation, or mainframe that administrators use to control and monitor the Tivoli Storage Manager server. Contrast with *backup-archive client*.

### **administrative command schedule**

A database record that describes the planned processing of an administrative

command during a specific time period.  
See also *client schedule*.

**administrative privilege class**

See *privilege class*.

**administrative session**

A period of time during which an administrator user ID communicates with a server to perform administrative tasks. Contrast with *client node session*.

**administrator**

A user who is registered to the server as an administrator, and who is authorized to perform tasks and issue commands through the assignment of an administrative privilege class.

**Advanced Program-to-Program Communication (APPC)**

An implementation of the SNA LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

**agent node**

A client node that has been granted proxy authority to perform operations on behalf of another client node, which is the target node.

**aggregate**

An object, stored in one or more storage pools, consisting of a group of logical files that are packaged together. See also *logical file* and *physical file*.

**aggregate data transfer rate**

A performance statistic that indicates the average number of bytes that were transferred per second while processing a given operation.

**APPC** See *Advanced Program-to-Program Communication*.

**application client**

A program that is installed on a system to protect an application. The Tivoli Storage Manager server provides backup services to an application client.

**archive**

To copy programs, data, or files to another storage media, usually for long-term storage or security. Contrast with *retrieve*.

**archive copy**

A file or group of files that was archived to server storage.

**archive copy group**

A policy object containing attributes that control the generation, destination, and expiration of archived files.

**archive-retention grace period**

The number of days that the storage manager retains an archived file when the server is unable to rebind the file to an appropriate management class. See also *bind*.

**association**

(1) The defined relationship between a client node and a client schedule. An association identifies the name of a schedule, the name of the policy domain to which the schedule belongs, and the name of a client node that performs scheduled operations.

(2) On a configuration manager, the defined relationship between a profile and an object such as a policy domain. Profile associations define the configuration information that is distributed to a managed server when it subscribes to the profile.

**audit**

To check for logical inconsistencies between information that the server has and the actual condition of the system. The storage manager can audit information about items such as volumes, libraries, and licenses. For example, when a storage manager audits a volume, the server checks for inconsistencies between information about backed-up or archived files that are stored in the database and the actual data that are associated with each backup version or archive copy in server storage.

**authentication**

The process of checking a user's password before permitting user access to the Tivoli Storage Manager server. Authentication can be turned on or off by an administrator with system privilege.

**authentication rule**

A specification that another user can use to either restore or retrieve files from storage.

**authority**

The right to access objects, resources, or functions. See also *privilege class*.

**authorization rule**

A specification that permits another user to either restore or retrieve a user's files from storage.

**authorized user**

A user who has administrative authority for the Tivoli Storage Manager client on a workstation. This user changes passwords, performs open registrations, and deletes file spaces.

**AutoFS**

See *automounted file system*.

**automatic detection**

A feature that detects, reports, and updates the serial number of a drive or library in the database when the path from the local server is defined.

**automatic migration**

The process that is used to automatically move files from a local file system to storage, based on options and settings that are chosen by a root user on a workstation. See also *threshold migration* and *demand migration*.

**automatic reconciliation**

The process that is used to reconcile file systems at regular intervals. The intervals are set by a user with root user authority. See also *reconciliation*.

**automounted file system (AutoFS)**

A file system that is managed by an automounter daemon. The automounter daemon monitors a specified directory path, and automatically mounts the file system to access data.

**B****backup-archive client**

A program that runs on a workstation or file server and provides a means for users to back up, archive, restore, and retrieve files. Contrast with *administrative client*.

**backup copy group**

A policy object containing attributes that control the generation, destination, and expiration of backup versions of files. A backup copy group belongs to a management class.

**backup-retention grace period**

The number of days the storage manager retains a backup version after the server is unable to rebind the file to an appropriate management class.

**backup set**

A portable, consolidated group of active versions of backup files that are generated for a backup-archive client.

**backup set collection**

A group of backup sets that are created at the same time and which have the same backup set name, volume names, description, and device classes. The server identifies each backup set in the collection by its node name, backup set name, and file type.

**backup version**

A file or directory that a client node backed up to server storage. More than one backup version can exist in server storage, but only one backup version is the active version. See also *active version* and *inactive version*.

**bind** To associate a file with a management class name. See *rebind*.

**bindery**

A database that consists of three system files for a NetWare server. The files contain user IDs and user restrictions.

**C**

**cache** To place a duplicate copy of a file on random access media when the server migrates a file to another storage pool in the hierarchy.

**cache file**

A snapshot of a logical volume created by Logical Volume Snapshot Agent. Blocks are saved immediately before they are modified during the image backup and their logical extents are saved in the cache files.

**CAD** See *client acceptor*.

**central scheduler**

A function that permits an administrator to schedule client operations and administrative commands. The operations can be scheduled to occur periodically or on a specific date. See *client schedule* and *administrative command schedule*.

**client** A software program or computer that requests services from a server.

**client acceptor**

An HTTP service that serves the Java applet for the Web client to Web browsers. On Windows systems, the client acceptor is installed and run as a service. On AIX, UNIX, and Linux systems, the client acceptor is run as a daemon, and is also called the *client acceptor daemon* (CAD).

**client acceptor daemon (CAD)**

See *client acceptor*.

**client domain**

The set of drives, file systems, or volumes that the user selects to back up or archive data, using the backup-archive client.

**client node**

A file server or workstation on which the backup-archive client program has been installed, and which has been registered to the server.

**client node session**

A session in which a client node communicates with a server to perform backup, restore, archive, retrieve, migrate, or recall requests. Contrast with *administrative session*.

**client options file**

An editable file that identifies the server and communication method, and provides the configuration for backup, archive, hierarchical storage management, and scheduling.

**client option set**

A group of options that are defined on the server and used on client nodes in conjunction with client options files.

**client-polling scheduling mode**

A method of operation in which the client queries the server for work. Contrast with *server-prompted scheduling mode*.

**client schedule**

A database record that describes the planned processing of a client operation during a specific time period. The client operation can be a backup, archive, restore, or retrieve operation, a client operating system command, or a macro. See also *administrative command schedule*.

**client/server**

Pertaining to the model of interaction in distributed data processing in which a program on one computer sends a request to a program on another computer and awaits a response. The requesting program is called a client; the answering program is called a server.

**client system-options file**

A file, used on AIX, UNIX, or Linux system clients, containing a set of processing options that identify the servers to be contacted for services. This file also specifies communication methods and options for backup, archive, hierarchical storage management, and scheduling. This file is also called the *dsm.sys* file. See also *client user-options file*.

**client user-options file**

A file that contains the set of processing options that the clients on the system use. The set can include options that determine the server that the client contacts, and options that affect backup operations, archive operations, hierarchical storage management operations, and scheduled operations. This file is also called the *dsm.opt* file. For AIX, UNIX, or Linux systems, see also *client system-options file*.

**closed registration**

A registration process in which only an administrator can register workstations as client nodes with the server. Contrast with *open registration*.

**collocation**

The process of keeping all data belonging to a single-client file space, a single client node, or a group of client nodes on a minimal number of sequential-access volumes within a storage pool. Collocation can reduce the number of volumes that must be accessed when a large amount of data must be restored.

**collocation group**

A user-defined group of client nodes whose data is stored on a minimal number of volumes through the process of collocation.

**commit point**

A point in time when data is considered consistent.

**Common Programming Interface for Communications (CPI-C)**

A call-level interface that provides a consistent application programming interface (API) for applications that use program-to-program communications. CPI-C uses LU 6.2 architecture to create a set of interprogram services that can establish and end a conversation, send and receive data, exchange control information, and notify a partner program of errors.

**communication method**

The method by which a client and server exchange information. See also *Transmission Control Protocol/Internet Protocol*.

**communication protocol**

A set of defined interfaces that permit computers to communicate with each other.

**compression**

A function that removes repetitive characters, spaces, or strings of characters from the data being processed and replaces the repetitive characters with control characters. Compression reduces the amount of storage space that is required for the data.

**configuration manager**

A server that distributes configuration information, such as policies and schedules, to managed servers according to their profiles. Configuration information can include policy and schedules. See also *managed server* and *profile*.

**conversation**

A connection between two programs over a session that allows them to communicate with each other while processing a transaction.

**copy backup**

A full backup in which the transaction log files are not deleted so that backup procedures that use incremental or differential backups are not disrupted.

**copy group**

A policy object containing attributes that control how backup versions or archive copies are generated, where backup versions or archive copies are initially

located, and when backup versions or archive copies expire. A copy group belongs to a management class. See also *archive copy group*, *backup copy group*, *backup version*, and *management class*.

**copy storage pool**

A named set of volumes that contain copies of files that reside in primary storage pools. Copy storage pools are used only to back up the data that is stored in primary storage pools. A copy storage pool cannot be a destination for a backup copy group, an archive copy group, or a management class (for space-managed files). See also *primary storage pool* and *destination*.

**CPI-C** See *Common Programming Interface for Communications*.

**D****daemon**

A program that runs unattended to perform continuous or periodic functions, such as network control.

**damaged file**

A physical file in which Tivoli Storage Manager has detected read errors.

**data access control mode**

A mode that controls whether a command can access a migrated file, see a migrated file as zero-length, or receive an input/output error if it attempts to access a migrated file. See also *execution mode*.

**database backup series**

One full backup of the database, plus up to 32 incremental backups made since that full backup. Each full backup that is run starts a new database backup series. A number identifies each backup series.

**database snapshot**

A complete backup of the entire database to media that can be taken off-site. When a database snapshot is created, the current database backup series is not interrupted. A database snapshot cannot have incremental database backups associated with it. See also *database backup series*. Contrast with *full backup*.

**data deduplication**

A method of reducing storage needs by eliminating redundant data. Only one instance of the data is retained on storage

media. Other instances of the same data are replaced with a pointer to the retained instance.

**data manager server**

A server that collects metadata information for client inventory and manages transactions for the storage agent over the local area network. The data manager server informs the storage agent with applicable library attributes and the target volume identifier.

**data mover**

A device that moves data on behalf of the server. A network-attached storage (NAS) file server is a data mover.

**data storage-management application-programming interface (DSMAPI)**

A set of functions and semantics that can monitor events on files, and manage and maintain the data in a file. In an HSM environment, a DSMAPI uses events to notify data management applications about operations on files, stores arbitrary attribute information with a file, supports managed regions in a file, and uses DSMAPI access rights to control access to a file object.

**default management class**

A management class that is assigned to a policy set. This class is used to govern backed up or archived files when a file is not explicitly associated with a specific management class through the include-exclude list.

**deduplication**

See *data deduplication*.

**demand migration**

The process that is used to respond to an out-of-space condition on a file system for which hierarchical storage management (HSM) is active. Files are migrated to server storage until space usage drops to the low threshold that was set for the file system. If the high threshold and low threshold are the same, one file is migrated.

**desktop client**

The group of backup-archive clients that includes clients on Microsoft Windows, Apple, and Novell NetWare operating systems.

**destination**

A copy group or management class attribute that specifies the primary storage pool to which a client file will be backed up, archived, or migrated.

**device class**

A named set of characteristics that are applied to a group of storage devices. Each device class has a unique name and represents a device type of disk, file, optical disk, or tape.

**device configuration file**

(1) For a server, a file that contains information about defined device classes, and, on some servers, defined libraries and drives. The information is a copy of the device configuration information in the database.

(2) For a storage agent, a file that contains the name and password of the storage agent, and information about the server that is managing the SAN-attached libraries and drives that the storage agent uses.

**device driver**

A program that provides an interface between a specific device and the application program that uses the device.

**disaster recovery manager (DRM)**

A function that assists in preparing and using a disaster recovery plan file for the server.

**disaster recovery plan**

A file that is created by the disaster recovery manager (DRM) that contains information about how to recover computer systems if a disaster occurs and scripts that can be run to perform some recovery tasks. The file includes information about the software and hardware that is used by the server, and the location of recovery media.

**domain**

A grouping of client nodes with one or more policy sets, which manage data or storage resources for the client nodes. See *policy domain* or *client domain*.

**DRM** See *disaster recovery manager*.

**DSMAPI**

See *data storage-management application-programming interface*.

**dynamic serialization**

A type of copy serialization in which a file or folder is backed up or archived on the first attempt regardless of whether it changes during a backup or archive.

**E**

**EA** See *extended attribute*.

**EB** See *exabyte*.

**EFS** See *Encrypted File System*.

**Encrypted File System (EFS)**

A file system that uses file system-level encryption.

**enterprise configuration**

A method of setting up servers so that the administrator can distribute the configuration of one of the servers to the other servers, using server-to-server communication. See also *configuration manager*, *managed server*, *profile*, and *subscription*.

**enterprise logging**

The process of sending events from a Tivoli Storage Manager server to a designated event server. The event server routes the events to designated receivers, such as to a user exit. See also *event*.

**error log**

A data set or file that is used to record error information about a product or system.

**estimated capacity**

The available space, in megabytes, of a storage pool.

**event** (1) An administrative command or a client operation that is scheduled to be run using Tivoli Storage Manager scheduling.

(2) A message that an Tivoli Storage Manager server or client issues. Messages can be logged using Tivoli Storage Manager event logging.

**event record**

A database record that describes actual status and results for events.

**event server**

A server to which other servers can send events for logging. The event server routes the events to any receivers that are enabled for the sending server's events.

**exabyte (EB)**

For processor storage, real and virtual storage, and channel volume, 1 152 921 504 606 846 976 bytes. For disk storage capacity and communications volume, 1 000 000 000 000 000 000 bytes.

**exclude**

The process of identifying files in an include-exclude list. This process prevents the files from being backed up or migrated whenever a user or schedule enters an incremental or selective backup operation. A file can be excluded from backup and space management, backup only, or space management only.

**exclude-include list**

See *include-exclude list*.

**execution mode**

A mode that controls the space-management related behavior of commands that run under the **dsmmode** command.

**expiration**

The process by which files, data sets, or objects are identified for deletion because their expiration date or retention period has passed.

**expiring file**

A migrated or premigrated file that has been marked for expiration and removal from storage. If a stub file or an original copy of a premigrated file is deleted from a local file system, or if the original copy of a premigrated file is updated, the corresponding migrated or premigrated file is marked for expiration the next time reconciliation is run.

**extend**

To increase the portion of available space that can be used to store database or recovery log information.

**extended attribute (EA)**

Names or value pairs that are associated with files or directories. There are three classes of extended attributes: user attributes, system attributes, and trusted attributes.

**extent** The part of a file that is created during the data-deduplication process. Extents are compared with other file extents to identify duplicates.

**external library**

A type of library that is provided by Tivoli Storage Manager that permits LAN-free data movement for StorageTek libraries that are managed by Automated Cartridge System Library Software (ACSL). To activate this function, the Tivoli Storage Manager library type must be EXTERNAL.

**F****file access time**

On AIX, UNIX, or Linux systems, the time when the file was last accessed.

**file age**

For migration prioritization purposes, the number of days since a file was last accessed.

**file device type**

A device type that specifies the use of sequential access files on disk storage as volumes.

**file server**

A dedicated computer and its peripheral storage devices that are connected to a local area network that stores programs and files that are shared by users on the network.

**file space**

A logical space in server storage that contains a group of files that have been backed up or archived by a client node, from a single logical partition, file system, or virtual mount point. Client nodes can restore, retrieve, or delete their file spaces from server storage. In server storage, files belonging to a single file space are not necessarily stored together.

**file space ID (FSID)**

A unique numeric identifier that the server assigns to a file space when it is stored in server storage.

**file state**

The space management mode of a file that resides in a file system to which space management has been added. A file can be in one of three states: resident, premigrated, or migrated. See also *resident file*, *premigrated file*, and *migrated file*.

**file system migrator (FSM)**

A kernel extension that intercepts all file system operations and provides any space

management support that is required. If no space management support is required, the operation is passed to the operating system, which performs its normal functions. The file system migrator is mounted over a file system when space management is added to the file system.

**file system state**

The storage management mode of a file system that resides on a workstation on which the hierarchical storage management (HSM) client is installed. A file system can be in one of these states: native, active, inactive, or global inactive.

**frequency**

A copy group attribute that specifies the minimum interval, in days, between incremental backups.

**FSID** See *file space ID*.

**FSM** See *file system migrator*.

**full backup**

The process of backing up the entire server database. A full backup begins a new database backup series. See also *database backup series* and *incremental backup*. Contrast with *database snapshot*.

**fuzzy backup**

A backup version of a file that might not accurately reflect what is currently in the file because the file was backed up at the same time as it was being modified.

**fuzzy copy**

A backup version or archive copy of a file that might not accurately reflect the original contents of the file because it was backed up or archived the file while the file was being modified. See also *backup version* and *archive copy*.

**G****General Parallel File System**

A high-performance shared-disk file system that can provide data access from nodes in a cluster environment.

**gigabyte (GB)**

In decimal notation, 1 073 741 824 when referring to memory capacity; in all other cases, it is defined as 1 000 000 000.

**global inactive state**

The state of all file systems to which

space management has been added when space management is globally deactivated for a client node. When space management is globally deactivated, hierarchical storage management (HSM) cannot perform migration, recall, or reconciliation. However, a root user can update space management settings and add space management to additional file systems. Users can access resident and premigrated files.

**Globally Unique Identifier (GUID)**

An algorithmically determined number that uniquely identifies an entity within a system.

**GPFS** See *General Parallel File System*.

**GPFS node set**

A mounted, defined group of GPFS file systems.

**group backup**

The backup of a group containing a list of files from one or more file space origins.

**GUID** See *Globally Unique Identifier*.

**H**

**hierarchical storage management (HSM)**

A function that automatically distributes and manages data on disk, tape, or both by regarding devices of these types and potentially others as levels in a storage hierarchy that range from fast, expensive devices to slower, cheaper, and possibly removable devices. The objectives are to minimize access time to data and maximize available media capacity.

**hierarchical storage management (HSM) client**

A client program that works with the Tivoli Storage Manager server to provide hierarchical storage management (HSM) for a system. See also *hierarchical storage management* and *space manager client*.

**HSM** See *hierarchical storage management*.

**HSM client**

See *hierarchical storage management client*.

**I**

**ILM** See *information lifecycle management*.

**image** A file system or raw logical volume that is backed up as a single object.

**image backup**

A backup of a full file system or raw logical volume as a single object.

**inactive file system**

A file system for which space management has been deactivated. Contrast with *active file system*.

**inactive version**

A backup version of a file that is either not the most recent backup version, or that is a backup version of a file that no longer exists on the client system. Inactive backup versions are eligible for expiration processing according to the management class assigned to the file. Contrast with *active version*.

**include-exclude file**

A file containing statements to determine the files to back up and the associated management classes to use for backup or archive. See also *include-exclude list*.

**include-exclude list**

A list of options that include or exclude selected files for backup. An exclude option identifies files that should not be backed up. An include option identifies files that are exempt from the exclusion rules or assigns a management class to a file or a group of files for backup or archive services.

**incremental backup**

(1) A copy of all database data that has changed since the most recent successful full backup operation. An incremental backup is also known as a *cumulative backup image* because each incremental backup includes the contents of the previous incremental backup.

(2) The process of backing up information in the database that is new or changed since the last full backup. Contrast with *full backup*. See also *database backup series*.

(3) For Data Protection for Microsoft Exchange Server, a backup in which the transaction logs are backed up and then cleared.

**individual mailbox restore**

See *mailbox restore*.

**information lifecycle management (ILM)**

GPFS policy-based file management for storage pools and file sets.

**i-node** The internal structure that describes the individual files on AIX, UNIX, or Linux systems. An i-node contains the node, type, owner, and location of a file.

**i-node number**  
A number specifying a particular i-node file in the file system.

**IP address**  
A unique address for a device or logical unit on a network that uses the IP standard.

## J

**job file**  
A generated file that contains configuration information for a migration job. The file is XML format and can be created and edited in the hierarchical storage management (HSM) client for Windows client graphical user interface.

**journal-based backup**  
A method for backing up Windows clients and AIX clients that exploits the change notification mechanism in a file to improve incremental backup performance by reducing the need to fully scan the file system.

**journal daemon**  
On AIX, UNIX, or Linux systems, a program that tracks change activity for files residing in file systems.

**journal service**  
In Microsoft Windows, a program that tracks change activity for files residing in file systems.

## K

**kilobyte (KB)**  
For processor storage, real and virtual storage, and channel volume, 210 or 1 024 bytes. For disk storage capacity and communications volume, 1 000 bytes.

## L

**LAN** See *local area network*.

**LAN-free data movement**  
The movement of client data between a client system and a storage device on a storage area network (SAN), bypassing the local area network. This process is also referred to as *LAN-free data transfer*.

**LAN-free data transfer**  
See *LAN-free data movement*.

**leader data**  
Bytes of data, from the beginning of a migrated file, that are stored in the file's corresponding stub file on the local file system. The amount of leader data that is stored in a stub file depends on the stub size that is specified.

**library**  
(1) A repository for demountable recorded media, such as magnetic disks and magnetic tapes.  
(2) A collection of one or more drives, and possibly robotic devices (depending on the library type), which can be used to access storage volumes.

**library client**  
A server that uses server-to-server communication to access a library that is managed by another storage management server. See also *library manager*.

**library manager**  
A server that controls device operations when multiple storage management servers share a storage device. See also *library client*.

**local** Pertaining to a device, file, or system that is accessed directly from a user's system, without the use of a communication line.

**local area network (LAN)**  
A network that connects several devices in a limited area (such as a single building or campus) and that can be connected to a larger network.

**local shadow volumes**  
Data that is stored on shadow volumes localized to a disk storage subsystem.

**LOFS** See *loopback virtual file system*.

**logical file**  
A file that is stored in one or more server storage pools, either by itself or as part of an aggregate. See also *aggregate* and *physical file*.

**logical occupancy**  
The space that is used by logical files in a storage pool. This space does not include the unused space created when logical

files are deleted from aggregate files, so it might be less than the physical occupancy.

**logical unit (LU)**

An access point through which a user or application program accesses the Systems Network Architecture (SNA) network to communicate with another user or application program.

**logical unit number (LUN)**

In the Small Computer System Interface (SCSI) standard, a unique identifier that is used to differentiate devices, each of which is a logical unit (LU).

**logical volume**

A portion of a physical volume that contains a file system.

**logical volume backup**

A back up of a file system or logical volume as a single object.

**Logical Volume Snapshot Agent (LVSA)**

Software that can act as the snapshot provider for creating a snapshot of a logical volume during an online image backup.

**loopback virtual file system (LOFS)**

A file system that is created by mounting a directory over another local directory, also known as mount-over-mount. A LOFS can also be generated using an automounter.

**LU** See *logical unit*.

**LUN** See *logical unit number*.

**LVSA** See *Logical Volume Snapshot Agent*.

**M**

**macro file**

A file that contains one or more storage manager administrative commands, which can be run only from an administrative client using the MACRO command. Contrast with *Tivoli Storage Manager command script*.

**mailbox restore**

A function that restores Microsoft Exchange Server data (from IBM Data Protection for Exchange backups) at the mailbox level or mailbox-item level.

**managed object**

In Tivoli Storage Manager, a definition in

the database of a managed server that was distributed to the managed server by a configuration manager. When a managed server subscribes to a profile, all objects that are associated with that profile become managed objects in the database of the managed server. In general, a managed object cannot be modified locally on the managed server. Objects can include policy, schedules, client option sets, server scripts, administrator registrations, and server and server group definitions.

**managed server**

A Tivoli Storage Manager server that receives configuration information from a configuration manager using a subscription to one or more profiles. Configuration information can include definitions of objects such as policy and schedules. See also *configuration manager*, *subscription*, and *profile*.

**management class**

A policy object that users can bind to each file to specify how the server manages the file. The management class can contain a backup copy group, an archive copy group, and space management attributes. See also *copy group*, *space manager client*, *bind*, and *rebind*.

**maximum transmission unit**

The largest possible unit of data that can be sent on a given physical medium in a single frame. For example, the maximum transmission unit for Ethernet is 1500 bytes.

**MB** See *megabyte*.

**megabyte (MB)**

(1) 1 048 576 bytes (two to the twentieth power) when used in this publication.

(2) For processor storage, real and virtual storage, and channel volume, 2 to the power of 20 or 1 048 576 bits. For disk storage capacity and communications volume, 1 000 000 bits.

**metadata**

Data that describes the characteristics of data; descriptive data.

**migrate**

To move data from one storage location to another. In Tivoli Storage Manager products, migrating can mean moving

data from a client node to server storage, or moving data from one storage pool to the next storage pool defined in the server storage hierarchy. In both cases the movement is controlled by policy, such as thresholds that are set. See also *migration threshold*.

**migrated file**

A file that has been copied from a local file system to Tivoli Storage Manager storage. For HSM clients on UNIX or Linux systems, the file is replaced with a stub file on the local file system. On Windows systems, creation of the stub file is optional. See also *stub file* and *resident file*. For HSM clients on UNIX or Linux systems, contrast with *premigrated file*.

**migrate-on-close recall mode**

A mode that causes a migrated file to be recalled back to its originating file system temporarily. Contrast with *normal recall mode* and *read-without-recall recall mode*.

**migration job**

A specification of files to migrate, and actions to perform on the original files after migration. See also *job file*.

**migration threshold**

High and low capacities for storage pools or file systems, expressed as percentages, at which migration is set to start and stop.

**mirroring**

The process of writing the same data to multiple locations at the same time. Mirroring data protects against data loss within the recovery log.

**mode** A copy group attribute that specifies whether to back up a file that has not been modified since the last time the file was backed up. See *modified mode* and *absolute mode*.

**modified mode**

In storage management, a backup copy-group mode that specifies that a file is considered for incremental backup only if it has changed since the last backup. A file is considered a changed file if the date, size, owner, or permissions of the file have changed. See also *absolute mode*.

**mount limit**

The maximum number of volumes that can be simultaneously accessed from the

same device class. The mount limit determines the maximum number of mount points. See also *mount point*.

**mount point**

On the Tivoli Storage Manager server, a logical drive through which volumes in a sequential access device class are accessed. For removable-media device types, such as tape, a mount point is a logical drive that is associated with a physical drive. For the file device type, a mount point is a logical drive that is associated with an I/O stream. The number of mount points for a device class is defined by the value of the mount limit attribute for that device class. See also *mount limit*.

**mount retention period**

The maximum number of minutes that the server retains a mounted sequential-access media volume that is not being used before it dismounts the sequential-access media volume.

**mount wait period**

The maximum number of minutes that the server waits for a sequential-access volume mount request to be satisfied before canceling the request.

**MTU** See *maximum transmission unit*.

**N****Nagle algorithm**

An algorithm that reduces congestion of TCP/IP networks by combining smaller packets and sending them together.

**named pipe**

A type of interprocess communication that permits message data streams to pass between peer processes, such as between a client and a server.

**NAS** See *network-attached storage*.

**NAS node**

A client node that is a network-attached storage (NAS) file server. Data for the NAS node is transferred by a NAS file server that is controlled by the network data management protocol (NDMP). A NAS node is also called a NAS file server node.

**native file system**

A file system that is locally added to the file server and is not added for space

management. The hierarchical storage manager (HSM) client does not provide space management services to the file system.

**native format**

A format of data that is written to a storage pool directly by the Tivoli Storage Manager server. Contrast with *non-native data format*.

**NDMP**

See *Network Data Management Protocol*.

**NetBIOS**

See *Network Basic Input/Output System*.

**network-attached storage (NAS) file server**

A dedicated storage device with an operating system that is optimized for file-serving functions. A NAS file server can have the characteristics of both a node and a data mover.

**Network Basic Input/Output System (NetBIOS)**

A standard interface to networks and personal computers that is used on local area networks to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not have to handle the details of LAN data link control (DLC) protocols.

**Network Data Management Protocol (NDMP)**

A protocol that allows a network storage-management application to control the backup and recovery of an NDMP-compliant file server, without installing vendor-acquired software on that file server.

**network data-transfer rate**

A rate that is calculated by dividing the total number of bytes that are transferred by the data transfer time. For example, this rate can be the time that is spent transferring data over a network.

**node** A file server or workstation on which the backup-archive client program has been installed, and which has been registered to the server.

**node name**

A unique name that is used to identify a workstation, file server, or PC to the server.

**node privilege class**

A privilege class that gives an administrator the authority to remotely

access backup-archive clients for a specific client node or for all clients in a policy domain. See also *privilege class*.

**non-native data format**

A format of data that is written to a storage pool that differs from the format that the server uses for operations.

**normal recall mode**

A mode that causes a migrated file to be copied back to its originating file system when it is accessed.

**O**

**offline volume backup**

A backup in which the volume is locked so that no other system applications can access it during the backup operation.

**online volume backup**

A backup in which the volume is available to other system applications during the backup operation.

**open registration**

A registration process in which users can register their workstations as client nodes with the server. Contrast with *closed registration*.

**operator privilege class**

A privilege class that gives an administrator the authority to disable or halt the server, enable the server, cancel server processes, and manage removable media. See also *privilege class*.

**options file**

A file that contains processing options. On Windows and NetWare systems, the file is called *dsm.opt*. On AIX, UNIX, Linux, and Mac OS X systems, the file is called *dsm.sys*.

**originating file system**

The file system from which a file was migrated. When a file is recalled using normal or migrate-on-close recall mode, it is always returned to its originating file system.

**orphaned stub file**

A file for which no migrated file can be found on the Tivoli Storage Manager server that the client node is contacting for space management services. For example, a stub file can be orphaned when the client system-options file is

modified to contact a server that is different than the one to which the file was migrated.

**out-of-space protection mode**

A mode that controls whether the program intercepts out-of-space conditions. See also *execution mode*.

**P**

**pacing**

In SNA, a technique by which the receiving system controls the rate of transmission of the sending system to prevent overrun.

**packet** In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole.

**page** A defined unit of space on a storage medium or within a database volume.

**partial-file recall mode**

A recall mode that causes the hierarchical storage management (HSM) function to read just a portion of a migrated file from storage, as requested by the application accessing the file.

**password generation**

A process that creates and stores a new password in an encrypted password file when the old password expires. Automatic generation of a password prevents password prompting. Password generation can be set in the options file (passwordaccess option). See also *options file*.

**path** An object that defines a one-to-one relationship between a source and a destination. Using the path, the source accesses the destination. Data can flow from the source to the destination, and back. An example of a source is a data mover (such as a network-attached storage [NAS] file server), and an example of a destination is a tape drive.

**pattern-matching character**

See *wildcard character*.

**physical file**

A file that is stored in one or more storage pools, consisting of either a single logical file, or a group of logical files that are packaged together as an aggregate. See also *aggregate* and *logical file*.

**physical occupancy**

The amount of space that is used by physical files in a storage pool. This space includes the unused space that is created when logical files are deleted from aggregates. See also *physical file*, *logical file*, and *logical occupancy*.

**plug-in**

A self-contained software component that modifies (adds, or changes) the function in a particular system. When a plug-in is added to a system, the foundation of the original system remains intact.

**policy domain**

A grouping of policy users with one or more policy sets, which manage data or storage resources for the users. The users are client nodes that are associated with the policy domain.

**policy privilege class**

A privilege class that gives an administrator the authority to manage policy objects, register client nodes, and schedule client operations for client nodes. Authority can be restricted to certain policy domains. See also *privilege class*.

**policy set**

A group of rules in a policy domain. The rules specify how data or storage resources are automatically managed for client nodes in the policy domain. Rules can be contained in management classes. See also *active policy set* and *management class*.

**premigrated file**

A file that has been copied to Tivoli Storage Manager storage, but has not been replaced with a stub file on the local file system. An identical copy of the file resides both on the local file system and in Tivoli Storage Manager storage. Premigrated files occur on UNIX and Linux file systems to which space management has been added. Contrast with *migrated file* and *resident file*.

**premigrated files database**

A database that contains information about each file that has been premigrated to Tivoli Storage Manager storage. The database is stored in a hidden directory

named `.SpaceMan` in each file system to which space management has been added.

### **premigration**

The process of copying files that are eligible for migration to Tivoli Storage Manager storage, but leaving the original file intact on the local file system.

### **premigration percentage**

A space management setting that controls whether the next eligible candidates in a file system are premigrated following threshold or demand migration.

### **primary storage pool**

A named set of volumes that the server uses to store backup versions of files, archive copies of files, and files migrated from client nodes. See also *destination* and *copy storage pool*.

### **privilege class**

A level of authority that is granted to an administrator. The privilege class determines which administrative tasks the administrator can perform. See also *node privilege class*, *operator privilege class*, *policy privilege class*, *storage privilege class*, and *system privilege class*.

### **profile**

A named group of configuration information that can be distributed from a configuration manager when a managed server subscribes. Configuration information can include registered administrator IDs, policies, client schedules, client option sets, administrative schedules, storage manager command scripts, server definitions, and server group definitions. See also *configuration manager* and *managed server*.

## **Q**

**quota** (1) For HSM on AIX, UNIX, or Linux systems, the limit (in megabytes) on the amount of data that can be migrated and premigrated from a file system to server storage.

(2) For HSM on Windows systems, a user-defined limit to the space that is occupied by recalled files.

## **R**

### **randomization**

The process of distributing schedule start

times for different clients within a specified percentage of the schedule's startup window.

### **raw logical volume**

A portion of a physical volume that is comprised of unallocated blocks and has no journaled file system (JFS) definition. A logical volume is read/write accessible only through low-level I/O functions.

### **read-without-recall recall mode**

A mode that causes hierarchical storage management (HSM) to read a migrated file from storage without storing it back on the local file system. The last piece of information read from the file is stored in a buffer in memory on the local file system. Contrast with *normal recall mode* and *migrate-on-close recall mode*.

### **rebind**

To associate a backed-up file with a new management class name. For example, rebinding occurs when the management class associated with a file is deleted. See also *bind*.

**recall** In Tivoli Storage Manager, to copy a migrated file from server storage back to its originating file system using the space management client. See also *transparent recall*, *selective recall*, and *recall mode*.

### **recall mode**

A mode that is assigned to a migrated file with the `dsmattr` command that determines how the file is processed when it is recalled. It determines whether the file is stored on the local file system, is migrated back to Tivoli Storage Manager storage when it is closed, or is read from Tivoli Storage Manager storage without storing it on the local file system.

### **receiver**

A server repository that contains a log of server and client messages as events. For example, a receiver can be a file exit, a user exit, or the Tivoli Storage Manager server console and activity log. See also *event*.

### **reclamation**

The process of consolidating the remaining data from many sequential-access volumes onto fewer, new sequential-access volumes.

**reclamation threshold**

The percentage of space that a sequential-access media volume must have before the server can reclaim the volume. Space becomes reclaimable when files are expired or are deleted.

**reconciliation**

The process of synchronizing a file system with the Tivoli Storage Manager server, and then removing old and obsolete objects from the Tivoli Storage Manager server.

**recovery log**

A log of updates that are about to be written to the database. The log can be used to recover from system and media failures. The recovery log consists of the active log (including the log mirror) and archive logs.

**register**

To define a client node or administrator ID that can access the server.

**registry**

A repository that contains access and configuration information for users, systems, and software.

**resident file**

On a Windows system, a complete file on a local file system that might also be a migrated file because a migrated copy can exist in Tivoli Storage Manager storage. On a UNIX or Linux system, a complete file on a local file system that has not been migrated or premigrated, or that has been recalled from Tivoli Storage Manager storage and modified. Contrast with *stub file* and *premigrated file*. See *migrated file*.

**restore**

To copy information from its backup location to the active storage location for use. For example, to copy information from server storage to a client workstation.

**retention**

The amount of time, in days, that inactive backed-up or archived files are kept in the storage pool before they are deleted. Copy group attributes and default retention grace periods for the domain define retention.

**retrieve**

To copy archived information from the

storage pool to the workstation for use. The retrieve operation does not affect the archive version in the storage pool.

**roll back**

To remove changes that were made to database files since the last commit point.

**root user**

A system user who operates without restrictions. A root user has the special rights and privileges needed to perform administrative tasks.

**S**

**SAN** See *storage area network*.

**schedule**

A database record that describes client operations or administrative commands to be processed. See *administrative command schedule* and *client schedule*.

**scheduling mode**

The type of scheduling operation for the server and client node that supports two scheduling modes: client-polling and server-prompted.

**scratch volume**

A labeled volume that is either blank or contains no valid data, that is not defined, and that is available for use.

**script** A series of commands, combined in a file, that carry out a particular function when the file is run. Scripts are interpreted as they are run. Contrast with *Tivoli Storage Manager command script*.

**Secure Sockets Layer (SSL)**

A security protocol that provides communication privacy. With SSL, client/server applications can communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

**selective backup**

The process of backing up certain files or directories from a client domain. The files that are backed up are those that are not excluded in the include-exclude list. The files must meet the requirement for serialization in the backup copy group of the management class that is assigned to each file. Contrast with *incremental backup*.

**selective migration**

The process of copying user-selected files

- from a local file system to Tivoli Storage Manager storage and replacing the files with stub files on the local file system. Contrast with *threshold migration* and *demand migration*.
- selective recall**  
The process of copying user-selected files from Tivoli Storage Manager storage to a local file system. Contrast with *transparent recall*.
- serialization**  
The process of handling files that are modified during backup or archive processing. See *dynamic serialization*, *static serialization*, *shared static serialization*, and *shared dynamic serialization*.
- server** A software program or a computer that provides services to other software programs or other computers.
- server options file**  
A file that contains settings that control various server operations. These settings affect such things as communications, devices, and performance.
- server-prompted scheduling mode**  
A client/server communication technique where the server contacts the client node when tasks must be done. Contrast with *client-polling scheduling mode*.
- server storage**  
The primary, copy, and active-data storage pools that are used by the server to store user files such as backup versions, archive copies, and files migrated from space manager client nodes (space-managed files). See also *active-data pool*, *primary storage pool*, *copy storage pool*, *storage pool volume*, and *volume*.
- session**  
A logical or virtual connection between two stations, software programs, or devices on a network that allows the two elements to communicate and exchange data.
- session resource usage**  
The amount of wait time, processor time, and space that is used or retrieved during a client session.
- shared dynamic serialization**  
A value for serialization that specifies that a file must not be backed up or archived if it is being modified during the operation. Tivoli Storage Manager retries the backup or archive operation a number of times; if the file is being modified during each attempt, Tivoli Storage Manager will back up or archive the file on its last try. See also *serialization*. Contrast with *dynamic serialization*, *shared static serialization*, and *static serialization*.
- shared library**  
A library device that is used by multiple storage manager servers.
- shared static serialization**  
A copy-group serialization value that specifies that a file must not be modified during a backup or archive operation. Tivoli Storage Manager attempts to retry the operation a number of times. If the file is in use during each attempt, the file is not backed up or archived. See also *serialization*. Contrast with *dynamic serialization*, *shared dynamic serialization*, and *static serialization*.
- snapshot**  
An image backup type that consists of a point-in-time view of a volume.
- space-managed file**  
A file that is migrated from a client node by the space manager client. The space manager client recalls the file to the client node on demand.
- space management**  
The process of keeping sufficient free storage space available on a local file system for new data by migrating files to server storage. Synonymous with *hierarchical storage management*.
- space manager client**  
A program that runs on a UNIX or Linux system to manage free space on the local file system by migrating files to server storage. The program can recall the files either automatically or selectively. Also called *hierarchical storage management (HSM) client*.
- space monitor daemon**  
A daemon that checks space usage on all file systems for which space management is active, and automatically starts threshold migration when space usage on a file system equals or exceeds its high threshold.

**sparse file**

A file that is created with a length greater than the data it contains, leaving empty spaces for the future addition of data.

**special file**

On AIX, UNIX, or Linux systems, a file that defines devices for the system, or temporary files that are created by processes. There are three basic types of special files: first-in, first-out (FIFO); block; and character.

**SSL** See *Secure Sockets Layer*.

**stabilized file space**

A file space that exists on the server but not on the client.

**stanza** A group of lines in a file that together have a common function or define a part of the system. Each stanza is identified by a name that occurs in the first line of the stanza. Depending on the type of file, a stanza is ended by the next occurrence of a stanza name in the file, or by an explicit end-of-stanza marker. A stanza can also be ended by the end of the file.

**startup window**

A time period during which a schedule must be initiated.

**static serialization**

A copy-group serialization value that specifies that a file must not be modified during a backup or archive operation. If the file is in use during the first attempt, the storage manager cannot back up or archive the file. See also *serialization*. Contrast with *dynamic serialization*, *shared dynamic serialization*, and *shared static serialization*.

**storage agent**

A program that enables the backup and restoration of client data directly to and from storage attached to a storage area network (SAN).

**storage area network (SAN)**

A dedicated storage network that is tailored to a specific environment, combining servers, systems, storage products, networking products, software, and services.

**storage hierarchy**

(1) A logical order of primary storage pools, as defined by an administrator. The

order is typically based on the speed and capacity of the devices that the storage pools use. The storage hierarchy is defined by identifying the next storage pool in a storage pool definition. See also *storage pool*.

(2) An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include: main storage, such as memory and direct-access storage device (DASD) cache; primary storage (DASD containing user-accessible data); migration level 1 (DASD containing data in a space-saving format); and migration level 2 (tape cartridges containing data in a space-saving format).

**storage pool**

A named set of storage volumes that are the destination that is used to store client data. A storage pool contains backup versions, archive copies, and files that are migrated from space manager client nodes. A primary storage pool is backed up to a copy storage pool. See also *primary storage pool*, *copy storage pool*, and *active-data pool*.

**storage pool volume**

A volume that has been assigned to a storage pool. See also *volume*, *active-data pool*, *copy storage pool*, and *primary storage pool*.

**storage privilege class**

A privilege class that gives an administrator the authority to control how storage resources for the server are allocated and used, such as monitoring the database, the recovery log, and server storage. See also *privilege class*.

**stub**

A shortcut on the Windows file system that is generated by the hierarchical storage management (HSM) client for a migrated file that allows transparent user access. A stub is the sparse file representation of a migrated file, with a reparse point attached.

**stub file**

A file that replaces the original file on a local file system when the file is migrated to storage. A stub file contains the information that is necessary to recall a migrated file from Tivoli Storage Manager storage. It also contains additional

information that can be used to eliminate the need to recall a migrated file.

**stub file size**

The size of a file that replaces the original file on a local file system when the file is migrated to Tivoli Storage Manager storage. The size that is specified for stub files determines how much leader data can be stored in the stub file. The default for stub file size is the block size defined for a file system minus 1 byte.

**subscription**

In a Tivoli environment, the process of identifying the subscribers that the profiles are distributed to. For Tivoli Storage Manager, a subscription is the process by which a managed server receives configuration information associated with a particular profile on a configuration manager. See also *managed server*, *configuration manager*, and *profile*.

**system privilege class**

A privilege class that gives an administrator the authority to issue all server commands. See also *privilege class*.

**Systems Network Architecture (SNA)**

The description of the logical structure, formats, protocols, and operational sequences for transmitting information through and controlling the configuration and operation of networks.

**T**

**tape library**

A set of equipment and facilities that support an installation's tape environment. The tape library can include tape storage racks, mechanisms for automatic tape mounting, a set of tape drives, and a set of related tape volumes mounted on those drives.

**tape volume prefix**

The high-level-qualifier of the file name or the data set name in the standard tape label.

**target node**

A client node for which other client nodes (called agent nodes) have been granted proxy authority. The proxy authority allows the agent nodes to perform operations such as backup and restore on behalf of the target node, which owns the data.

**TCA** See *trusted communications agent*.

**TCP/IP**

See *Transmission Control Protocol/Internet Protocol*.

**threshold migration**

The process of moving files from a local file system to Tivoli Storage Manager storage based on the high and low thresholds that are defined for the file system. Contrast with *demand migration*, *selective migration*, and *migration job*.

**throughput**

In storage management, the total bytes in the workload, excluding overhead, that are backed up or restored, divided by elapsed time.

**timeout**

A time interval that is allotted for an event to occur or complete before operation is interrupted.

**timestamp control mode**

A mode that determines whether commands preserve the access time for a file or set it to the current time.

**Tivoli Storage Manager command script**

A sequence of Tivoli Storage Manager administrative commands that are stored in the database of the Tivoli Storage Manager server. The script can run from any interface to the server. The script can include substitution for command parameters and conditional logic.

**tombstone object**

A small subset of attributes of a deleted object. The tombstone object is retained for a specified period, and at the end of the specified period, the tombstone object is permanently deleted.

**Transmission Control Protocol/Internet Protocol (TCP/IP)**

An industry-standard, nonproprietary set of communication protocols that provides reliable end-to-end connections between applications over interconnected networks of different types.

**transparent recall**

The process that is used to automatically recall a file to a workstation or file server when the file is accessed. See also *recall mode*. Contrast with *selective recall*.

**trusted communications agent (TCA)**

A program that handles the sign-on password protocol when clients use password generation.

**U**

**UCS-2** A 2-byte (16-bit) encoding scheme based on ISO/IEC specification 10646-1. UCS-2 defines three levels of implementation: Level 1-No combining of encoded elements allowed; Level 2-Combining of encoded elements is allowed only for Thai, Indic, Hebrew, and Arabic; Level 3-Any combination of encoded elements are allowed.

**UNC** See *Universal Naming Convention name*.

**Unicode**

A character encoding standard that supports the interchange, processing, and display of text that is written in the common languages around the world, plus some classical and historical texts. The Unicode standard has a 16-bit character set defined by ISO 10646.

**Unicode-enabled file space**

Unicode file space names provide support for multilingual workstations without regard for the current locale.

**Unicode transformation format 8**

Unicode Transformation Format (UTF), 8-bit encoding form, which is designed for ease of use with existing ASCII-based systems. The CCSID value for data in UTF-8 format is 1208.

**Universal Naming Convention (UNC) name**

A name that is used to access a drive or directory containing files shared across a network. The UNC name includes the system name and a SharePoint name that represents the shared drive or directory.

**Universally Unique Identifier (UUID)**

The 128-bit numerical identifier that is used to ensure that two components do not have the same identifier.

**UTF-8** See *Unicode transformation format 8*.

**UUID** See *Universally Unique Identifier*.

**V****validate**

To check a policy set for conditions that can cause problems if that policy set becomes the active policy set. For

example, the validation process checks whether the policy set contains a default management class.

**version**

A backup copy of a file stored in server storage. The most recent backup copy of a file is the active version. Earlier copies of the same file are inactive versions. The number of versions retained by the server is determined by the copy group attributes in the management class.

**virtual file space**

A representation of a directory on a network-attached storage (NAS) file system as a path to that directory.

**virtual volume**

An archive file on a target server that represents a sequential media volume to a source server.

**volume**

A discrete unit of storage on disk, tape or other data recording medium that supports some form of identifier and parameter list, such as a volume label or input/output control. See also *scratch volume*, and *storage pool volume*.

**volume history file**

A file that contains information about volumes that have been used by the server for database backups and for export of administrator, node, policy, or server data. The file also has information about sequential-access storage pool volumes that have been added, reused, or deleted. The information is a copy of volume information that is recorded in the server database.

**Volume Shadow Copy Service**

A set of Microsoft application-programming interfaces (APIs) that you can use to create shadow copy backups of volumes, exact copies of files, including all open files, and so on.

**VSS** See *Volume Shadow Copy Service*.

**VSS Backup**

A backup operation that uses Microsoft Volume Shadow Copy Service (VSS) technology. The backup operation produces an online snapshot (point-in-time consistent copy) of Exchange data. This copy can be stored

on local shadow volumes or on Tivoli Storage Manager server storage.

**VSS Fast Restore**

A function that uses a Microsoft Volume Shadow Copy Service (VSS) software provider to restore VSS Backups (IBM Data Protection for Exchange database files and log files) that reside on local shadow volumes.

**VSS Instant Restore**

A volume-level hardware-assisted Microsoft Volume Shadow Copy Service (VSS) function where target volumes that contain the snapshot are copied back to the original source volumes.

**VSS offloaded backup**

A backup operation that uses a Microsoft Volume Shadow Copy Service (VSS) hardware provider (installed on an alternate system) to move IBM Data Protection for Exchange data to the Tivoli Storage Manager server. This type of backup operation shifts the backup load from the production system to another system.

**VSS Restore**

A function that uses a Microsoft Volume Shadow Copy Service (VSS) software provider to restore VSS Backups (IBM Data Protection for Exchange database files and log files) that reside on Tivoli Storage Manager server storage to their original location.

**W**

**wildcard character**

A special character such as an asterisk (\*) or a question mark (?) that can be used to represent one or more characters. Any character or set of characters can replace the wildcard character.

**workstation**

A configuration of input/output equipment at which an operator works. A workstation is a terminal or microcomputer at which a user can run applications and that is usually connected to a mainframe or a network.

**worldwide name**

A 64-bit, unsigned name identifier that is unique.

**workload partition (WPAR)**

A partition within a single operating system instance.



---

# Index

## Special characters

\$\$CONFIG\_MANAGER\$\$ 702

## Numerics

3480 tape drive  
  cleaner cartridge 162  
  device support 42  
  mixing drive generations 204  
3490 tape drive  
  cleaner cartridge 162  
  device support 42  
  mixing drive generations 204  
3494 automated library device 44, 99  
3494 library  
  configuration with a single drive device 101  
3494SHARED server option 66  
3570 tape drive  
  defining device class 63, 199  
  device support 42  
3590 tape drive  
  ASSISTVCRRECOVERY server option 67  
  defining device class 63, 199, 200  
3592 drives and media  
  as element of storage hierarchy 241  
  cleaning 160  
  data encryption 158, 206, 496  
  defining device class 63  
  DEVICETYPE parameter 138  
  enabling for WORM media 141  
  mixing drive generations 204  
4mm tape device support 200  
8mm tape device support 200

## A

absolute mode, description of 463  
ACCEPT DATE command 558  
access authority, client 394, 395  
access mode, volume  
  changing 238  
  description 239  
  determining for storage pool 227, 358  
access, managing 421, 427  
accessibility features 877  
accounting record  
  description of 645  
  monitoring 645  
accounting variable 645  
ACSL (Automated Cartridge System Library Software)  
  StorageTek library 43  
  configuring 112  
  description 44  
  mixing 3592 drive generations 204  
  Tivoli Storage Manager server options for 66  
ACTIVATE POLICYSET command 470  
active data, protecting with active-data pools 223  
active files, storage-pool search order 225  
active log  
  description 596  
  active log (*continued*)  
    increasing the size 605  
    mirroring 759  
    move to another directory 609  
    out of space 605  
    size of 767  
    when to backup 759  
  active log mirror  
    description 596, 601  
  active log space 601  
  ACTIVE policy set  
    creating 460, 470  
    replacing 439  
  active-data pool  
    auditing volumes in 784  
    backup-set file source 502  
    client restore operations, optimizing 514  
    collocation on 316  
    defining 357  
    export file source 727, 735, 736  
    import operations 745  
    overview 223, 246, 756  
    reclamation of 325  
    RECONSTRUCT parameter 515  
    restoring  
      storage pools 757, 777  
      volumes 757, 780  
    simultaneous-write function 285  
    specifying in policy definition 458  
    storage pool search-and-selection order 225  
  ACTIVELOGDIR server option 605, 609  
  ACTIVELOGSIZE server option 605  
  activity log  
    description of 623  
    logging events to 628  
    monitoring 623  
    querying 624  
    setting size limit 625  
    setting the retention period 625  
  administration center  
    deploying backup-archive packages automatically 380  
    using 547  
  Administration center  
    managing servers 547  
  Administration Center  
    backing up 553  
    commands not supported 550  
    features not supported 550  
    protecting 552  
    restoring 553  
    starting and stopping 550  
  Administration Center scripts 621  
  administrative client  
    description of 3  
    viewing information after IMPORT or EXPORT 750  
  administrative clients  
    preventing from accessing the server 724  
  administrative commands  
    ACCEPT DATE 563  
    ASSIGN DEFMGMTCLASS 470  
    AUDIT LIBVOLUME 149

administrative commands (continued)

AUDIT LICENSE 557  
 AUDIT VOLUME 788  
 BACKUP DB 606, 607, 771  
 BACKUP DEVCONFIG 770  
 BACKUP NODE 188, 189  
 BACKUP STGPOOL 760  
 BACKUP VOLHISTORY 768  
 BEGIN EVENTLOGGING 628  
 CANCEL PROCESS 565  
 CANCEL RESTORE 218  
 CANCEL SESSION 415  
 CHECKIN LIBVOLUME 137  
 CHECKOUT LIBVOLUME 148  
 CLEAN DRIVE 159  
 COMMIT 590  
 COPY ACTIVATEDATA 225, 246  
 COPY CLOPTSET 413  
 COPY DOMAIN 460  
 COPY POLICYSET 460  
 COPY SCHEDULE 532  
 COPY SCRIPT 585  
 COPY SERVERGROUP 711  
 DEFINE ASSOCIATION 32, 527  
 DEFINE BACKUPSET 508  
 DEFINE CLIENTACTION 543  
 DEFINE CLIENTOPT 543  
 DEFINE CLOPTSET 411  
 DEFINE COPYGROUP 462, 467, 468  
 DEFINE DATAMOVER 128, 186  
 DEFINE DEVCLASS  
   3592 204  
   FILE device classes 207  
   LTO device classes 211  
   REMOVEABLEFILE device classes 207  
   SERVER device classes 214  
   tape device classes 200  
   VOLSAFE device classes 215  
 DEFINE DRIVE 127  
 DEFINE GRPMEMBER 710  
 DEFINE LIBRARY 44, 126  
 DEFINE MACHINE 808  
 DEFINE MACHNODEASSOCIATION 808  
 DEFINE PATH 129, 187  
 DEFINE POLICYSET 460, 461  
 DEFINE PROFASSOCIATION 690, 691  
 DEFINE PROFILE 690  
 DEFINE RECMEDMACHASSOCIATION 810  
 DEFINE RECOVERYMEDIA 810  
 DEFINE SCHEDULE 573, 606  
 DEFINE SCRIPT 578  
 DEFINE SERVER 678, 707, 714  
 DEFINE SERVERGROUP 710  
 DEFINE STGPOOL 231, 232, 242, 243  
 DEFINE SUBSCRIPTION 701  
 DEFINE VIRTUALFSMAPPING 195  
 DEFINE VOLUME 50, 237  
 DELETE ASSOCIATION 534  
 DELETE BACKUPSET 511  
 DELETE COPYGROUP 492  
 DELETE DOMAIN 493  
 DELETE DRIVE 163  
 DELETE EVENT 536  
 DELETE GRPMEMBER 712  
 DELETE KEYRING 424  
 DELETE LIBRARY 156  
 DELETE MGMTCLASS 493

administrative commands (continued)

DELETE POLICYSET 493  
 DELETE PROFASSOCIATION 695  
 DELETE PROFILE 697  
 DELETE SCHEDULE 532  
 DELETE SCRIPT 586  
 DELETE SERVER 684  
 DELETE SERVERGROUP 711  
 DELETE STGPOOL 361  
 DELETE SUBSCRIBER 706  
 DELETE SUBSCRIPTION 697, 702  
 DELETE VOLHISTORY 768, 769  
 DELETE VOLUME 362, 363  
 DISABLE EVENTS 627  
 DISABLE SESSIONS 417  
 DISMOUNT VOLUME 154  
 DSMSERV DISPLAY DBSPACE 603  
 DSMSERV DISPLAY LOG 603  
 DSMSERV FORMAT 774  
 ENABLE EVENTS 627  
 ENABLE SESSIONS 417  
 END EVENTLOGGING 628  
 EXPIRE INVENTORY 34  
 EXPORT ADMIN 721  
 EXPORT NODE 734  
 EXPORT POLICY 734  
 EXPORT SERVER 734  
 EXTEND DBSPACE 604  
 GENERATE BACKUPSET 503  
 GRANT AUTHORITY 425  
 HALT 561  
 HELP 569  
 IMPORT 748, 750  
 IMPORT ADMIN 737  
 IMPORT NODE 737, 746  
 IMPORT POLICY 737  
 IMPORT SERVER 737, 746  
 LABEL LIBVOLUME 94, 123  
 LOCK ADMIN 430  
 LOCK NODE 388  
 LOCK PROFILE 694, 695  
 MOVE DATA 350  
 MOVE NODEDATA 354  
 NOTIFY SUBSCRIBERS 695  
 PING SERVER 712  
 PREPARE 810  
 QUERY ACTLOG 624  
 QUERY BACKUPSETCONTENTS 510  
 QUERY CONTENT 338  
 QUERY COPYGROUP 490, 744  
 QUERY DB 603  
 QUERY DBSPACE 603  
 QUERY DEVCLASS 216  
 QUERY DOMAIN 491  
 QUERY DRIVE 156  
 QUERY DRMSTATUS 801  
 QUERY ENABLED 640  
 QUERY EVENT 528  
 QUERY FILESPACE 409  
 QUERY LIBRARY 155  
 QUERY LICENSE 557  
 QUERY MGMTCLASS 490  
 QUERY MOUNT 154  
 QUERY NODE 392  
 QUERY NODEDATA 347  
 QUERY OCCUPANCY  
   backed-up, archived, and space-managed files 348

administrative commands (continued)

QUERY OCCUPANCY (continued)  
  client file spaces 346  
  client nodes 346  
  device classes 347  
  storage pools 347  
QUERY OPTION 616  
QUERY POLICYSET 491  
QUERY PROCESS 352  
QUERY REQUEST 152  
QUERY RESTORE 418  
QUERY RPFCONTENT 812  
QUERY RPFFILE 812  
QUERY SCHEDULE 528  
QUERY SCRIPT 585  
QUERY SERVERGROUP 711  
QUERY STGPOOL 332, 342, 740  
QUERY SUBSCRIPTION 701  
QUERY SYSTEM 617  
QUERY VOLHISTORY 768  
QUERY VOLUME 335, 353  
RECONCILE VOLUMES 718  
REGISTER ADMIN 428  
REGISTER LICENSE 556  
REMOVE ADMIN 429  
REMOVE NODE 389  
RENAME ADMIN 429  
RENAME FILESPACE 748  
RENAME NODE 388  
RENAME SCRIPT 586  
RENAME SERVERGROUP 711  
RENAME STGPOOL 357  
RESTORE DB 562  
RESTORE NODE 188, 189  
RESTORE STGPOOL 763, 794  
RESTORE VOLUME 796  
ROLLBACK 591  
RUN 586  
SELECT 618  
SET ACCOUNTING 645  
SET AUTHENTICATION 433  
SET CLIENTACTDURATION 543  
SET CONFIGMANAGER 687, 689  
SET CONFIGREFRESH 701  
SET CONTEXTMESSAGING 628  
SET CROSSDEFINE 679, 682  
SET DBRECOVERY 606  
SET DBREPORTMODE 603  
SET DRMCHECKLABEL 806  
SET DRMCOPYSTGPOOL 802  
SET DRMCOURIERNAME 805  
SET DRMDBBACKUPRXPIREDAYS 806  
SET DRMFILEPROCESS 806  
SET DRMINSTPREFIX 804  
SET DRMNOTMOUNTABLE 805  
SET DRMPPLANPOSTFIX 803  
SET DRMPPLANPREFIX 804  
SET DRMPRIMSTGPOOL 802  
SET DRMRPFEXPIREDAYS 812  
SET DRMVaultNAME 806  
SET EVENTRETENTION 536, 577  
SET INVALIDPWLIMIT 433  
SET LICENSEAUDITPERIOD 557  
SET MAXCMDRETRIES 542  
SET MAXSCHEDULESESSIONS 540  
SET MINPWLENGTH 433  
SET PASSEXP 432

administrative commands (continued)

SET QUERYSCHEDPERIOD 542  
SET RANDOMIZE 540  
SET REGISTRATION 368  
SET RETRYPERIOD 543  
SET SCHEDMODES 538  
SET SERVERHLADDRESS 679, 682  
SET SERVERLLADDRESS 679, 682  
SET SERVERNAME 616, 678, 682  
SET SERVERPASSWORD 678, 679, 682  
SET SUBFILE 511  
SET SUMMARYRETENTION 622  
SETOPT 569  
UNLOCK PROFILE 694, 695  
UPDATE ADMIN 428  
UPDATE ARCHIVE 522  
UPDATE BACKUPSET 510  
UPDATE CLIENTOPT 413  
UPDATE CLOPTSET 413  
UPDATE COPYGROUP 462, 468  
UPDATE DEVCLASS 200  
UPDATE DOMAIN 460  
UPDATE DRIVE 157  
UPDATE LIBRARY 155  
UPDATE LIBVOLUME 50, 147  
UPDATE MGMTCLASS 461  
UPDATE NODE 380  
UPDATE POLICYSET 460  
UPDATE RECOVERYMEDIA 810  
UPDATE SCHEDULE 573  
UPDATE SCRIPT 584  
UPDATE SERVER 684  
UPDATE SERVERGROUP 711  
UPDATE VOLUME 237  
VALIDATE LANFREE 125  
administrative privilege class  
  description 425  
  granting authority 425  
  reducing 432  
  revoking all 432  
administrative user ID  
  creating automatically 396  
  description of 368  
  preventing automatic creation of 396  
administrative Web interface  
  description 19  
  limitation of browser for script definitions 578  
administrator  
  authorizing to manage a policy domain 425  
  locking 430  
  managing registration 555  
  querying 429  
  registering 428  
  removing 429  
  renaming 429  
  restrictions when registering 428  
  unlocking 430  
  updating 428  
  viewing information about 429  
administrators  
  managing 428  
aggregates  
  controlling the size of 243  
  estimating size 338  
  RECONSTRUCT parameter 353, 515  
  reconstructing 319, 326, 353  
  viewing information about 340, 346

- ANR8914I message 162
- ANR9999D message 628
- application client
  - adding node for 368
  - description 4
  - policy for 483
- application program interface (API)
  - client, registering 371
  - compression option 371
  - deletion option 371
  - description of 3
  - registering to server 371
  - simultaneous-write function, version support for 287
- ARCHFAILOVERLOGDIR server option 610
- archive
  - allowing while file changes 468
  - backup set, uses for 9, 13
  - determining storage usage 348
  - directory 520
  - instant 9, 13
  - package 520
  - policy, defining 457
  - policy, introduction 28
  - process description 455
  - storage usage, minimizing 521
  - storage usage, reducing 521, 522
  - uses for 8, 12
- archive copy group 29
  - defining 468, 469
  - deleting 492
  - description of 443
- archive data
  - expiration 474
  - managing 520
  - protection 474
- archive failover log
  - description 598
  - move to another directory 609
- archive failover log space
  - description 602
- archive log
  - description 597
  - move to another directory 609
- archive log space
  - description 602
- archiving
  - file 441, 455
  - file management 441
  - FILE-type volume, archiving many small objects to 208
- ASCII restriction for browser script definition 578
- ASSIGN DEFMGMTCLASS command 470
- association, client with schedule
  - defining 527
  - deleting 534
- association, file with management class 449, 450
- association, object with profile
  - administrative command schedule 694
  - administrator 690, 705
  - client option set 690
  - deleting 695
  - policy domain 692
  - script 690
- AUDIT LIBVOLUME command 149
- AUDIT LICENSE command 557
- AUDIT VOLUME command 782, 788
- auditing
  - library's volume inventory 149
- auditing (*continued*)
  - license, automatic by server 557
  - multiple volumes in sequential access storage pool 789
  - single volume in sequential access storage pool 789
  - volume in disk storage pool 788
  - volume, reasons for 782
  - volumes by date 790
  - volumes by storage pool 790
- authentication, client/server 432
- authority
  - client access 395
  - granting to administrators 425
  - privilege classes 425
  - server options 425
- auto deployment 380
- auto-update 380, 381, 382, 383, 384, 385, 386, 387, 388
- AUTOFSRENAME parameter 402
- AUTOLABEL parameter for tape volumes 136
- Automated Cartridge System Library Software (ACSL)
  - StorageTek library
    - configuring 112
    - description 44
    - mixing 3592 drive generations 204
    - Tivoli Storage Manager server options for 66
- automated library device
  - auditing 149
  - changing volume status 147
  - checking in volumes 137
  - defining 44
  - informing server of new volumes 137
  - labeling volumes 135
  - overflow location 227
  - removing volumes 148
  - scratch and private volumes 51
  - updating 155
  - volume inventory 51
- automatically renaming file spaces 402
- automating
  - administrative commands 34
  - client operations 32, 526
  - server operations 572
  - server startup 558
- awk script 809, 836

## B

- background processes 564
- backup
  - amount of space used by client 348
  - comparison of types 10, 13
  - database 771
  - default policy 437
  - defining criteria for client files 457
  - differential, for NAS node 10, 57
  - file 440, 452, 454
  - file management 440
  - file while open 462
  - FILE-type volume, backing up small objects to 208
  - frequency for file 463
  - full, for NAS node 57
  - group 12
  - incremental 440, 452
  - logical volume 454
  - NAS file server 175
  - NAS filer to Tivoli Storage Manager server 190
  - policy 28
  - selective 440, 454

- backup (*continued*)
  - SnapMirror to Tape 196
  - snapshot, using hardware 9, 11
  - storage pool 760
  - subfiles, server set-up 27, 511
  - types available 10, 13
  - when to perform for database 768
- backup and restore
  - multiple commands 518
- backup Chris Katsura
  - NAS filer to Tivoli Storage Manager server 191
- BACKUP command 608
- backup copy group 29
  - defining 462, 467
  - deleting 492
  - description of 443
  - frequency 452
  - mode 452
  - serialization 452
- backup data, protecting active 223
- BACKUP DB command 606, 607, 771
- BACKUP DEVCONFIG command 770
- backup period, specifying for incremental 539
- backup set
  - adding subfiles to 513
  - deleting 511
  - description of 503
  - displaying contents of 510
  - example of generating 505
  - generating 503
  - how the server manages and tracks 509
  - media, selecting 504
  - moving to other servers 508
  - OST extension on 504
  - selecting a name for 505
  - selecting retention period for 509
  - suggested usage 9, 27
  - updating 510
  - use as archive 12, 15, 27
- backup sets
  - displaying information 510
  - generate to a point-in-time 506
  - generate with multiple data types 506
- backup sets, managing 509
- BACKUP STGPOOL command 760
- BACKUP VOLHISTORY command 768
- backup volumes
  - create single set 507
- backup-archive client
  - description of 3
  - operations summary 10
  - performing operations for 495, 531, 537
  - policy for 446
  - registering node 368
  - scheduling operations for 526
  - using to back up NAS file server 169, 189
- bar-code reader
  - auditing volumes in a library 149
- barcode reader
  - checking in volumes for a library 140
  - labeling volumes in a library 136
- base file 511
- batch file, scheduling on client 529
- binding a file to a management class 449
- browser, limited to ASCII entry for script definition 578

## C

- cache
  - deleting files from 264, 350
  - description of 25
  - disabling for disk storage pools 263
  - effect on performance 263
  - effect on statistics 264
  - enabling for disk storage pools 227, 263
  - monitoring utilization on disk 344
- CANCEL PROCESS command 343, 565
- CANCEL RESTORE command 418
- CANCEL SESSION command 415
- capacity, tape 217
- cartridge
  - cleaner cartridge 162
  - device support 42
  - mixing drive generations 204
- categories, 349X library 99
- Celerra
  - file server integrated checkpoints 197
- center, administration
  - using 547
- Centera storage device
  - unsupported functions 231
- central scheduling
  - client operations 495, 525, 531, 537
  - controlling the workload 539
  - coordinating 537
  - description of 30, 34, 525
  - server operations 572
- certificate
  - adding to the key database 423, 424
  - homegrown certificate authority 424
- changing date and time on server 563
- changing hostname 568
- characteristics, machine 808
- check in
  - cleaner cartridge 161
  - library volume 137
  - setting a time interval for volume 204
  - VolSafe-enabled volumes 215
- CHECKIN LIBVOLUME command 137
- checking the log file generated by processed schedules 536
- checklist for DRM project plan 831
- CHECKOUT LIBVOLUME command 148
- class, administrator privilege
  - description 425
  - granting authority 425
  - reducing 432
  - revoking all 432
- class, device
  - 3570 199, 200
  - 3590 199, 200
  - 3592 200
  - 4MM 199, 200
  - 8MM 199, 200
  - amount of space used 347
  - CARTRIDGE 200
  - defining 199
  - defining for database backup 767
  - description of 22
  - DISK 199
  - DLT 199, 200
  - DTF 199, 200
  - ECARTRIDGE 200
  - FILE 199
  - FORMAT parameter 201

- class, device (*continued*)
  - GENERICTAPE 199, 200
  - LTO 211
  - OPTICAL 199
  - QIC 199, 200
  - REMOVABLEFILE 207
  - requesting information about 216
  - selecting for import and export 733
  - sequential 200
  - SERVER 199, 200, 714
  - StorageTek devices 200, 215
  - tape 200
  - Ultrium, LTO 200
  - updating 200
  - VOLSAFE 215
  - WORM 199, 200
  - WORM12 199, 200
  - WORM14 199, 200
- class, policy privilege
  - description 425, 430
  - granting 431
  - revoking 431, 432
- class, storage privilege
  - description 425
  - granting 431
  - reducing 431
  - revoking 432
- CLEAN DRIVE command 159
- cleaner cartridge
  - checking in 161
  - how often to use 160
  - operations with 162
  - restrictions on cleaning 160
- CLEANFREQUENCY parameter 160
- client
  - access user ID 395
  - administrative 3
  - API (application program interface) 4, 371
  - application client 4, 483
  - backup-archive 3
  - controlling resource utilization 519
  - how to protect 8
  - operations summary 10
  - options file 372
  - restore without primary volumes available 765
  - Tivoli Storage Manager for Space Management (HSM client) 4, 446
  - using to back up NAS file server 175, 189
- client file
  - allowing archive while changing 437
  - allowing backup while changing 437, 462
  - archive package 520
  - associating with management class 449, 450
  - damaged 765
  - delaying migration of 257
  - deleting 361
  - deleting from a storage pool 360
  - deleting from cache 264
  - deleting when deleting a volume 361
  - duplication when restoring 765
  - eligible for archive 437, 451
  - eligible for backup 437, 451
  - eligible for expiration 439
  - eligible for space management 455
  - how IBM Tivoli Storage Manager stores 243
  - on a volume, querying 338
  - server migration of 252
- client migration 455, 456
- client node
  - adding 367
  - agent 390
  - amount of space used 346
  - creating backup sets for 503
  - file spaces, QUERY OCCUPANCY command 346
  - finding tapes used by 341
  - immediate processing 543
  - importing 746
  - locking 388
  - managing registration 368, 377, 555
  - options file 372
  - performing operations for 495, 531, 537
  - privilege class for scheduling operations for 526
  - proxy node relationships 389
  - querying 392
  - reducing archive packages for 522
  - registering 371
  - removing 389
  - renaming 388
  - scheduling operations for 526
  - setting password authentication 433
  - setting scheduling mode 539
  - setting up subfile backups 512
  - target 390
  - unlocking 388
  - updating 380
  - viewing information about 392
- client nodes
  - defining 398
  - file spaces 398
  - managing 377
  - managing across a firewall 378
  - prevent from accessing server 724
- client option
  - TXNBYTELIMIT 243
  - VIRTUALMOUNTPOINT 397
- client option set
  - adding client options to 412
  - assigning clients to 413
  - copying 413
  - creating 411
  - deleting 413
  - deleting an option from 413
  - for NAS node 175
  - requesting information about 413
  - updating description for 413
- client options
  - UNIX and Linux 373
- client options file
  - create 373
  - update 373
- client point-in-time restore, enabling 488
- client queries to the server, setting the frequency 542
- client reports 650
- client restartable restore session
  - canceling 418
  - interrupting, active 419
  - requesting information about 418
- client session
  - canceling 415
  - DSMC loop 414
  - held volume 414
  - managing 414
  - querying 414, 614
  - viewing information about 414, 614

- client system options file 371
- client-polling scheduling 538, 541
- client, application
  - adding node for 368
  - description 4
  - policy for 483
- client/server, description of 3
- clients
  - adding through the command line client 375
- closed registration
  - description 368
  - process 369
  - setting 368
- collocation
  - active-data pools 316
  - changing, effect of 315
  - copy storage pools 316
  - definition 227, 310, 358
  - description of 25
  - determining whether to use collocation 227, 310, 358
  - effects on operations 311
  - effects on volume reclamation 329
  - enabling for sequential storage pool 227, 310, 358
  - how the server selects volumes when disabled 315
  - how the server selects volumes when enabled 313
  - migration thresholds 258
- command file, scheduling on client 529
- command retry attempts
  - setting the amount of time between 543
  - setting the number of 542
- command routing 707
- command script 578
- commands
  - grant authority 431
- commands, administrative
  - ACCEPT DATE 563
  - ASSIGN DEFMGMTCLASS 470
  - AUDIT LIBVOLUME 149
  - AUDIT LICENSE 557
  - AUDIT VOLUME 788
  - BACKUP DB 606, 607, 771
  - BACKUP DEVCONFIG 770
  - BACKUP NODE 188, 189
  - BACKUP STGPOOL 760
  - BACKUP VOLHISTORY 768
  - BEGIN EVENTLOGGING 628
  - CANCEL PROCESS 565
  - CANCEL RESTORE 418
  - CANCEL SESSION 415
  - CHECKIN LIBVOLUME 137
  - CHECKOUT LIBVOLUME 148
  - CLEAN DRIVE 159
  - COMMIT 590
  - COPY ACTIVATEDATA 225, 246
  - COPY CLOPTSET 413
  - COPY DOMAIN 460
  - COPY POLICYSET 460
  - COPY SCHEDULE 532
  - COPY SCRIPT 585
  - COPY SERVERGROUP 711
  - DEFINE ASSOCIATION 32, 527
  - DEFINE BACKUPSET 508
  - DEFINE CLIENTACTION 543
  - DEFINE CLIENTOPT 543
  - DEFINE CLOPTSET 411
  - DEFINE COPYGROUP 462, 467, 468
  - DEFINE DATAMOVER 128, 186

- commands, administrative (*continued*)
  - DEFINE DEVCLASS
    - 3592 204
    - FILE device classes 207
    - LTO device classes 211
    - REMOVEABLEFILE device classes 207
    - SERVER device classes 214
    - tape device classes 200
    - VOLSAFE device classes 215
  - DEFINE DRIVE 127
  - DEFINE GRPMEMBER 710
  - DEFINE LIBRARY 44, 126
  - DEFINE MACHINE 808
  - DEFINE MACHNODEASSOCIATION 808
  - DEFINE PATH 129, 187
  - DEFINE POLICYSET 460, 461
  - DEFINE PROFASSOCIATION 690, 691
  - DEFINE PROFILE 690
  - DEFINE RECMEDMACHASSOCIATION 810
  - DEFINE RECOVERYMEDIA 810
  - DEFINE SCHEDULE 573, 606
  - DEFINE SCRIPT 578
  - DEFINE SERVER 678, 707, 714
  - DEFINE SERVERGROUP 710
  - DEFINE STGPOOL 231, 232, 242, 243
  - DEFINE SUBSCRIPTION 701
  - DEFINE VIRTUALFSMAPPING 195
  - DEFINE VOLUME 50, 237
  - DELETE ASSOCIATION 534
  - DELETE BACKUPSET 511
  - DELETE COPYGROUP 492
  - DELETE DOMAIN 493
  - DELETE DRIVE 163
  - DELETE EVENT 536
  - DELETE GRPMEMBER 712
  - DELETE LIBRARY 156
  - DELETE MGMTCLASS 493
  - DELETE POLICYSET 493
  - DELETE PROFASSOCIATION 695
  - DELETE PROFILE 697
  - DELETE SCHEDULE 532
  - DELETE SCRIPT 586
  - DELETE SERVER 684
  - DELETE SERVERGROUP 711
  - DELETE STGPOOL 361
  - DELETE SUBSCRIBER 706
  - DELETE SUBSCRIPTION 697, 702
  - DELETE VOLHISTORY 768, 769
  - DELETE VOLUME 362, 363
  - DISABLE EVENTS 627
  - DISABLE SESSIONS 417
  - DISMOUNT VOLUME 154
  - DSMSERV DISPLAY DBSPACE 603
  - DSMSERV DISPLAY LOG 603
  - DSMSERV FORMAT 774
  - ENABLE EVENTS 627
  - ENABLE SESSIONS 417
  - END EVENTLOGGING 628
  - EXPIRE INVENTORY 34
  - EXPORT ADMIN 721
  - EXPORT NODE 734
  - EXPORT POLICY 734
  - EXPORT SERVER 734
  - EXTEND DBSPACE 604
  - GENERATE BACKUPSET 503
  - GRANT AUTHORITY 425
  - HALT 561

commands, administrative (continued)

HELP 569  
 IMPORT 748, 750  
 IMPORT ADMIN 737  
 IMPORT NODE 737, 746  
 IMPORT POLICY 737  
 IMPORT SERVER 737, 746  
 LABEL LIBVOLUME 94, 123  
 LOCK ADMIN 430  
 LOCK NODE 388  
 LOCK PROFILE 694, 695  
 MOVE DATA 350  
 MOVE NODEDATA 354  
 NOTIFY SUBSCRIBERS 695  
 PING SERVER 712  
 PREPARE 810  
 QUERY ACTLOG 624  
 QUERY BACKUPSETCONTENTS 510  
 QUERY CONTENT 338  
 QUERY COPYGROUP 490, 744  
 QUERY DB 603  
 QUERY DBSPACE 603  
 QUERY DEVCLASS 216  
 QUERY DOMAIN 491  
 QUERY DRIVE 156  
 QUERY DRMSTATUS 801  
 QUERY ENABLED 640  
 QUERY EVENT 528  
 QUERY FILESPACE 409  
 QUERY LIBRARY 155  
 QUERY LICENSE 557  
 QUERY MGMTCLASS 490  
 QUERY MOUNT 154  
 QUERY NODE 392  
 QUERY NODEDATA 347  
 QUERY OCCUPANCY  
     backed-up, archived, and space-managed files 348  
     client file spaces 346  
     client nodes 346  
     device classes 347  
     storage pools 347  
 QUERY OPTION 616  
 QUERY POLICYSET 491  
 QUERY PROCESS 352  
 QUERY REQUEST 152  
 QUERY RESTORE 418  
 QUERY RPFCONTENT 812  
 QUERY RPFIL 812  
 QUERY SCHEDULE 528  
 QUERY SCRIPT 585  
 QUERY SERVERGROUP 711  
 QUERY STGPOOL 332, 342, 740  
 QUERY SUBSCRIPTION 701  
 QUERY SYSTEM 617  
 QUERY VOLHISTORY 768  
 QUERY VOLUME 335, 353  
 RECONCILE VOLUMES 718  
 REGISTER ADMIN 428  
 REGISTER LICENSE 556  
 REMOVE ADMIN 429  
 REMOVE NODE 389  
 RENAME ADMIN 429  
 RENAME FILESPACE 748  
 RENAME NODE 388  
 RENAME SCRIPT 586  
 RENAME SERVERGROUP 711  
 RENAME STGPOOL 357

commands, administrative (continued)

RESTORE DB 562  
 RESTORE NODE 188, 189  
 RESTORE STGPOOL 763, 794  
 RESTORE VOLUME 796  
 ROLLBACK 591  
 RUN 586  
 SELECT 618  
 SET ACCOUNTING 645  
 SET AUTHENTICATION 433  
 SET CLIENTACTDURATION 543  
 SET CONFIGMANAGER 687, 689  
 SET CONFIGREFRESH 701  
 SET CONTEXTMESSAGING 628  
 SET CROSSDEFINE 679, 682  
 SET DBRECOVERY 606  
 SET DBREPORTMODE 603  
 SET DRMCHECKLABEL 806  
 SET DRMCOPYSTGPOOL 802  
 SET DRMCOURIERNAME 805  
 SET DRMDBBACKUPRXPIREDAYS 806  
 SET DRMFILEPROCESS 806  
 SET DRMINSTPREFIX 804  
 SET DRMNOTMOUNTABLE 805  
 SET DRMPANPOSTFIX 803  
 SET DRMPANPREFIX 804  
 SET DRMPRIMSTGPOOL 802  
 SET DRMRPFEXPIREDAYS 812  
 SET DRMVAULTNAME 806  
 SET EVENTRETENTION 536, 577  
 SET INVALIDPWLIMIT 433  
 SET LICENSEAUDITPERIOD 557  
 SET MAXCMDRETRIES 542  
 SET MAXSCHEDSESSIONS 540  
 SET MINPWLENGTH 433  
 SET PASSEXP 432  
 SET QUERYSCHEDPERIOD 542  
 SET RANDOMIZE 540  
 SET REGISTRATION 368  
 SET RETRYPERIOD 543  
 SET SCHEDMODES 538  
 SET SERVERHLADDRESS 679, 682  
 SET SERVERLLADDRESS 679, 682  
 SET SERVERNAME 616, 678, 682  
 SET SERVERPASSWORD 678, 679, 682  
 SET SUBFILE 511  
 SET SUMMARYRETENTION 622  
 SETOPT 569  
 UNLOCK PROFILE 694, 695  
 UPDATE ADMIN 428  
 UPDATE ARCHIVE 522  
 UPDATE BACKUPSET 510  
 UPDATE CLIENTOPT 413  
 UPDATE CLOPTSET 413  
 UPDATE COPYGROUP 462, 468  
 UPDATE DEVCLASS 200  
 UPDATE DOMAIN 460  
 UPDATE DRIVE 157  
 UPDATE LIBRARY 155  
 UPDATE LIBVOLUME 50, 147  
 UPDATE MGMTCLASS 461  
 UPDATE NODE 380  
 UPDATE POLICYSET 460  
 UPDATE RECOVERYMEDIA 810  
 UPDATE SCHEDULE 573  
 UPDATE SCRIPT 584  
 UPDATE SERVER 684

- commands, administrative (*continued*)
  - UPDATE SERVERGROUP 711
  - UPDATE VOLUME 237
  - VALIDATE LANFREE 125
- COMMIT command 590
- Common reporting 647
- COMMTIMEOUT server option 415, 416
- communication set up
  - among servers 676
  - command routing, for 680
  - cross definition 677, 678, 682
  - enterprise configuration, for 676
  - enterprise event logging, for 639, 676
  - security 679
  - server-to-server virtual volumes 714
  - using Secure Sockets Layer 421
- compression
  - choosing client or drive 218
  - option for API 371
  - options for clients 370
  - setting 370
  - tape volume capacity, effect on 218
- configuration file, device
  - backing up 770
  - information 770
- configuration information, enterprise management
  - administrative command schedule 687, 694
  - administrator 690, 705
  - changing 694
  - client option set 687, 690
  - client schedule 687, 688, 692
  - copy group 687, 692
  - deleting 695, 696
  - distributing 685, 691, 695
  - management class 692
  - policy domain 687, 688, 692
  - refreshing 695, 701, 703
  - script 687, 690
  - server 693
  - server group 693
- configuration manager
  - communication setup 676
  - default profile 686, 693
  - scenario 686
  - setting up 686, 689, 690
- configuring
  - connect 349x library to server 183
  - connecting ACSLS library to server 184
  - devices, automated library example 91, 100
  - devices, manual library example 122
  - NDMP operations for NAS file servers 173
  - planning your storage environment 65
  - shared library 95, 105
- configuring storage 89
- console mode 750
- contents of a volume 338
- context messaging for ANR9999D 628
- continuation characters, using 582
- conventions
  - typographic xix
- COPY CLOPTSET command 413
- COPY DOMAIN command 460
- copy group
  - archive, description of 443
  - backup, description of 443
  - defining archive 468
  - defining backup 462
- copy group (*continued*)
  - deleting 492
- COPY MGMTCLASS command 461
- COPY POLICYSET command 460
- COPY SCHEDULE command 532, 576
- COPY SCRIPT command 585
- COPY SERVERGROUP command 711
- copy storage pool
  - compared with primary 359
  - defining a 357
  - restore from multiple 765
  - role in storage pool migration 263
  - simultaneous-write function 285
- creating
  - backup sets 27
  - client schedules 32
  - new policy 29
  - server scripts 578
- creating backup sets
  - benefits of 503
  - example for 505
- cross definition 677, 678, 682
- current server status workspaces 657
- custom reporting 647
- customer support
  - contact xvii
- cyclic redundancy check
  - during a client session 495
  - for storage pool volumes 784
  - for virtual volumes 713
  - performance considerations for nodes 496
  - performance considerations for storage pools 788

## D

- daily monitoring 646
- damaged files 757, 790, 791
- data
  - active backup versions, storing 223
  - considering user needs for recovering 64
  - exporting 721
  - importing 721
- data compression 370
- data deduplication
  - client-side 280
    - changing location 282
    - client and server settings 264, 276
    - multiple nodes 281
    - overview 266
    - single node 281
  - controlling duplicate-identification manually 278
  - DEDUPLICATION parameter 277
  - DEDUPREQUIRESBACKUP server option 275
  - definition 264
  - detecting security attacks 271
  - duplicate-identification processes 274, 277, 280
  - IDENTIFY DUPLICATES command 278
  - improving read performance 284
  - limitations 269
  - managing 274
  - moving or copying data 275
  - planning 270
  - protecting data 275
  - reclamation 275
  - server-side 264, 265, 276, 277
  - specifying the size of objects to be deduplicated 282

- data deduplication (*continued*)
  - statistics
    - displaying information about files with links to a volume 283
    - querying a duplicate-identification process 283
    - querying a storage pool 283
  - testing
    - restore operations 272
    - space savings 273
  - turning off 276
  - virtual volumes, server-to-server
    - data deduplication 275
- data format for storage pool 170, 172
  - definition 227
  - operation restrictions 231
- data movement, querying 352
- data mover
  - defining 128, 186
  - description 49, 50
  - managing 171
  - NAS file server 49, 50
- data protection with WORM media 141
- data retention protection 474, 550
- data retention using Centera
  - unsupported functions 231
- data shredding
  - BACKUP STGPOOL command 501, 761
  - COPY ACTIVATEDATA command 501
  - DEFINE STGPOOL command 501
  - DELETE FILESPACE, command 501
  - DELETE VOLUME, command 501
  - description 499
  - enforcing 501
  - EXPIRE INVENTORY command 501
  - EXPORT NODE command 501, 722
  - EXPORT SERVER command 501, 722
  - GENERATE BACKUPSET command 501, 502
  - MOVE DATA command 350, 501
  - setting up 500
  - UPDATE STGPOOL command 501
- data storage
  - active-data pools 223
  - client files, process for storing 5
  - concepts overview 15
  - considering user needs for recovering 64
  - deleting files from 361
  - example 225
  - managing 21
  - monitoring 782
  - planning 65
  - protection, methods 755
  - server options affecting 66
  - tailoring definitions 744
  - using another IBM Tivoli Storage Manager server 712
  - using disk devices 69
  - using the storage hierarchy 251
- data validation
  - during a client session 495
  - for storage pool volumes 784
  - for virtual volumes 713
  - performance considerations for nodes 496
  - performance considerations for storage pools 788
- database
  - audits 594
  - backup 605, 606
  - buffer size 594
  - description of 593
  - database (*continued*)
    - estimating the space needed 599
    - full backup 605
    - increase the size 35
    - increasing the size 604
    - incremental backup 605
    - managing 593
    - manual backup 607
    - monitoring 603
    - move to another directory 608
    - moving 562
    - relocating on a server 607
    - reorganization 594
    - restoring to a point in time 607
    - restoring to most current state 607
    - SQL queries 594
    - statistics collection 594
    - transactions 593, 610
  - database backup
    - manual 607
    - scheduling 606
  - database backup and recovery
    - database backup trigger 776
    - defining device classes for 767
    - example recovery procedures 792
    - full backup 768, 771
    - general strategy 712, 755
    - incremental backup 768
    - methods 712, 755
    - point-in-time 774
    - providing 712, 755
    - roll-forward 776
    - to most current state 776
    - when to back up 768
  - database restore
    - most current 759
    - point-in-time 759
  - database,
    - restoring 767
  - database, IBM Tivoli Storage Manager
    - backup 771
    - description of 34
    - ensuring integrity of 37
    - mirroring 759
    - querying using SQL 618
    - recovering 772
  - date and time, changing on the server 563
  - day of the week parameter 573
  - DB2 tools, use of 593
  - DB2 Universal Database
    - Enterprise Extended Edition 390
  - DBMEMPERCENT server option 606
  - deactivating policy 439
  - deduplicate-identification processes 278
  - default management class 29
    - assigning for a policy set 470
    - binding files to 451
    - description of 443
    - purpose 447
    - recommendation for using 449
  - default policy 437
  - default profile 686, 693, 700
  - DEFINE ASSOCIATION command 32, 527
  - DEFINE BACKUPSET command 508
  - DEFINE CLIENTACTION command 543
  - DEFINE CLIENTOPT command 412
  - DEFINE CLOPTSET command 411

DEFINE COPYGROUP command 462, 467, 468, 469  
 DEFINE DEVCLASS command 200  
 DEFINE DRIVE command 127  
 DEFINE GRPMEMBER command 710  
 DEFINE LIBRARY command 126  
 DEFINE MACHNODEASSOCIATION command 808  
 DEFINE MGMTCLASS command 461  
 DEFINE POLICYSET command 460  
 DEFINE PROFASSOCIATION command 691  
 DEFINE PROXYNODE command 390  
 DEFINE RECMEDMACHASSOCIATION command 810  
 DEFINE RECOVERYMEDIA command 810  
 DEFINE SCHEDULE command 32, 34, 573, 606  
 DEFINE SCRIPT command 578  
 DEFINE SERVER command 678, 707, 714  
 DEFINE STGPOOL command 231, 232, 242, 243  
 DEFINE SUBSCRIPTION command 701  
 DEFINE VIRTUALFSMAPPING command 195  
 DEFINE VOLUME command 23, 237  
 defining  
     archive copy group 29  
     backup copy group 29  
     client nodes 398  
     management class 29  
     policy 29  
     policy set 29  
 definitions 883  
 delaying migration for files 257  
 delaying reuse of volumes 329  
 DELETE ASSOCIATION command 534  
 DELETE BACKUPSET command 511  
 DELETE CLIENTOPT command 413  
 DELETE COPYGROUP command 492  
 DELETE DOMAIN command 493  
 DELETE DRIVE command 163  
 DELETE EVENT command 536, 578  
 DELETE FILESPACE command 410  
 DELETE GRPMEMBER command 712  
 DELETE KEYRING command 424  
 DELETE LIBRARY command 156  
 DELETE MGMTCLASS command 493  
 DELETE POLICYSET command 493  
 DELETE PROFASSOCIATION command 695  
 DELETE PROFILE command 697  
 DELETE SCHEDULE command 532, 576  
 DELETE SCRIPT command 586  
 DELETE SERVER command 684  
 DELETE SERVERGROUP command 711  
 DELETE STGPOOL command 361  
 DELETE SUBSCRIBER command 706  
 DELETE SUBSCRIPTION command 702  
 DELETE VOLHISTORY command 768, 769  
 DELETE VOLUME command 362, 363  
 deleting  
     cached files on disk 350  
     empty volume 362, 769  
     file spaces 410  
     files 361, 472  
     media-managed storage pools 121  
     scratch volume 235, 769  
     storage volume 362  
     subfile 513  
     volume history information 769  
     volume with residual data 363  
 deletion hold 475  
 deployment  
     automatic 380  
     deployment (*continued*)  
         command-line interface 384, 385  
         configure 382  
         FTP site 381  
         importing 386  
         importing backup-archive client packages 382  
         mirror FTP site 383  
         overview 381  
         properties file 383  
         requirements 382  
         schedule 384, 387  
         verifying 388  
     descriptions, for archive packages 520, 521  
     DESTINATION parameter (storage pool) 437, 462  
     destroyed volume access mode 240, 757  
     determining  
         cause of ANR9999D messages 628  
         the time interval for volume check in 204  
     DEVCONFIG option 770  
     device  
         attaching to server 180  
         element number 186  
         multiple types in a library 58  
         name 87  
         supported by IBM Tivoli Storage Manager 42  
         zfc device driver 84  
     device class  
         3570 199, 200  
         3590 199, 200  
         3592 200  
         4MM 199, 200  
         8MM 199, 200  
         amount of space used 347  
         CARTRIDGE 200  
         defining 199  
         defining for database backup 767  
         description of 22  
         DISK 199  
         DLT 199, 200  
         DTF 199, 200  
         ECARTRIDGE 200  
         FILE 199  
         FORMAT parameter 201  
         GENERICTAPE 199, 200  
         LTO 211  
         OPTICAL 199  
         QIC 199, 200  
         REMOVABLEFILE 207  
         requesting information about 216  
         selecting for import and export 733  
         sequential 200  
         SERVER 199, 200, 714  
         StorageTek devices 200, 215  
         tape 200  
         Ultrium, LTO 200  
         updating 200  
         VOLSAFE 215  
         WORM 199, 200  
         WORM12 199, 200  
         WORM14 199, 200  
     device configuration file  
         backing up 770  
         information 770  
     device driver  
         configuring 83  
         for IBM 3490, 3570, and 3590 tape drives 82  
         for manual tape devices 79, 80

- device driver (*continued*)
  - IBM Tivoli Storage Manager, installing 79, 80, 89
  - installing 79, 80
  - requirements 79, 80
- device drivers 81
  - installing 82
- device sharing 65
- device support 15
- device type
  - 3570 199, 200
  - 3590 200
  - 4MM 199, 200
  - 8MM 199, 200
  - CARTRIDGE 200
  - DISK 199
  - DLT 199, 200
  - DTF 199, 200
  - ECARTRIDGE 200
  - FILE 199
  - GENERICTAPE 199, 200
  - LTO 200, 211
  - multiple in a single library 58
  - OPTICAL 199
  - QIC 199, 200
  - REMOVABLEFILE 199
  - SERVER 199, 200, 714, 716
  - VOLSAFE 215
  - WORM 199, 200
  - WORM12 200
  - WORM14 200
- device, storage
  - automated library device 91, 100
  - disk 69
  - manual library device 122
  - optical device 118, 122
  - removable media device 118, 207
  - required IBM Tivoli Storage Manager definitions 63
  - supported devices 42
- devices
  - defining 126
- diagnosing ANR9999D messages 628
- differential backup
  - compared to incremental 13
  - of image, description 10, 57
- direct-to-tape, policy for 482
- directories
  - deleting from archive packages 522
  - directory-level backup 195
  - preventing archive of 523
  - storage usage for archive packages 520
- DISABLE EVENTS command 627
- DISABLE SESSIONS command 417
- disaster recovery
  - auditing storage pool volumes 791
  - example recovery procedures 792
  - general strategy 712, 755
  - methods 37, 712, 755
  - providing 712, 755
  - server
    - disaster recovery 823
    - server recovery 823
    - when to backup 755, 768
- disaster recovery manager
  - awk script 836
  - client recovery information 801
  - creating a disaster recovery plan 810
  - customizing 801
  - disaster recovery manager (*continued*)
    - displaying a disaster recovery plan 812
    - expiring a disaster recovery plan 812
    - features 801
    - moving volumes back on-site 817
    - not available in Administration Center 550
    - project plan, checklist 831
    - querying a disaster recovery plan 812
    - recovery media 810
    - saving machine characteristics 808
    - stanzas, recovery instructions 807
    - storing a disaster recovery plan 810
  - disaster, protecting against 36
  - disk device class, defined 199
  - disk devices
    - random access 71
    - sequential access 71
  - disk space
    - for the database and recovery log 599
  - disk storage
    - and file-system requirements 69
    - random access (DISK) 45
    - sequential access (FILE) 46
  - disk storage pool
    - cache, use of 264
    - deleting cached files from 350
    - estimating space 330
    - estimating space for archived files 331
    - estimating space for backed up files 331
    - migration threshold 253
    - setting up 69
  - DISMOUNT VOLUME command 154
  - display information 392
    - specific client node 392
  - DISPLAY OBJNAME command 399
  - distribute workloads
    - to reduce backup and restore time 390
  - distribution 380, 384
  - DLT WORM media 141
  - domain, policy
    - active-data pools, specifying 458
    - associating a client with 29
    - changing 439
    - creating 460
    - deleting 493
    - description of 443
    - distributing via profile 489, 688
    - for NAS file server node 174
    - querying 491
    - updating 456, 458
  - drive
    - cleaning 159
    - defining 127
    - defining path for 129
    - deleting 163
    - detecting changes on a SAN 131
    - element address 127, 129
    - multiple device types in a library 93
    - querying 156
    - serial number 127
    - server retries for acquiring 67
    - simultaneous-write function, requirements for 306
    - updating 157
    - updating to use for NDMP operations 172
  - DRIVEACQUIRERETRY server option 67
  - DRIVEENCRYPTION parameter
    - 3592 device class 206

- DRIVEENCRYPTION parameter *(continued)*
  - ECARTRIDGE device class 216
  - LTO device class 213
- driver, device
  - configuring 83
  - for IBM 3490, 3570, and 3590 tape drives 82
  - for manual tape devices 79, 80
  - IBM Tivoli Storage Manager, installing 79, 80, 89
  - installing 79, 80
  - requirements 79, 80
- dsm.opt file 372, 411, 525
- dsmacct.log 645
- DSMADMC command 730, 743, 750
- DSMC loop session 414
- dmsched.log file 536
- DSMSERV DISPLAY DBSPACE command 603
- DSMSERV DISPLAY LOG command 603, 605
- DSMSERV FORMAT utility 560
- DSMSERV\_ACCOUNTING\_DIR 645
- duplicate-identification processes 274, 277
- duplication of restored data 765
- DVD-RAM support
  - defining and updating a device class 207
- dynamic serialization, description of 462, 468

## E

- ECARTRIDGE device class 200
- education
  - see Tivoli technical training xv
- element address 127
- ENABLE EVENTS command 627
- ENABLE SESSIONS command 417
- encoding events to UTF-8 632
- encryption
  - changing method 498
  - choosing a method 497
  - DRIVEENCRYPTION parameter
    - 3592 Generation 2 206
    - ECARTRIDGE 216
    - LTO-4 213
  - methods 158, 496
- END EVENTLOGGING command 628
- Enterprise Administration
  - description 671
- enterprise configuration
  - communication setup 676
  - description 672, 685
  - procedure for setup 685
  - profile for 686
  - scenario 674, 685
  - subscription to 688
- enterprise event logging 639, 676
- environment file
  - modifying queries 669
  - modifying reporting performance 669
- environment variable, accounting 645
- environment variables 560
- error analysis 617
- error checking for drive cleaning 163
- error reporting for ANR9999D messages 628
- error reports for volumes 336
- establishing server-to-server communications
  - enterprise configuration 676
  - enterprise event logging 676
  - virtual volumes 684
- estimated capacity for storage pools 333

- estimated capacity for tape volumes 336
- event logging 550, 626, 632
- event record (for a schedule)
  - deleting 536, 578
  - description of 528, 535
  - managing 576
  - querying 577
  - removing from the database 536, 577
  - setting retention period 536, 577
- event server 639
- example
  - assigning a default management class 470
  - register three client nodes with CLI 375
  - validating and activating a policy set 472
- expiration date, setting 574
- expiration processing 34
  - description 766
  - files eligible 439, 472
  - of subfiles 439, 464, 472, 513
  - starting 472
  - using disaster recovery manager 473
- EXPIRE INVENTORY command 34
  - duration of process 473
- export
  - administrator information 730
  - client node information 731
  - data from virtual volumes 751
  - decided when 723
  - directly to another server 724
  - labeling tapes 726, 733
  - monitoring 748
  - options to consider 724
  - planning for sequential media 733
  - policy information 732
  - PREVIEW parameter 732
  - previewing results 729
  - querying about a process 748
  - querying the activity log 750
  - replacing definitions before 726
  - server data 732
  - using scratch media 733
  - viewing information about a process 748
- EXPORT ADMIN command 735
- export and import data
  - sequential media volumes 732
- EXPORT commands 749, 750
- EXPORT NODE command 735
- EXPORT POLICY command 736
- EXPORT SERVER command 732, 736
- exporting
  - administrator data 735
  - client node data 735
  - data to tape 734
  - description of 721
  - policy data 736
  - server data 736
  - subfiles 513
- EXPQUIET server option 473
- EXTEND DBSPACE command 604
- EXTERNAL library type 864
- external media management
  - IBM Tivoli Storage Manager setup 120
  - initialization requests 864
  - interface description 859
  - overview 120
  - processing during server initialization 860
  - using with IBM Tivoli Storage Manager 121

external media management (*continued*)

volume dismount requests 869

volume mount requests 866

volume release requests 865

## F

file data, importing 721

file deletion option

setting 372

FILE device type

active-data pools 223

backing up or archiving many small objects 208

benefits 46

concurrent access to FILE volumes 46

defining device class 199

deleting scratch volumes 769

free space in directories 348

setting up storage pool 76

file exit 626

logging events to 629

file name for a device 87, 187

file path name 399

file retrieval date 264

file server, network-attached storage (NAS)

backup methods 175

registering a NAS node for 185

using NDMP operations 55, 167

file size, determining maximum for storage pool 227

file space

deleting, effect on reclamation 319

deleting, overview 410

description of 397

merging on import 725, 738

names that do not display correctly 409

QUERY OCCUPANCY command 346

querying 397

renaming 748

Unicode enabled 408

viewing information about 397

file space identifier (FSID) 408

file spaces

defining 398

FILE volumes

shared 130

file-level restore

managing 193

planning 192

file-system requirements for disk storage 69

file, client

allowing archive while changing 437

allowing backup while changing 437, 462

archive package 520

associating with management class 449, 450

damaged 765

delaying migration of 257

deleting 361

deleting from a storage pool 360

deleting from cache 264

deleting when deleting a volume 361

duplication when restoring 765

eligible for archive 437, 451

eligible for backup 437, 451

eligible for expiration 439

eligible for space management 455

how IBM Tivoli Storage Manager stores 243

on a volume, querying 338

file, client (*continued*)

server migration of 252

files, damaged 757, 765, 790, 791

files, unreadable 757, 790, 791

firewall, client nodes

client-initiated sessions 378

server-initiated sessions 379

fixes, obtaining xvii

format for storage pool 170, 172

definition 227

operation restrictions 231

formatting

event classes 630

storage pool volume 75, 236

frequency of backup 463

FSID 408

full image backup, NAS node 57

full library 148

## G

GENERATE BACKUPSET command 503

Global Security Kit (GSKit) 421

glossary 883

GRANT AUTHORITY command 425

group backup, on the client 12

group, server

copying 711

defining 710

deleting 711

member, deleting 712

moving a member 712

querying 711

renaming 711

updating description 711

## H

HALT command 561

halting the server 561

held volume in a client session 414

HELP command 569

hierarchy, storage 24

copying active backup data 223

defining in reverse order 231, 242

establishing 241

example 225

for LAN-free data movement 241

how the server stores files in 243

next storage pool

definition 241

deleting 361

migration to 252, 342

restrictions 241

staging data on disk for tape storage 251

historical reports

server trends 654

HL ADDRESS 379

hostname

changing 568

how to cause the server to accept date and time 563

## I

IBM error analysis 617

- IBM Software Support
  - submitting a problem xix
- IBM Support Assistant xvi
- IBM Tivoli Monitoring 647
- IBM Tivoli Storage Manager (Tivoli Storage Manager)
  - introduction 3
  - server network 35
- IDLETIMEOUT server option 415, 416
- image backup
  - policy for 483, 485
  - suggested use 8, 11
- import
  - data from virtual volumes 751
  - monitoring 748
  - options to consider 738
  - PREVIEW parameter 732, 740
  - querying about a process 748
  - querying the activity log 750
  - recovering from an error 747
  - replace existing definitions 739
  - viewing information about a process 748
- IMPORT ADMIN command 737
- IMPORT commands 749, 750
- IMPORT NODE command 737, 746
- IMPORT POLICY command 737
- IMPORT SERVER command 737, 746
- importing
  - active-data pools 745
  - data 737
  - data storage definitions 742, 744
  - date of creation 739, 745
  - description of 721
  - directing messages to an output file 730, 743
  - duplicate file spaces 745
  - file data 745
  - policy definitions 742
  - server control data 743
  - subfiles 513
  - subsets of information 747
- include-exclude file 31
  - creating 31
  - description of 28, 448
  - for policy environment 443, 448
- incomplete copy storage pool, using to restore 765
- incremental backup 452
- incremental backup, client
  - file eligibility for 452
  - frequency, specifying 539
  - full 452
  - partial 453
  - progressive 13
- inheritance model for the simultaneous-write function 293
- initial start date for schedule 573
- initial start time for schedule 573
- initializing
  - tape volumes 23
- installing IBM Tivoli Storage Manager 368
- instance user ID 560
- instant archive
  - creating on the server 502
  - description of 9, 13
- interface, application program
  - client, registering 371
  - compression option 371
  - deletion option 371
  - description of 3
  - registering to server 371

- interface, application program (*continued*)
  - simultaneous-write function, version support for 287
- interfaces to IBM Tivoli Storage Manager 19
- Internet, searching for problem resolution xvi, xvii
- introduction to IBM Tivoli Storage Manager 3

## K

- key database
  - adding certificates 423, 424
  - password change 423, 424
- knowledge bases, searching xvi

## L

- label
  - automatic labeling in SCSI libraries 136
  - checking media 140
  - overwriting existing labels 134, 135
  - sequential storage pools 133, 237
  - volume examples 135
  - volumes using a library device 135
- LABEL LIBVOLUME command
  - identifying drives 134
  - insert category 137
  - labeling sequential storage pool volumes 134
  - manually mounted devices 122
  - overwriting existing volume labels 134
  - removable media volumes 134
  - restrictions for VolSafe-enabled drives 215
  - SCSI libraries used by one server 91
  - using a library device 135
  - using a manual library 123
  - using an automated library 94, 104, 117
  - volume labeling examples 135
- LAN-free data movement 124
  - configuration 124
  - description 14, 54
  - storage pool hierarchy restriction 241
  - suggested usage 10
- libraries
  - NDMP operations 179
- library
  - ACSL (Automated Cartridge System Library Software) 44, 112
  - adding volumes 137
  - attaching for NAS file server backup 180
  - auditing volume inventory 149
  - automated 147
  - categories for volumes in IBM 3494 99
  - configuration example 91, 100, 122
  - configure for more than one device type 58
  - defining 126, 156
  - defining path for 129, 187
  - deleting 156
  - detecting changes to, on a SAN 126, 131
  - external 44
  - full 148
  - IBM 3494 44, 99
  - managing 154
  - manual 43, 122, 152
  - mixing device types 58, 204, 211
  - mode, random or sequential 80
  - overflow location 227
  - querying 155
  - SCSI 44

- library (*continued*)
  - serial number 126
  - sharing among servers 95, 105
  - type 52
  - updating 155
  - volume inventory 51
- library client, shared library 53, 98, 106, 116
- library drive usage
  - determining 177
- library manager, shared library 53, 96, 106, 115
- license
  - compliance 557
  - features 555
  - monitoring 557
  - registering 556
  - using 555
- limitation for script definition on administrative Web interface 578
- limitations, Administration Center 550
- LL ADDRESS 379
- location, volume
  - changing 238
  - overflow for storage pool 227
  - querying volume 337
- LOCK ADMIN command 430
- LOCK NODE command 388
- LOCK PROFILE command 694, 695
- log mirror 596, 605
- logical devices 76
- logical volume on client
  - backup 440
  - management class for 449
  - policy for 451, 483
  - process for backup 454
  - restore 441
- logical volume, raw 75
- loop session, DSMC 414
- LTO Ultrium devices and media
  - device class, defining and updating 211
  - encryption 158, 213, 496
  - WORM 141, 215
- LUN
  - using in paths 129

## M

- machine characteristics 808
- machine recovery information 809
- macro
  - commit individual commands 590
  - continuation characters 589
  - controlling command processing 590
  - running 590
  - scheduling on client 529
  - substitution variables 589
  - testing 591
  - using 587
  - writing commands 588
  - writing comments 588
- MACRO administrative command, using 375
- magnetic disk devices 45, 69
- maintenance 380, 384
- maintenance distribution 380, 384
- maintenance plan
  - modify 581
- maintenance script
  - create 579

- maintenance script (*continued*)
  - custom 580
  - modify 34
- maintenance updates 380, 384
- managed server
  - changing the configuration manager 700, 706
  - communication setup 676
  - deleting a subscription 702
  - description 672
  - managed objects 672, 699
  - refreshing configuration information 703
  - renaming 706
  - returning managed objects to local control 704
  - setting up 688
  - subscribing to a profile 688, 699, 700, 701
- management class
  - assigning a default 470
  - associating a file with 449
  - binding a file to 449
  - configuration 446
  - controlling user access 446
  - copying 456, 461
  - default 32, 447
  - define new 487
  - defining 461
  - deleting 493
  - description of 443, 446
  - querying 490
  - rebinding a file 451
  - updating 450, 456, 461
- managingserver operation 33
- manual drive
  - attaching 79
- manual library
  - defining devices 122
- manual library device 122
- MAXSCRATCH parameter 227, 238, 358
- media
  - loss, recovery from 796
  - tape rotation 60, 144
- media label
  - checking 140
  - for tape 134
  - recording 134
- merging file spaces 725, 738
- messages
  - determining cause of ANR9999D message 628
  - directing import messages to an output file 730, 743
  - for automated libraries 152
  - for drive cleaning 163
  - getting help on 569
  - mount, using the administrative client 152
  - severe 628
- Microsoft Management Console (MMC) snap-in 19
- MIGRATE STGPOOL command 261
- migrating a file 441, 455
- migration, client
  - automatic, for HSM client
    - demand 442
    - files, eligible 455
    - threshold 442
    - using management class 456
  - premigration for HSM client 442
  - reconciliation 442
  - selective, for HSM client 442
  - stub file on HSM client 442

- migration, server
  - canceling the server process 343
  - controlling by file age 257
  - controlling duration 261
  - controlling start of, server 256
  - copy storage pool, role of 263
  - defining threshold for disk storage pool 256
  - defining threshold for tape storage pool 258
  - delaying by file age 257
  - description, server process 254
  - minimizing access time to migrated files 258
  - monitoring thresholds for storage pools 342
  - multiple concurrent processes
    - random access storage pool 227, 254
    - sequential access storage pool 227, 261
  - problems, diagnosing and fixing 252
  - providing additional space for server process 344
  - starting manually 261
  - starting server process 251, 256
  - threshold for a storage pool
    - random access 254
    - sequential access 258, 259
- mirroring
  - active log 759
  - advantages 759
  - description of 37
- MIRRORLOGDIR server option 610
- mixed device types in a library 58, 204, 211
- mobile client support 511
- mode
  - client backup 463
  - library (random or sequential) 80
  - scheduling 538
- modified mode, description of 463
- monitoring
  - server-to-server export 730
- monitoring the Tivoli Storage Manager server 613
- monitoring workspaces
  - agent status 657
  - availability 657
  - client missed files 657
  - client node status 657
  - client node storage 657
  - database 657
  - node activity 657
  - schedule 657
  - server status 657
  - storage device 657
  - storage pool 657
  - tape usage 657
  - tape volume 657
  - Tivoli Enterprise Portal
    - monitoring workspaces 668
- mount
  - count of number of times per volume 337
  - library 202
  - limit 202
  - mode 152
  - operations 151
  - query 154
  - retention period 203
  - wait period 204
- mount point
  - preemption 566
  - queue, server option 67
  - relationship to mount limit in a device class 202, 210
  - requirements for simultaneous-write operations 305

- mount point (*continued*)
  - settings for a client session 369
- MOVE DATA command 350
- MOVE DRMEDIA command 818
- MOVE NODEDATA 354
- moving a backup set
  - benefits of 508
  - to another server 508
- moving data 410
  - from off-site volume in a copy storage pool 351
  - monitoring the movement of 353
  - procedure 351
  - requesting processing information 352
  - to another storage pool 350
  - to other volumes in same storage pool 350
- multipath I/O 85
- multiple
  - copy storage pools, restoring from 765
  - managing IBM Tivoli Storage Manager servers 35
  - managing Tivoli Storage Manager servers 671
- multiple commands
  - backup and restore 518
- multiple server instances 560
- multiple servers 707
- multiple sessions
  - on clients for a restore 519

## N

- name of device 87
- NAS file server, NDMP operations
  - backing up a NAS file server 189
  - backing up a NAS file server to native pools 190, 191
  - configuration checklist 173
  - data format 170
  - data mover, description 49, 50, 128
  - defining a data mover 128, 186
  - defining a path for data mover and a library 187
  - defining a storage pool 179
  - defining a tape drive 186
  - differential image backup, description 57
  - full image backup, description 57
  - interfaces used with 169
  - managing NAS nodes 170
  - path, description 49, 50, 129
  - planning 177
  - policy configuration 174, 485
  - registering a NAS node 185, 370
  - requirements for set up 167
  - restoring a NAS file server 189
  - scheduling a backup 188
  - storage pools for NDMP operations 179
- NAS node
  - defining 185
  - deleting 171
  - registering 185
  - renaming 171
- NATIVE data format 170
- NDMP
  - operations 170
- NDMP operations 197
- NDMP operations for Celerra data movers 197
- NDMP operations for NAS file servers
  - backing up a NAS file server 189
  - backing up a NAS file server to native pools 190, 191
  - configuration checklist 173
  - data format 170

- NDMP operations for NAS file servers *(continued)*
  - data mover, description 49, 50, 128
  - defining a data mover 128, 186
  - defining a path for data mover and a library 187
  - defining a storage pool 179
  - defining a tape drive 186
  - differential image backup, description 57
  - full image backup, description 57
  - interfaces used with 169
  - managing NAS nodes 170
  - path, description 49, 50, 129
  - planning 177
  - policy configuration 174, 485
  - registering a NAS node 185, 370
  - requirements for set up 167
  - restoring a NAS file server 189
  - scheduling a backup 188
  - storage pools for NDMP operations 179
- NetApp file server
  - data format for backup 170
  - international characters 194
- NETAPPDUMP data format 170, 179
- Network Appliance file server
  - backup methods 175
  - requirements 167
  - storage pool for backup 231
  - tape device for backup 167
  - using NDMP operations 55, 167
- network attached storage
  - virtual file spaces 188
- network of IBM Tivoli Storage Manager servers 35
- network of Tivoli Storage Manager servers 671
- network-attached nodes
  - comparing to local nodes 374
- network-attached storage (NAS) file server
  - backup methods 175
  - registering a NAS node for 185
  - using NDMP operations 55, 167
- next storage pool
  - definition 241
  - deleting 361
  - migration to 252, 342
- no query restore 517
- node
  - registering 396, 438
- node privilege class
  - description of 394
  - granting 395
- node, client
  - adding 367
  - agent 390
  - amount of space used 346
  - creating backup sets for 503
  - file spaces, QUERY OCCUPANCY command 346
  - finding tapes used by 341
  - immediate processing 543
  - importing 746
  - locking 388
  - managing registration 368, 377, 555
  - options file 372
  - performing operations for 495, 531, 537
  - privilege class for scheduling operations for 526
  - proxy node relationships 389
  - querying 392
  - reducing archive packages for 522
  - registering 371
  - removing 389

- node, client *(continued)*
  - renaming 388
  - scheduling operations for 526
  - setting password authentication 433
  - setting scheduling mode 539
  - setting up subfile backups 512
  - target 390
  - unlocking 388
  - updating 380
  - viewing information about 392
- nodes
  - moving nodes from schedule 534
  - overview of client and server 367
- NOPREEMPT server option 566
- NORETRIEVEDATE server option 264
- NOTIFY SUBSCRIBERS command 695
- number of times mounted, definition 337

## O

- occupancy, querying 345
- off-site recovery media (for DRM)
  - volumes
    - sending off-site 816
- off-site volume
  - limiting the number to be reclaimed 328
- off-site volumes
  - moving data in a copy storage pool 351
- offsite recovery media (for DRM)
  - volumes
    - moving back on-site 817
    - states 815
- offsite volume access mode 240
- offsite volumes
  - limiting the number to be reclaimed 227
- one-drive library, volume reclamation 227, 323
- open registration
  - description 368
  - enabling 375
  - process 369
  - setting 368
- operations available to client 10
- operator privilege class
  - reducing 432
  - revoking 432
- optical device
  - defining device class 199
- option set, client
  - adding client options to 412
  - assigning clients to 413
  - copying 413
  - creating 411
  - deleting 413
  - deleting an option from 413
  - for NAS node 175
  - requesting information about 413
  - updating description for 413
- option, server
  - 3494SHARED 66
  - ACSL options 66
  - ASSISTVCRRECOVERY 67
  - AUDITSTORAGEstorage audit 557
  - changing with SETOPT command 569
  - COMMTIMEOUTcommunication timeout 415, 416
  - DEVCONFIG 770
  - DRIVEACQUIRERETRY 67
  - EXPINTERVAL 473

- option, server *(continued)*
  - EXPQUIET 473
  - IDLETIMEOUTidle timeout 415, 416, 615
  - NOPREEMPT 67, 566
  - NORETRIEVEDATEfile retrieval date 264
  - overview 20
  - QUERYAUTH 425
  - REQSYSAUTHOUTFILE 425
  - RESCOURCETIMEOUT 67
  - RESTOREINTERVALrestore interval 417, 439, 473
  - SEARCHMPQUEUE 67
  - THROUGHPUTDATATHRESHOLD 416
  - THROUGHPUTTIMETHRESHOLD 416
  - TXNGROUPMAXmaximum transaction group size 243
  - VOLUMEHISTORY 769
- options file, client 372
- options, querying
  - VIRTUALMOUNTPOINT client option 398
- overflow location 227
- owner authority, client 394, 396

## P

- PARALLEL command 581
- passthru driver 81
- password
  - changing the key database 423, 424
  - resetting an administrative 428
  - setting authentication for a client 433
  - setting expiration 432
  - setting invalid limit 433
  - setting minimum length 433
- path
  - defining 129
  - description 49, 50, 177
  - to library 187
- paths
  - defining 126
- pending, volume state 337
- performance
  - cache, considerations for using 77, 263
  - clients, optimizing restore 223, 514
  - concurrent client/server operation considerations 540
  - data protection, increasing with simultaneous-write function 285
  - data validation for nodes 496
  - data validation for storage pools 788
  - database read, increase with mirroring 759
  - file system effects on 75
    - random-access disk (DISK) 69, 236
  - FILE-type volumes, backing up many small objects to 208
  - fragmentation, private FILE volumes for reducing disk 46
  - migration, multiple concurrent processes 227, 261
  - mobile client 511
  - reclamation, multiple concurrent processes
    - copy storage pools 325
    - primary sequential access storage pools 227, 322
  - storage pool backup, reducing time required for 285
  - storage pool volume 258, 779
  - ulimits, setting for optimal performance 19
  - volume frequently used, improve with longer mount retention 203
- period, specifying for an incremental backup 539
- point-in-time restore
  - enable for clients 9, 488
  - for database 774

- policy
  - default 5, 437
  - deleting 492
  - description of 443
  - distributing with enterprise management 489
  - effect of changing 470, 471
  - for application clients 483
  - for clients using SAN devices 486
  - for direct-to-tape backup 482
  - for logical volume backups 483
  - for NAS file server node 174
  - for point-in-time restore 488
  - for server as client 488
  - for space management 437, 455, 461
  - importing 742
  - managing 435
  - operations controlled by 440
  - planning 436
  - querying 489
- policy domain
  - active-data pools, specifying 458
  - associating a client with 29
  - changing 439
  - creating 460
  - define 486
  - deleting 493
  - description of 443
  - distributing via profile 489, 688
  - for NAS file server node 174
  - querying 491
  - updating 456, 458
- policy privilege class
  - description 425, 430
  - granting 431
  - revoking 431, 432
- policy set
  - activating 471
  - changing, via the active policy set 439
  - copying 439, 456, 460
  - defining 460
  - deleting 493
  - description of 443
  - querying 491
  - updating 460
  - validating 470, 472
- pool, storage
  - 3592, special considerations for 204
  - active-data pool 223
  - amount of space used 347
  - auditing a volume 782
  - backup
    - full 760
    - incremental 760
  - backup and recovery 760
  - comparing primary and copy types 359
  - copy 223
  - creating a hierarchy 241
  - data format 170, 227, 231
  - defining 227
  - defining a copy storage pool 357
  - defining for disk, example 231, 242
  - defining for NDMP operations 179
  - defining for tape, example 231, 242
  - deleting 361
  - description of 222
  - destination in copy group 462, 468
  - determining access mode 227, 358

- pool, storage (*continued*)
    - determining maximum file size 227
    - determining whether to use collocation 227, 310, 358
    - disk 25
    - duplicate, using to restore 765
    - enabling cache for disk 227, 263
    - estimating space for archived files on disk 331
    - estimating space for backed up files on disk 331
    - estimating space for disk 330
    - estimating space for sequential 332
    - estimating space in multiple 241
    - incomplete, using to restore 765
    - increasing sizes 25
    - LTO Ultrium, special considerations for 211
    - managing 221
    - monitoring 332
    - moving files 350
    - moving files between 350
    - multiple, using to restore 765
    - next storage pool
      - definition 241
      - deleting 361
      - migration to 252, 342
    - overview 48
    - policy use 462, 468
    - primary 222
    - querying 332
    - renaming 357
    - restoring 763, 794
    - search-and-selection order for active files 225
    - simultaneous-write function 285
    - updating 227
    - updating for disk, example 232, 243
    - using cache on disk 227, 263
    - validation of data 784
    - viewing information about 332
  - portable media
    - description of 6, 8, 502
    - restoring from 507
  - preemption
    - mount point 566
    - volume access 566
  - prefix, for recovery instructions 804
  - prefix, for recovery plan file 804
  - prefix, server 708
  - premigration 442
  - PREPARE command 810
  - PREVIEW parameter 732, 740
  - primary volumes unavailable for restore 765
  - private category, 349X library 99
  - private volumes 50
  - privilege class, administrator
    - description 425
    - granting authority 425
    - reducing 432
    - revoking all 432
  - privilege class, policy
    - description 425, 430
    - granting 431
    - revoking 431, 432
  - privilege class, storage
    - description 425
    - granting 431
    - reducing 431
    - revoking 432
  - problem determination
    - describing problem for IBM Software Support xviii
    - problem determination (*continued*)
      - determining business impact for IBM Software Support xviii
      - migration 252
      - submitting a problem to IBM Software xix
  - process
    - background 564
    - canceling 565
    - drive clean error checking 163
    - expiration 766
    - number for migration 227, 254
    - number for storage pool backup 762
    - number for storage pool restore 778
    - reclamation 319, 327
  - profile
    - associating configuration information with 690
    - changing 690, 694, 695
    - default 693, 700
    - defining 690, 691
    - deleting 695, 696
    - description 690
    - getting information about 697
    - locking 694
    - problems with synchronization 705
    - unlocking 694
  - progressive incremental backup 13
  - protecting your data 36, 141, 755
    - active-data pools 223, 760
    - data deduplication 275
    - simultaneous-write function 285
  - protection options
    - client 8
    - server 14, 37
  - proxy node relationships 391
  - publications
    - download xiii
    - order xiii
    - related hardware xv
    - search xiii
    - Tivoli Storage Manager xiii
- ## Q
- query
    - authority 425
    - for general information 335
    - policy objects 489
  - QUERY ACTLOG command 624, 750
  - QUERY ADMIN command 429
  - query association output 534
  - QUERY BACKUPSETCONTENTS command 510
  - QUERY CONTENT command 338
  - QUERY COPYGROUP command 490, 744
  - QUERY DB command 603
  - QUERY DBSPACE command 603
  - QUERY DEVCLASS command 733
  - QUERY DIRSPACE 348
  - QUERY DOMAIN command 491
  - QUERY DRIVE command 156
  - QUERY DRMSTATUS command 801
  - QUERY ENABLED command 640
  - QUERY EVENT command 535, 577
  - QUERY FILESPACE command 397
  - QUERY LIBRARY command 155
  - QUERY LICENSE command 557
  - QUERY MGMTCLASS command 490
  - QUERY MOUNT command 154

- QUERY NODE command 392
- QUERY OCCUPANCY command 345, 346, 347, 348
- QUERY OPTION command 616
- QUERY POLICYSET command 491
- QUERY PROCESS command 565, 615, 748
  - identification numbers of migration processes 343
  - information about data movement process 352
- QUERY REQUEST command 152
- QUERY RESTORE command 418
- QUERY RPFCONTENT command 812
- QUERY RPFFILE command 812
- QUERY SCHEDULE command 528
- QUERY SCRIPT command 585
- QUERY SERVERGROUP command 711
- QUERY SESSION command 414, 614
- QUERY SHREDSTATUS command 500
- QUERY STATUS command 616
- QUERY STGPOOL command 332, 342, 344
- QUERY SUBSCRIPTION command 701
- QUERY SYSTEM command 617
- QUERY VOLHISTORY command 769
- QUERY VOLUME command 335, 353
- QUERYAUTH server option 425

## R

- random mode for libraries 80
- randomize, description of 540
- raw logical volume 75
- read-only access mode 240
- read/write access mode 239
- real-time monitoring 668
- rebinding
  - description of 451
  - file to a management class 451
- recalling a file
  - selective 442
  - transparent 442
- receiver 626
- RECLAIM STGPOOL command 322
- reclamation 328
  - active-data pools 325
  - aggregate reconstruction 319
  - controlling duration 322
  - delayed start of process 319
  - delaying reuse of volumes 329, 766
  - description of 25
  - effects of collocation 329
  - effects of DELETE FILESPACE 319
  - multiple concurrent processes
    - copy storage pools 325
    - primary sequential access storage pools 227, 322
  - off-site volume
    - controlling when reclamation occurs 326
    - setting a threshold for sequential storage pool 227, 319, 358
    - starting reclamation manually 322
    - storage pool for 227
    - virtual volumes 324
    - with single drive 323
- RECONCILE VOLUMES command 718
- reconstructing aggregates 319, 326, 353
- recovering storage pools 760
- recovering the database 772
- recovery instructions file 841
- recovery log 593, 596
  - active log 595, 596, 598
- recovery log (*continued*)
  - active log mirror 596, 601
  - active log space 601
  - archive failover log 595, 598
  - archive failover log space 602
  - archive log 595, 597, 598
  - archive log space 602
  - description of 34, 593
  - estimating space requirements 601
  - increasing the size 605
  - log mirror 595, 596, 598, 601
  - managing 593
  - monitoring 603
  - out of space 605
  - when to backup 755, 768
- recovery logs
  - move to another directory 608
  - relocating on a server 607
- recovery plan file
  - break out stanzas 836
    - using VBScript procedure 836
  - creating 810
  - example 838
  - prefix 804
  - stanzas 836
- recovery, disaster
  - auditing storage pool volumes 791
  - example recovery procedures 792
  - general strategy 712, 755
  - media 810
  - methods 712, 755
  - providing 712, 755
  - when to backup 755, 768
- REGISTER ADMIN command 428
- REGISTER LICENSE command 556
- REGISTER NODE command 396
- registering
  - client option sets 370
  - workstation 371
- registration
  - description of 368
  - licensing for a client node 555
  - licensing for an administrator 555
  - managing client node 368, 377
  - setting for a client node 368
  - source server 371
- relationships
  - among clients, storage, and policy 444
- remote access to clients 393
- removable file system device
  - labeling requirements 119
  - REMOVABLEFILE device type, defining and updating 207
    - support for 118, 207
- removable media 45
- REMOVE ADMIN command 429
- REMOVE NODE command 389
- RENAME ADMIN command 429
- RENAME FILESPACE command 748
- RENAME NODE command 388
- RENAME SCRIPT command 586
- RENAME SERVERGROUP command 711
- RENAME STGPOOL command 357
- renamed file spaces 408
- renaming
  - administrator ID 429
  - NAS node 171

- renaming (*continued*)
  - storage pool 357
- reporting 647
  - custom 647
  - modifying for performance 669
  - modifying queries 669
- reporting ANR9999D messages 628
- reports
  - client activity 650
  - historical reports
    - client activity 650
  - server trends 654
  - viewing reports 667
- REQSYSAUTHOUTFILE server option 425
- resetting
  - administrative password 428
  - user password expiration 432
- RESTORE DB command 562
- RESTORE STGPOOL command 763, 794
- restore to point-in-time, enabling for clients 488
- RESTORE VOLUME command 796
- RESTOREINTERVAL server option
  - restore interval for restartable restore sessions 417, 439, 473
- restoring
  - clients, optimizing restore 223, 514
  - database
    - point-in-time 772
    - to its most current state 776
  - file 440
  - storage pools and volumes from active-data pools 777, 780
  - storage pools with incomplete volumes 765
- restoring image data
  - from backup sets 508
- restriction
  - ASCII characters in administrative Web interface 578
  - drive cleaning 160
  - serial number detection 131
- retain extra versions, description of 437, 464
- retain only version, description of 437, 464
- retaining data using Centera
  - unsupported functions 231
- retention grace period
  - description of archive 458
  - description of backup 458
  - for backup sets 505
  - using archive 458
  - using backup 458
- RETEXTTRA parameter 437, 464
- REONLY parameter 437, 464
- retrieval date for files 264
- retrieval from archive
  - archive package 520
  - file 441
- reuse of sequential volumes
  - delaying 329, 766

- reuse of sequential volumes (*continued*)
  - storage pool volumes 144
  - volume pending state 337
- roll-forward recovery 596
  - database backup trigger 776
  - mirroring recovery log 776
  - recovery log 776
- ROLLBACK command 591
- routing commands to servers 707
- RUN command 586

## S

- SAN (storage area network)
  - client access to devices 54
  - device changes, detecting 131
  - LAN-free data movement 54
  - NDMP operations 55, 167
  - policy for clients using LAN-free data movement 486
  - sharing a library among servers 52, 95, 105
  - storage agent role 54
- scale capacity 205
- scenarios
  - policy configuration 482
- schedule
  - administrative command 571
  - associating client node with 527
  - checking the log file 536
  - coordinating 537
  - copying 532, 576
  - day of the week 573
  - defining 526, 567, 573
  - deleting 532, 576
  - description of 525
  - expiration date 574
  - failed, querying 528, 535
  - for NAS file server backup 188
  - frequency of service 574
  - initial start date 573
  - initial time 573
  - maintenance 581
  - mode, setting 538
  - priority 574
  - querying 528
  - results of 535, 577
  - server administrative command 572
  - startup window 539, 574
  - type of action 575
  - uncertain status 536, 577
  - updating 573
  - viewing information about 528
- schedule event
  - managing 534, 576
  - querying 535, 577
  - viewing information about 535, 577
- scheduled operations, setting the maximum 540
- scheduler workload, controlling 539
- scheduling
  - administrative commands 34
  - client 32
- scheduling mode
  - client-polling 538
  - overview of 538
  - selecting 538
  - server-prompted 538
  - setting on a client node 539
  - setting on the server 538

- scratch category, 349X library 99
- scratch volume
  - deleting 235, 769
  - description 50
  - FILE volumes 78
  - number allowed in a storage pool 227, 358
  - using in storage pools 238
- script
  - maintenance 581
- script, scheduling on client 529
- script, server
  - continuation characters 582
  - copying 585
  - defining 578
  - deleting 586
  - EXIT statement 583
  - GOTO statement 583
  - IF clause 583
  - querying 585
  - renaming 586
  - routing commands in 708
  - running 586
  - running commands in parallel 581
  - running commands serially 581
  - substitution variables 582
  - updating 584
  - used with SNMP 634
  - Web browser, restricted to ASCII entry 578
- SCSI
  - automatic labeling of volumes 136
  - library with different tape technologies 204
- SCSI library
  - connect to NAS file server 182
  - connecting to the server 181
- SCSI tape library
  - setting up for NDMP operations 179
- SEARCHMPQUEUE server option 67
- Secure Sockets Layer
  - adding certificates 423, 424
  - Administration Center 425
  - communication using 421
  - digital certificate file protection 760
  - encryption key 760
  - Global Security Kit 421
  - specifying communication ports 422
- security
  - client access, controlling 395
  - data encryption
    - 3592 generation 2 496
    - 3592 Generation 2 158, 206
    - 3592 generation 3 496
    - 3592 Generation 3 158
    - ECARTRIDGE 216
    - IBM LTO Generation 4 158, 213, 496
    - Sun StorageTek T10000B 158, 496
  - features, overview 27
  - for the server 421
  - locking and unlocking administrators 430
  - locking and unlocking nodes 388
  - managing access 421, 427
  - password expiration for nodes 432
  - privilege class authority for administrators 425
  - server options 425
- SELECT command 618
  - customizing queries 619
- selective backup 440, 454
- selective recall 442
- sending commands to servers 707
- sequence number 412, 413
- sequential mode for libraries 80
- sequential storage pool
  - auditing a single volume in 789
  - auditing multiple volumes in 789
  - collocation 316
  - estimating space 332
  - migration threshold 258
  - reclamation 319
- SERIAL command 581
- serial number
  - automatic detection by the server 126, 127, 129, 131
  - for a drive 127
  - for a library 126, 127, 129
- serialization parameter 437, 462, 468
- server
  - backing up subfiles on 511
  - canceling process 565
  - changing the date and time 563
  - console, MMC snap-in 19
  - deleting 684
  - description of 3
  - disabling access 417
  - disaster recovery 37
  - enabling access 417
  - halting 561
  - importing subfiles from 513
  - instances
    - multiple on single system 560
    - owner ID 560
  - maintaining, overview 19
  - managing multiple 35
  - managing operations 555
  - managing processes 564
  - messages 628
  - Microsoft Management Console snap-in 19
  - multiple instances 560
  - network of IBM Tivoli Storage Manager 35
  - network of Tivoli Storage Manager servers 671
  - options
    - adding or updating 569
    - prefix 708
    - protecting 37
    - querying about processes 565, 615
    - querying options 616
    - querying status 616
    - running multiple servers 560
    - setting the server name 567
    - starting 557, 558
    - stopping 561
    - updating 684
    - utilities 19
    - viewing information about 616
    - viewing information about processes 565, 615
    - wizards 19
- Server
  - starting 559
- server console
  - logging events to 628
- server console, description of 425
- SERVER device type 223, 712
- server group
  - copying 711
  - defining 710
  - deleting 711
  - member, deleting 712

- server group *(continued)*
  - moving a member 712
  - querying 711
  - renaming 711
  - updating description 711
- server instances
  - owner 560
- server option
  - 3494SHARED 66
  - ACSLs options 66
  - ACTIVELOGDIR 605, 609
  - ACTIVELOGSIZE 605
  - ASSISTVCRRECOVERY 67
  - AUDITSTORAGEstorage audit 557
  - changing with SETOPT command 569
  - COMMTIMEOUTcommunication timeout 415, 416
  - DEVCONFIG 770
  - DRIVEACQUIRERETRY 67
  - EXPINTERVAL 473
  - EXPQUIET 473
  - IDLETIMEOUTidle timeout 415, 416, 615
  - NOPREEMPT 67, 566
  - NORETRIEVEDATEfile retrieval date 264
  - overview 20
  - QUERYAUTH 425
  - REQSYSAUTHOUTFILE 425
  - RESCOURCETIMEOUT 67
  - RESTOREINTERVALrestore interval 417, 439, 473
  - SEARCHMPQUEUE 67
  - THROUGHPUTDATATHRESHOLD 416
  - THROUGHPUTTIMETHRESHOLD 416
  - TXNGROUPMAXmaximum transaction group size 243
  - VOLUMEHISTORY 769
- server options 568
  - ARCHFAILOVERLOGDIR 610
  - DBMEMPERCENT 606
  - MIRRORLOGDIR 610
  - TECUTF8EVENT 632
- server script
  - continuation characters 582
  - copying 585
  - defining 578
  - deleting 586
  - EXIT statement 583
  - GOTO statement 583
  - IF clause 583
  - querying 585
  - renaming 586
  - routing commands in 708
  - running 586
  - running commands in parallel 581
  - running commands serially 581
  - substitution variables 582
  - updating 584
  - used with SNMP 634
  - Web browser, restricted to ASCII entry 578
- server scripts 578
- server session
  - states 415
- server storage
  - active-data pools 223
  - client files, process for storing 5
  - concepts overview 15
  - considering user needs for recovering 64
  - deleting files from 361
  - example 225
  - managing 21
- server storage *(continued)*
  - monitoring 782
  - planning 65
  - protection, methods 755
  - server options affecting 66
  - tailoring definitions 744
  - using another IBM Tivoli Storage Manager server 712
  - using disk devices 69
  - using the storage hierarchy 251
- server trends reports 654
- server-free data movement
  - not available in Administration Center 550
  - suggested usage 10
- server-prompted scheduling 538
- server-to-server communications, establishing
  - enterprise configuration 676
  - enterprise event logging 676
  - virtual volumes 684
- server-to-server virtual volumes
  - deduplication 712
  - reclaiming 324
  - using to store data 712
- session
  - canceling 415
  - negative number 418
  - server-initiated 379
  - setting the maximum percentage for scheduled operations 540
- session, client
  - canceling 415
  - DSMC loop 414
  - held volume 414
  - managing 414
  - querying 414, 614
  - viewing information about 414, 614
- SET ACCOUNTING command 645
- SET ACTLOGRETENTION command 625
- SET AUTHENTICATION command 433
- SET CLIENTACTDURATION command 544
- SET CONFIGMANAGER command 687, 689
- SET CONFIGREFRESH command 701
- SET CONTEXTMESSAGING command 628
- SET CROSSDEFINE command 679, 682
- SET DBRECOVERY command 606
- SET DBREPORTMODE command 603
- SET DRMCHECKLABEL command 806
- SET DRMCOPYSTGPOOL command 802
- SET DRMCOURIERNAME command 805
- SET DRMDBBACKUPEXPIREDAYS command 806
- SET DRMFILEPROCESS command 805, 806
- SET DRMINSTPREFIX command 804
- SET DRMNOTMOUNTABLE command 805
- SET DRMPPLANPOSTFIX command 803
- SET DRMPPLANPREFIX command 804
- SET DRMPRIMSTGPOOL command 802
- SET DRMRPFEXPIREDAYS 812
- SET DRMVaultNAME command 806
- SET EVENTRETENTION command 536, 577
- SET INVALIDPWLIMIT command 433
- SET LICENSEAUDITPERIOD command 557
- SET MAXCMDRETRIES command 542
- SET MAXSCHEDULESESSIONS command 540
- SET PASSEXP command 432
- SET QUERYSCHEDPERIOD command 542
- SET RANDOMIZE command 540
- SET REGISTRATION command 368
- SET RETRYPERIOD command 543

- SET SCHEDMODES command 538
- SET SERVERHLADDRESS command 679, 682
- SET SERVERLLADDRESS command 679, 682
- SET SERVERNAME command 567, 678, 679, 682
- SET SERVERPASSWORD 678, 679, 682
- SET SUBFILE 511
- SET SUMMARYRETENTION 622
- SETOPT command 569
- setting
  - clients to use subfile backup 512
  - compression 370
  - library mode 80
  - password 432
  - time interval for checking in volumes 204
- shared access, nodes 391
- shared dynamic serialization, description of 462, 468
- shared file system 76
- shared library 95, 105
- shared static serialization, description of 462, 468
- SHRED DATA command 500
- shredding
  - BACKUP STGPOOL command 501, 761
  - COPY ACTIVATEDATA command 501
  - DEFINE STGPOOL command 501
  - DELETE FILESPACE, command 501
  - DELETE VOLUME, command 501
  - deleting empty volumes 362
  - deleting volumes with data 362
  - description 499
  - enforcing 501
  - EXPIRE INVENTORY command 501
  - EXPORT NODE command 501, 722
  - EXPORT SERVER command 501, 722
  - GENERATE BACKUPSET command 501, 502
  - MOVE DATA command 350, 501
  - setting up 500
  - UPDATE STGPOOL command 501
- SHREDDING server option 500
- simultaneous-write operations to primary and copy storage pools
  - drives 306, 307
  - inheritance model 292
  - mount points 305
  - storage pools 307
- SnapLock
  - data protection, ensuring 481
  - event-based retention 480
  - reclamation 477
  - retention periods 477
  - WORM FILE volumes, setting up 481
- SnapMirror to Tape 196
- snapshot, using in backup 9, 11
  - using in directory-level backups 195
- SNMP
  - agent 636
  - communications 636, 637
  - configuring 637
  - enabled as a receiver 626, 634
  - heartbeat monitor 626, 634
  - manager 636
  - subagent 636
- Software Support
  - contact xvii
  - describing problem for IBM Software Support xviii
  - determining business impact for IBM Software Support xviii
- Sony WORM media (AIT50 and AIT100) 141
- source server 714
- space
  - directories associated with FILE-type device classes 348
- space-managed file 441
- special file names 87
- SQL 618
- SQL activity summary table 622
- SSL
  - key certificate 423, 424
- SSLTCPADMINPORT
  - server option 422
- SSLTCPPOINT
  - server option 422
- stand-alone mode 559
- standard label 23
- standard management class, copying 461
- standard storage management policies, using 437
- start time, randomizing for a schedule 540
- starting the server 557, 558, 560
- startup window, description of 540
- static serialization, description of 462, 468
- status of a volume in an automated library 51
- stopping the server 561
- storage agent 54
- storage area network (SAN)
  - client access to devices 54
  - device changes, detecting 131
  - LAN-free data movement 54
  - NDMP operations 55, 167
  - policy for clients using LAN-free data movement 486
  - sharing a library among servers 52, 95, 105
  - storage agent role 54
- storage devices 89, 199, 200
- storage hierarchy 24
  - copying active backup data 223
  - defining in reverse order 231, 242
  - establishing 241
  - example 225
  - for LAN-free data movement 241
  - how the server stores files in 243
  - next storage pool
    - definition 241
    - deleting 361
    - migration to 252, 342
  - restrictions 241
  - staging data on disk for tape storage 251
- storage management policies
  - description of 28, 443
  - managing 435
  - tailoring 456
  - using standard 437
- storage occupancy, querying 345
- storage pool
  - 3592, special considerations for 204
  - active-data pool 223
  - amount of space used 347
  - auditing a volume 782
  - backup
    - full 760
    - incremental 760
  - backup and recovery 760
  - comparing primary and copy types 359
  - copy 223
  - creating a hierarchy 241
  - data format 170, 227, 231
  - defining 227
  - defining a copy storage pool 357

- storage pool (*continued*)
    - defining for disk, example 231, 242
    - defining for NDMP operations 179
    - defining for tape, example 231, 242
    - deleting 361
    - description of 222
    - destination in copy group 462, 468
    - determining access mode 227, 358
    - determining maximum file size 227
    - determining whether to use collocation 227, 310, 358
    - disk 25
    - duplicate, using to restore 765
    - enabling cache for disk 227, 263
    - estimating space for archived files on disk 331
    - estimating space for backed up files on disk 331
    - estimating space for disk 330
    - estimating space for sequential 332
    - estimating space in multiple 241
    - incomplete, using to restore 765
    - increasing sizes 25
    - LTO Ultrium, special considerations for 211
    - managing 221
    - monitoring 332
    - moving files 350
    - moving files between 350
    - multiple, using to restore 765
    - next storage pool
      - definition 241
      - deleting 361
      - migration to 252, 342
    - overview 48
    - policy use 462, 468
    - primary 222
    - querying 332
    - renaming 357
    - restoring 763, 794
    - search-and-selection order for active files 225
    - simultaneous-write function 285
    - updating 227
    - updating for disk, example 232, 243
    - using cache on disk 227, 263
    - validation of data 784
    - viewing information about 332
  - storage pool backups
    - one copy storage pool
      - scheduling 764
    - scheduling 764
  - storage privilege class
    - description 425
    - granting 431
    - reducing 431
    - revoking 432
  - storage volume
    - auditing 782
    - contents 338
    - formatting random access 75, 236
    - information about 335
    - labeling sequential access 133, 237
    - monitoring use 335
    - overview 48
    - preparing sequential access 133, 237
  - StorageTek devices 215
  - stub file 442
  - subfile backups
    - deleting 513
    - description of 511
    - example of 511
  - subfile backups (*continued*)
    - expiring 513
    - managing 513
    - restoring 513
  - subordinate storage pool 241
  - subscriber, deleting 706
  - subscription
    - defining 699, 701
    - deleting 702
    - scenario 700
  - substitution variables, using 582
  - support information xv
  - supported devices 42
  - swapping volumes in automated library 140
  - system catalog tables 618
  - system privilege class
    - revoking 432
- T**
- table of contents 192
    - generating for a backup set 509
    - managing 172, 193
  - tape
    - capacity 217
    - exporting data 734
    - finding for client node 341
    - label prefix 23
    - monitoring life 337
    - number of times mounted 337
    - planning for exporting data 733
    - recording format 201
    - reuse in storage pools 144
    - rotation 60, 144
    - scratch, determining use 227, 238, 358
    - setting mount retention period 203
    - volumes
      - initializing 23
      - labeling 23
  - target server 714
  - TCP/IP 378
    - IPv4 378
    - IPv6 378
  - TECUTF8EVENT option 632
  - text editor
    - to work with client 373
  - threshold
    - migration, for storage pool
      - random access 254
      - sequential access 259
    - reclamation 227, 319, 358
  - THROUGHPUTDATATHRESHOLD server option 416
  - THROUGHPUTTIMETHRESHOLD server option 416
  - time interval, setting for checking in volumes 204
  - timeout
    - client session 416
  - Tivoli Data Warehouse 647
  - Tivoli Enterprise Console 630
    - setting up as a receiver 632
  - Tivoli Enterprise Portal workspaces 657
  - Tivoli event console 626, 630
  - Tivoli Integrated Portal
    - configuring SSL 425
  - Tivoli Storage Manager 667, 668
    - server network 671
  - Tivoli Storage Manager definitions 613
  - Tivoli Storage Manager device drivers 81

- Tivoli Storage Manager for Space Management 461
  - archive policy, relationship to 456
  - backup policy, relationship to 456
  - description 441
  - files, destination for 461
  - migration of client files
    - description 442
    - eligibility 455
  - policy for, setting 455, 461
  - premigration 442
  - recall of migrated files 442
  - reconciliation between client and server 442
  - selective migration 442
  - setting policy for 456, 461
  - simultaneous-write function, version support for 287
  - space-managed file, definition 441
  - stub file 442
- Tivoli technical training xv
- training, Tivoli technical xv
- transactions, database 593, 610
- transparent recall 442
- troubleshooting
  - errors in database with external media manager 122
- TXNBYTELIMIT client option 243
- TXNGROUPMAX server option 243
- type, device
  - 3570 199, 200
  - 3590 200
  - 4MM 199, 200
  - 8MM 199, 200
  - CARTRIDGE 200
  - DISK 199
  - DLT 199, 200
  - DTF 199, 200
  - ECARTRIDGE 200
  - FILE 199
  - GENERICTAPE 199, 200
  - LTO 200, 211
  - multiple in a single library 58
  - OPTICAL 199
  - QIC 199, 200
  - REMOVABLEFILE 199
  - SERVER 199, 200, 714, 716
  - VOLSAFE 215
  - WORM 199, 200
  - WORM12 200
  - WORM14 200
- typographic conventions xix

## U

- ulimits 19
- Ultrium, LTO device type
  - device class, defining and updating 211
  - encryption 158, 213, 496
  - WORM 141, 215
- unavailable access mode
  - description 240
  - marked by server 154
  - marked with PERMANENT parameter 153
- uncertain, schedule status 536, 577
- Unicode
  - automatically renaming file space 402
  - client platforms supported 400
  - clients and existing backup sets 408
  - deciding which clients need enabled file spaces 401
  - description of 400

- Unicode (*continued*)
  - displaying Unicode-enabled file spaces 408
  - example of migration process 407
  - file space identifier (FSID) 408, 409
  - how clients are affected by migration 406
  - how file spaces are automatically renamed 404
  - migrating client file spaces 401
  - options for automatically renaming file spaces 402
- Unicode versions
  - planning for 404
- UNIQUETDPTECEVENTS option 630
- UNIQUETECEVENTS option 630
- UNLOCK ADMIN command 430
- UNLOCK NODE command 388
- UNLOCK PROFILE command 694, 695
- unplanned shutdown 561
- unreadable files 757, 790, 791
- UPDATE ADMIN command 428
- UPDATE ARCHIVE command 522
- UPDATE BACKUPSET command 510
- UPDATE CLIENTOPT command 413
- UPDATE CLOPTSET command 413
- UPDATE COPYGROUP command 462, 468
- UPDATE DEVCLASS command 200
- UPDATE DOMAIN command 460
- UPDATE DRIVE command 157
- UPDATE LIBRARY command 155
- UPDATE LIBVOLUME command 50, 147
- UPDATE MGMTCLASS command 461
- UPDATE NODE command 380, 407, 411
- UPDATE POLICYSET command 460
- UPDATE RECOVERYMEDIA command 810
- UPDATE SCHEDULE command 573
- UPDATE SCRIPT command 584
- UPDATE SERVER command 684
- UPDATE VOLUME command 237
- URL for client node 368
- user exit 626
- user exit declarations 641, 871
- user ID, administrative
  - creating automatically 396
  - description of 368
  - preventing automatic creation of 396
- user-exit program 643, 873
- utilities, for server 19

## V

- validate
  - node data 496
- VALIDATE LANFREE command 125
- VALIDATE POLICYSET command 470
- validating data
  - during a client session 495
  - for storage pool volumes 784
  - for virtual volumes 713
  - performance considerations for nodes 496
  - performance considerations for storage pools 788
- variable, accounting log 645
- VARY command 77
- varying volumes on or off line 77
- VERDELETED parameter 437, 464
- VEREXISTS parameter 437, 464
- versions data deleted, description of 437, 464
- versions data exists, description of 437, 464
- virtual file space mapping, command 195

- virtual volume
  - performance expectations 715
- virtual volumes, server-to-server
  - deduplication 712
  - reclaiming 324
  - using to store data 712
- VIRTUALMOUNTPOINT client option 398
- Vital Cartridge Records (VCR), corrupted condition 67
- VOLSAFE device class 215
- volume capacity 201
- volume history
  - deleting information from 769
  - file, establishing 768
  - for database restore 605
  - using backup to restore database 768, 774
- volume history file 78
- volume reuse 78
- VOLUMEHISTORY option 768
- volumes
  - access preemption 566
  - access, controlling 143
  - allocating space for disk 75, 236
  - assigning to storage pool 236
  - auditing 149, 782
  - auditing considerations 782
  - automated library inventory 51
  - capacity, compression effect 218
  - checking in new volumes to library 137
  - checking out 148
  - contents, querying 338
  - defining to storage pools 237
  - delaying reuse 329, 766
  - deleting 362, 363, 769
  - detailed report 340
  - determining which are mounted 154, 733
  - disk storage 237
  - disk storage pool, auditing 788
  - dismounting 154
  - errors, read and write 336
  - estimated capacity 336
  - finding for client node 341
  - help in dsmc loop session 414
  - inventory maintenance 143
  - location 337
  - managing 147
  - monitoring life 337
  - monitoring movement of data 353
  - monitoring use 335
  - mount retention time 203
  - moving files between 349
  - number of times mounted 337
  - off-site, limiting number to be reclaimed 328
  - offsite, limiting number to be reclaimed 227
  - overview 50
  - pending status 337
  - private 50
  - querying contents 338
  - querying for general information 335
  - random access storage pools 222, 236, 238
  - reclamation 323
  - removing from a library 148
  - restoring from active-data pools 780
  - restoring random-access 550
  - reuse delay 329, 766
  - scratch 50
  - scratch, using 238
  - sequential 237

- volumes (*continued*)
  - sequential storage pools 133, 237
  - setting access mode 239
  - standard report 339
  - status, in automated library 51
  - status, information on 336
  - swapping 140
  - updating 147, 237
  - using private 50, 51
  - varying on and off 77
  - WORM scratch category 100

## W

- Web administrative interface
  - description 19
  - limitation of browser for script definitions 578
- Web backup-archive client
  - granting authority to 395
  - remote access overview 393
  - URL 368, 393
- wizard
  - client configuration 373
  - description 19
  - setup 373
- workstation, registering 371
- WORM devices and media
  - DLT WORM 141
  - IBM 3592 141
  - LTO WORM 141
  - Quantum LTO3 141
  - Sony AIT50 and AIT100 141
  - special considerations for WORM media 141
  - Sun StorageTek T10000B drives 142
  - VolSafe
    - considerations for media 141
    - defining VOLSAFE device classes 215
  - WORM FILE and SnapLock 476
  - WORM parameter 215
- WORM scratch category for volumes in 349X library 100
- writing data simultaneously to primary and copy storage pools
  - use during client storage operations 285





Program Number: 5608-E01, 5608-E02, 5608-E03, 5608-E07, 5608-E12

Printed in USA

SC23-9771-02

