IBM Tivoli Storage Manager for Oracle Solaris
Version 7.1

Administrator's Guide



IBM Tivoli Storage Manager
for Oracle Solaris
Version 7.1

Administrator's Guide



Note:
Before using this information and the product it supports, read the information in "Notices" on page 1077.

First edition (December 2013)

This edition applies to version 7, release 1, modification 0 of IBM Tivoli Storage Manager (product numbers 5608-E01, 5608-E02, 5608-E03), and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1993, 2013.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface xv
Who should read this guide xv
Publications xv
Conventions used in this guide xv

New for IBM Tivoli Storage Manager Version 7.1 xvii
New in Version 7.1 xvii
Automatic client failover for restore operations from replicated servers xvii
Software upgrade for the server database manager xvii
Deprecated device types xvii
EMC Centera availability on Linux x86_64 systems for Tivoli Storage Manager servers . . xviii
File-space level collocation groups xviii
Improved server efficiency when migrating from disk for nodes with a large number of file spaces xviii
Shared memory for database backup and restore operations xix
Immediate use of space that is added to the server database xix
Server components that are not delivered with Version 7.1 xix
New best practices xx
Duplicate identification processes and memory requirements xx
Server startup and setting ulimits xx
Configuring the server to prevent issues with database backups xx

Part 1. Tivoli Storage Manager basics 1

Chapter 1. Tivoli Storage Manager overview 3
How client data is stored 5
Data-protection options 7
Data movement to server storage 13
Consolidation of backed-up client data 14
How the server manages storage 15
Device support for server storage 15
Data migration through the storage hierarchy . . 15
Removal of expired data 16

Chapter 2. Tivoli Storage Manager concepts 17
Interfaces to Tivoli Storage Manager 17
Storage configuration and management 18
Disk devices 18
Removable media devices 19
Defined volumes and scratch volumes 19
Migrating data from disk to tape 21
Techniques for managing your storage pools and volumes 21
Management of client operations 23
Managing client nodes. 23
Security management 25
Managing client data with policies 26
Schedules for client operations 28
Server maintenance. 30
Server-operation management 30
Server script automation 31
Database and recovery-log management 32
Sources of information about the server 32
Tivoli Storage Manager server networks 32
Exporting and importing data 33
Protecting Tivoli Storage Manager and client data 34
Protecting the server 34

Part 2. Configuring and managing storage devices 37

Chapter 3. Storage device concepts . . 39
Road map for key device-related task information 39
Tivoli Storage Manager storage devices 40
Tivoli Storage Manager storage objects 40
Libraries 40
Drives 44
Device class 44
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 57
Library sharing 59
Removable media mounts and dismounts 59
How Tivoli Storage Manager uses and reuses removable media 60
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 . . . 67

Chapter 4. Magnetic disk devices . . . 69
Requirements for disk systems 69

© Copyright IBM Corp. 1993, 2013

Comparison of random access and sequential access disk devices 71
File systems and raw logical volumes for random access storage 75
Configuring random access volumes on disk devices 76
Configuring FILE sequential volumes on disk devices 77
Varying disk volumes online or offline 78
Cache copies for files stored on disk 78
Freeing space on disk 79
Scratch FILE volumes 79
Volume history file and volume reuse 79

Chapter 5. Attaching devices for the server 81
Attaching a manual drive to your system 81
Attaching an automated library device to your system 82
Setting the library mode 83
Selecting a device driver 83
IBM device drivers 83
Tivoli Storage Manager device drivers 83
Device special file names 84
Correcting device special file names with SANDISCOVERY 85
Tivoli Storage Manager server device utilities . . . 85
autoconf (Auto configure devices) 85
rmstdev (Detect and delete device special files) 86
Displaying information about devices with tsmdlist. 86
Installing and configuring device drivers 87
Installing device drivers for IBM SCSI tape devices 87
Installing device drivers for IBM 349x libraries 89
Configuring Tivoli Storage Manager device drivers for vendor acquired SCSI-attached devices 89
Reinstalling or upgrading the Tivoli Storage Manager device driver. 91
Editing configuration files 92
Preventing tape labels from being overwritten . . . 94
Installing the Tivoli Storage Manager device driver to Solaris Zones 95
Adding devices to non-global zones 95
Displaying devices configured in zones 97
Removing devices from non-global zones . . . 98
Installing the Centera SDK for Centera shared libraries 99

Chapter 6. Configuring storage devices 101
Configuring libraries for use by one server . . . 102
Example: Configure a SCSI or virtual tape library with a single drive device type 103
Example: Configure a SCSI or virtual tape library with multiple drive device types . . . 104
Example: Configure an ACSLS library with a single drive device type 106
Example: Configure an ACSLS library with multiple drive device types. 107
Example: Configure a 3494 library with a single drive device type 108
Example: Configure a 3494 library with multiple drive device types 109
Configuring library sharing. 110
Example: Configure library sharing for SCSI or virtual tape libraries 111
Example: Configure library sharing for ACSLS libraries 112
Example: Configure library sharing for 3494 libraries 113
Managing virtual tape libraries 114
Considerations for using virtual tape libraries 114
Adding a virtual tape library to your environment. 116
Defining or deleting all drives and paths for a single library 116
Migrating a shared IBM 3494 library to a library manager 117
Sharing an IBM 3494 library by static partitioning of drives 118
Removable file device configuration 119
Example of removable file support 119
Labeling requirements for removable file device types 121
Configuration for libraries controlled by media manager programs 121
Setting up Tivoli Storage Manager to work with an external media manager. 121
Externally controlled IBM Tivoli Storage Manager media. 122
Configuring manually mounted devices 123
Defining devices as part of a manual library . . 123
Labeling volumes 125
Configuring IBM Tivoli Storage Manager for LAN-free data movement 125
Validating your LAN-free configuration . . . 126
SAN discovery functions for non-root users . . 127
Impacts of device changes on the SAN 127

Chapter 7. Managing removable media operations. 129
Preparing removable media 129
Labeling removable media volumes 130
Checking new volumes into a library 133
Write-once, read-many tape media 138
Managing the volume inventory 140
Controlling access to volumes 140
Reusing tapes in storage pools 141
Setting up a tape rotation 141
Reusing volumes used for database backups and export operations 143
Maintaining a supply of scratch volumes . . . 143
Maintaining a supply of volumes in a library containing WORM media 144
Managing volumes in automated libraries 144
Changing the status of a volume 145
Removing volumes from a library 145
Managing a full library 146
Auditing a library's volume inventory 147

Maintaining a supply of scratch volumes in an automated library 148
Operations with shared libraries 148
Managing server requests for media 149
Managing libraries 151
Requesting information about libraries 151
Updating libraries 152
Deleting libraries 153
Managing drives 153
Requesting information about drives 153
Updating drives 154
Drive encryption 155
Validating data during read/write operations to tape 156
Cleaning drives 160
Deleting drives 165
Managing paths 165
Requesting information about paths 165
Updating paths 165
Deleting paths 166
Managing data movers 166
Requesting information about data movers . . 166
Updating data movers 166
Deleting data movers 167
Tape alert messages 167

Chapter 8. Tips for defining devices 169

Defining devices and paths 169
Defining libraries 169
Defining drives 170
Defining data movers 171
Defining paths 172
Shared FILE volumes 173
Defining device classes 174
Sequential-access device types 175
Defining tape device classes 176
Defining device classes for removable media devices 183
Defining sequential-access disk (FILE) device classes 184
Defining LTO device classes 188
Defining SERVER device classes 191
Defining device classes for StorageTek VolSafe devices 192
Defining device classes for CENTERA devices 194
Obtaining information about device classes . . 195
How Tivoli Storage Manager fills volumes . . 196

Chapter 9. Using NDMP for operations with NAS file servers 199

NDMP requirements 199
Interfaces for NDMP operations 201
Data formats for NDMP backup operations . . 202
NDMP operations management 202
Managing NAS file server nodes 203
Managing data movers used in NDMP operations 204
Dedicating a Tivoli Storage Manager drive to NDMP operations 204
Storage pool management for NDMP operations 205
Managing tables of contents 205
Preventing long-running, inactive NDMP connections from closing 206
Configuring Tivoli Storage Manager for NDMP operations 207
Configuring Tivoli Storage Manager policy for NDMP operations 208
Tape libraries and drives for NDMP operations 211
Attaching tape library robotics for NAS-attached libraries 215
Registering NAS nodes with the Tivoli Storage Manager server 219
Defining a data mover for the NAS file server 220
Defining paths for NDMP operations 220
Labeling and checking tapes into the library . . 224
Scheduling NDMP operations 224
Defining virtual file spaces 224
Tape-to-tape copy to back up data 225
Tape-to-tape copy to move data 225
Backing up and restoring NAS file servers using NDMP 226
NAS file servers: backups to a single Tivoli Storage Manager server 227
Backing up NDMP file server to Tivoli Storage Manager server backups 228
File-level backup and restore for NDMP operations 229
Interfaces for file-level restore 230
International characters for NetApp file servers 230
File level restore from a directory-level backup image 231
Directory-level backup and restore 231
Directory-level backup and restore for NDMP operations 232
Backing up and restoring with snapshots . . . 232
Backup and restore using NetApp SnapMirror to Tape feature 233
NDMP backup operations using Celerra file server integrated checkpoints 234
Replicating NAS nodes with NDMP backup data 234

Chapter 10. Managing storage pools and volumes 235

Storage pools 236
Primary storage pools 236
Copy storage pools 237
Active-data pools 237
Example: Setting up server storage 239
Defining storage pools 241
Task tips for storage pools 247
Storage pool volumes 249
Random-access storage pool volumes 249
Sequential-access storage pool volumes 249
Preparing volumes for random-access storage pools 251
Preparing volumes for sequential-access storage pools 252
Updating storage pool volumes 254
Access modes for storage pool volumes . . . 255
Storage pool hierarchies 256
Setting up a storage pool hierarchy 257
How the server groups files before storing . . 258

Where the server stores files 259
Example: How the server determines where to store files in a hierarchy 260
Backing up the data in a storage hierarchy . . 261
Staging client data from disk to tape 266
Migrating files in a storage pool hierarchy. . . . 267
Migrating disk storage pools 269
Migrating sequential-access storage pools . . . 275
The effect of migration on copy storage pools and active-data pools. 279
Caching in disk storage pools 280
How the server removes cached files 280
Effect of caching on storage pool statistics . . . 281
Deduplicating data 281
Data deduplication overview 281
Data deduplication limitations. 284
Planning guidelines for data deduplication . . 287
Detecting possible security attacks during client-side data deduplication 298
Evaluating data deduplication in a test environment. 299
Managing deduplication-enabled storage pools 302
Controlling data deduplication 306
Displaying statistics about server-side data deduplication 314
Displaying statistics about client-side data deduplication 316
Querying about data deduplication in file spaces 319
Scenarios for data deduplication 320
Data deduplication and data compatibility . . 325
Data deduplication and disaster recovery management 326
Writing data simultaneously to primary, copy, and active-data pools 327
Guidelines for using the simultaneous-write function 328
Limitations that apply to simultaneous-write operations 329
Controlling the simultaneous-write function . . 331
Simultaneous-write operations: Examples . . . 335
Planning simultaneous-write operations . . . 348
Simultaneous-write function as part of a backup strategy: Example 352
Keeping client files together using collocation . . 353
The effects of collocation on operations. . . . 355
How the server selects volumes with collocation enabled 357
How the server selects volumes with collocation disabled 359
Collocation on or off settings 359
Collocation of copy storage pools and active-data pools 360
Planning for and enabling collocation 361
Reclaiming space in sequential-access storage pools 363
How Tivoli Storage Manager reclamation works 364
Reclamation thresholds 366
Reclaiming volumes with the most reclaimable space 366
Starting reclamation manually or in a schedule 367
Optimizing drive usage using multiple concurrent reclamation processes 367
Reclaiming volumes in a storage pool with one drive 368
Reducing the time to reclaim tape volumes with high capacity 369
Reclamation of write-once, read-many (WORM) media 369
Controlling reclamation of virtual volumes . . 370
Reclaiming copy storage pools and active-data pools 370
How collocation affects reclamation 374
Estimating space needs for storage pools 375
Estimating space requirments in random-access storage pools 376
Estimating space needs in sequential-access storage pools 377
Monitoring storage-pool and volume usage . . . 378
Monitoring space available in a storage pool 378
Monitoring the use of storage pool volumes . . 380
Monitoring migration processes 388
Monitoring the use of cache space on disk storage 391
Obtaining information about the use of storage space 392
Moving data from one volume to another volume 396
Data movement within the same storage pool 397
Data movement to a different storage pool . . 398
Data movement from offsite volumes in copy storage pools or active-data pools 398
Moving data 399
Moving data belonging to a client node 402
Moving data in all file spaces belonging to one or more nodes 402
Moving data in selected file spaces belonging to a single node 403
Obtaining information about data-movement processes 404
Troubleshooting incomplete data-movement operations 404
Renaming storage pools 405
Defining copy storage pools and active-data pools 405
Example: Defining a copy storage pool 407
Properties of primary, copy, and active-data pools 408
Deleting storage pools 409
Deleting storage pool volumes 410
Deleting empty storage pool volumes 411
Deleting storage pool volumes that contain data 411

Part 3. Managing client operations 413

Chapter 11. Adding client nodes . . . 415
Overview of clients and servers as nodes 415
Installing client node software. 416
Registering nodes with the server 416
Accepting default closed registration or enabling open registration 417
Registering nodes with client options sets . . . 419

Registering a network-attached storage file server as a node 419
Registering a source server as a node on a target server 420
Registering an API to the server 420
Connecting nodes with the server 421
Required client options 421
UNIX and Linux client options 422
Creating or updating a client options file 422
Using a text editor to create or configure a client options file 422
Using the client configuration wizard to create or update a client options file 422
Comparing network-attached nodes to local nodes 423
Adding clients through the administrative command line client 424
Enabling open registration 424
Example: registering three client nodes using the administrative command line 424

Chapter 12. Managing client nodes 425

Managing nodes 425
Managing client nodes across a firewall . . . 426
Updating client node information 428
Command-line backup-archive client automatic deployment feature: overview 428
Renaming client nodes 433
Locking and unlocking client nodes 434
Deleting client nodes 434
Consolidating multiple clients under a single client node name 435
Displaying information about client nodes. . . 438
Overview of remote access to web backup-archive clients 439
Managing client access authority levels. . . . 441
Collecting processor value unit information in a VMware host environment 444
Managing file spaces 445
Defining client nodes and file spaces 445
Supporting Unicode-enabled clients 447
Displaying information about file spaces . . . 456
Moving data for a client node 457
Deleting file spaces 457
Managing client option files 458
Creating client option sets on the server . . . 459
Managing client option sets 461
Managing IBM Tivoli Storage Manager sessions 461
Displaying information about IBM Tivoli Storage Manager sessions 462
Canceling an IBM Tivoli Storage Manager session 463
When a client session is automatically canceled 464
Disabling or enabling access to the server . . . 465
Managing client restartable restore sessions . . 465

Chapter 13. Implementing policies for client data 469

Basic policy planning. 470
Reviewing the standard policy 471
Getting users started 472
Changing policy 472
File expiration and expiration processing . . . 473
Client operations controlled by policy 474
Backup and restore 474
Archive and retrieve 475
Client migration and recall 475
The parts of a policy 476
Relationships among clients, storage, and policy 478
More on management classes 480
Contents of a management class 480
Default management classes 481
The include-exclude list 482
How files and directories are associated with a management class 483
How Tivoli Storage Manager selects files for policy operations 486
Incremental backup 486
Selective backup 488
Logical volume backup 489
Archive 489
Automatic migration from a client node . . . 490
Creating your own policies 490
Example: sample policy objects 491
Defining and updating a policy domain . . . 492
Defining and updating a policy set 494
Defining and updating a management class . . 495
Defining and updating a backup copy group 496
Defining and updating an archive copy group 502
Assigning a default management class 504
Validating and activating a policy set 504
Assigning client nodes to a policy domain. . . . 506
Running expiration processing to delete expired files 506
Running expiration processing automatically 507
Using commands to control expiration processing 507
Additional expiration processing with disaster recovery manager 508
Protection and expiration of archive data 508
Data retention protection 508
Deletion hold 509
Protecting data using the NetApp SnapLock licensed feature. 510
Reclamation and the SnapLock feature 511
Set up SnapLock volumes as Tivoli Storage Manager WORM FILE volumes 515
Policy configuration scenarios 516
Configuring policy for direct-to-tape backups 516
Configuring policy for Tivoli Storage Manager application clients 517
Policy for logical volume backups 517
Configuring policy for NDMP operations . . . 518
Configuring policy for LAN-free data movement 519
Policy for Tivoli Storage Manager servers as clients 521
Setting policy to enable point-in-time restore for clients 522
Distributing policy using enterprise configuration 522
Querying policy 523
Querying copy groups 523

Querying management classes 524
Querying policy sets 525
Querying policy domains 525
Deleting policy 525
Deleting copy groups 526
Deleting management classes 526
Deleting policy sets 526
Deleting policy domains 527

Chapter 14. Managing data for client nodes 529
Validating a node's data 529
Performance considerations for data validation 530
Validating a node's data during a client session 530
Encrypting data on tape 530
Choosing an encryption method 531
Changing your encryption method and hardware configuration 532
Securing sensitive client data 533
Setting up shredding 534
Ensuring that shredding is enforced 535
Creating and using client backup sets 537
Generating client backup sets on the server . . 538
Restoring backup sets from a backup-archive client 542
Moving backup sets to other servers 542
Managing client backup sets 543
Enabling clients to use subfile backup 546
Setting up clients to use subfile backup 547
Managing subfile backups 547
Archiving data 548
Archive operations overview 549
Managing storage usage for archives 549

Chapter 15. Scheduling operations for client nodes 553
Prerequisites to scheduling operations 553
Scheduling a client operation 554
Defining client schedules 555
Associating client nodes with schedules . . . 556
Starting the scheduler on the clients 556
Displaying schedule information 557
Checking the status of scheduled operations . . 557
Creating schedules for running command files . . 558
Updating the client options file to automatically generate a new password 559

Chapter 16. Managing schedules for client nodes 561
Managing IBM Tivoli Storage Manager schedules 561
Adding new schedules 561
Copying existing schedules 562
Modifying schedules 562
Deleting schedules 562
Displaying information about schedules . . . 563
Managing node associations with schedules . . . 563
Adding new nodes to existing schedules . . . 564
Moving nodes from one schedule to another 564
Displaying nodes associated with schedules . . 564
Removing nodes from schedules 565
Managing event records 565
Displaying information about scheduled events 565
Managing event records in the server database 567
Managing the throughput of scheduled operations 568
Modifying the default scheduling mode . . . 568
Specifying the schedule period for incremental backup operations 571
Balancing the scheduled workload for the server 571
Controlling how often client nodes contact the server 573
Specifying one-time actions for client nodes . . . 575
Determining how long the one-time schedule remains active 575

Part 4. Maintaining the server . . . 577

Chapter 17. Managing server operations 579
Licensing IBM Tivoli Storage Manager 579
Registering licensed features 580
Monitoring licenses 581
Role of processor value units in assessing licensing requirements 582
Estimating processor value units 585
Collecting processor value unit information in a VMware host environment 587
Starting the Tivoli Storage Manager server . . . 588
Starting the server on AIX, HP-UX, Linux, and Solaris 588
Running multiple server instances on a single system 597
Halting the server 598
Moving the Tivoli Storage Manager server to another system 599
Date and time on the server 600
Managing server processes 600
Requesting information about server processes 601
Canceling server processes 601
Preempting operations 602
Changing names in Tivoli Storage Manager . . . 604
Setting the server name 604
Changing the host name for a Tivoli Storage Manager server 604
Add or update server options 605
Adding or updating a server option without restarting the server 605
Deleting information about volume history . . . 606
Getting help on commands and error messages . . 607

Chapter 18. Automating server operations 609
Automating a basic administrative command schedule 610
Defining the schedule 610
Verifying the schedule 611
Tailoring schedules 611
Using classic and enhanced command schedules 614
Copying schedules 614
Deleting schedules 615
Managing scheduled event records 615

Querying events 615
Removing event records from the database . . 616
Tivoli Storage Manager server scripts 616
Defining a server script 617
Managing server scripts 623
Running a server script 626
Using macros 627
Writing commands in a macro. 627
Writing comments in a macro 628
Using continuation characters 628
Using substitution variables in a macro. . . . 629
Running a macro 630
Command processing in a macro 630

Chapter 19. Managing the database and recovery log 633
Database and recovery log overview 633
Database: Overview 634
Connecting the server to the database with TCP/IP 636
Recovery log 636
Setting the user data limit for the database . . . 640
Disk space requirements for the server database and recovery log 640
Capacity planning. 641
Estimating database space requirements . . . 642
Estimating recovery log space requirements . . 646
Monitoring space utilization for the database and recovery logs 659
Monitoring the database and recovery log. . . . 661
Increasing the size of the database 662
Manually redistributing data and reclaiming space after you add directories to the database . 664
Reducing the size of the database 666
Scheduling table and index reorganization. . . . 666
Restrictions to table and index reorganization 667
Scheduling table or index reorganization . . . 668
Increasing the size of the active log 668
Reducing the size of the active log 669
Moving the database and recovery log on a server 669
Moving both the database and recovery log . . 670
Moving only the database 670
Moving only the active log, archive log, or archive failover log 671
Specifying alternative locations for database log files 672
Specifying an alternative location with the `ARCHFAILOVERLOGDIRECTORY` server option or parameter 673
Specifying an alternative location with the `ARCHLOGDIRECTORY` server option or parameter . 673
Specifying the location of `RstDbLog` using the **`RECOVERYDIR`** parameter 674
Adding optional logs after server initialization . . 674
Transaction processing 675
Files moved as a group between client and server 675

Chapter 20. Managing a network of Tivoli Storage Manager servers . . . 677
Concepts for managing server networks 677
Enterprise configuration. 678
Command routing. 679
Central monitoring for the Tivoli Storage Manager server. 679
Data storage on another server 680
Examples: management of multiple Tivoli Storage Manager servers 680
Enterprise-administration planning 682
Setting up communications among servers . . . 682
Setting up communications for enterprise configuration and enterprise event logging . . 682
Setting up communications for command routing 686
Updating and deleting servers. 690
Setting up enterprise configurations 691
Enterprise configuration scenario 692
Creating the default profile on a configuration manager 696
Creating and changing configuration profiles 697
Getting information about profiles 704
Subscribing to a profile 707
Refreshing configuration information 711
Managing problems with configuration refresh 712
Returning managed objects to local control . . . 712
Setting up administrators for the servers 713
Managing problems with synchronization of profiles 713
Switching a managed server to a different configuration manager 714
Deleting subscribers from a configuration manager 714
Renaming a managed server 715
Completing tasks on multiple servers 715
Managing multiple servers through a web-based interface 715
Routing commands 716
Setting up server groups 718
Querying server availability 722
Using virtual volumes to store data on another server 722
Setting up source and target servers for virtual volumes 724
Performance limitations for virtual volume operations 725
Performing operations at the source server . . 726
Reconciling virtual volumes and archive files 729

Chapter 21. Exporting and importing data 731
Reviewing data that can be exported and imported 731
Exporting restrictions. 732
Deciding what information to export 732
Deciding when to export 733
Exporting data directly to another server 734
Options to consider before exporting 735
Preparing to export to another server for immediate import 738
Monitoring the server-to-server export process 740

Exporting administrator information to another server 741
Exporting client node information to another server 741
Exporting policy information to another server 742
Exporting server data to another server . . . 742
Exporting and importing data using sequential media volumes 743
Using preview before exporting or importing data 743
Planning for sequential media used to export data 744
Exporting tasks 745
Importing data from sequential media volumes 748
Monitoring export and import processes . . . 760
Exporting and importing data from virtual volumes 763

Part 5. Monitoring operations . . . 765

Chapter 22. Daily monitoring tasks 767
Monitoring operations using the command line 768
Monitoring your server processes daily 768
Monitoring your database daily 769
Monitoring disk storage pools daily 772
Monitoring sequential access storage pools daily 773
Monitoring scheduled operations daily 776

Chapter 23. Basic monitoring methods 779
Using IBM Tivoli Storage Manager queries to display information 779
Requesting information about IBM Tivoli Storage Manager definitions 779
Requesting information about client sessions 780
Requesting information about server processes 781
Requesting information about server settings 782
Querying server options 782
Querying the system 783
Using SQL to query the IBM Tivoli Storage Manager database 784
Using SELECT commands 784
Using SELECT commands in Tivoli Storage Manager scripts 787
Querying the SQL activity summary table . . . 788
Creating output for use by another application 789
Using the Tivoli Storage Manager activity log . . 789
Requesting information from the activity log 790
Setting a retention period for the activity log 791
Setting a size limit for the activity log 791

Chapter 24. Monitoring Tivoli Storage Manager accounting records 793

Chapter 25. Reporting and monitoring with Tivoli Monitoring for Tivoli Storage Manager 795

Chapter 26. Logging IBM Tivoli Storage Manager events to receivers . 797
Enabling and disabling events 798
Beginning and ending event logging 799
Logging events to the IBM Tivoli Storage Manager server console and activity log 799
Logging events to a file exit and a user exit . . . 800
Adding a file exit or user exit option 800
Logging events to the Tivoli Enterprise Console 801
Controlling the format of events 802
Encoding events to UTF-8 804
Setting up a Tivoli Enterprise Console as a receiver 804
Logging events to an SNMP manager 805
Configuring IBM Tivoli Storage Manager SNMP 809
Enterprise event logging: logging events to another server 811
Querying event logging 812
User exit and file exit receivers 813
Sample user exit declarations 813
Sample user-exit program 815
Readable text file exit (FILETEXTEXIT) format 816

Part 6. Protecting the server . . . 819

Chapter 27. Managing Tivoli Storage Manager security 821
Securing communications 821
Secure Sockets Layer and Transport Layer Security communication 822
Setting up SSL 823
Securing the server console 833
Managing access to the server and clients 834
Managing Tivoli Storage Manager administrator IDs 834
Restricting a non-root user ID from performing backups as root 842
Managing passwords and logon procedures . . . 843
Configuring a directory server for password authentication 845
Setting the policy for an LDAP-authenticated password 846
Configuring the Tivoli Storage Manager server to authenticate passwords with an LDAP directory server 847
Registering nodes and administrator IDs to authenticate passwords with an LDAP directory server 848
Updating nodes and administrator IDs to authenticate passwords with a directory server . 849

Determining which nodes and administrator IDs are configured to authenticate with an LDAP server 850
Modifying the default password expiration period for passwords that are managed by the Tivoli Storage Manager server 850
Scenarios for authenticating passwords 851
Setting a limit for invalid password attempts 853
Setting a minimum length for a password . . . 854
Disabling the default password authentication 855

Chapter 28. Protecting and recovering the server infrastructure and client data 857

Protecting the database and infrastructure setup files 858
Backing up the server database 858
Protecting infrastructure setup files 865
Protecting client data 870
Protecting the data that is in primary storage pools 870
Auditing storage pool volumes 875
Fixing damaged files 884
Scenario: Protecting the database and storage pools 886
Recovering the database and client data 888
Restoring the database 888
Restoring storage pools and storage pool volumes 895
Restoring to a point-in-time in a shared library environment. 903
Restoring to a point-in-time a library manager server 903
Restoring to a point-in-time a library client server 904
Example: Recovering to a point-in-time. . . . 904

Chapter 29. Replicating client node data 907

Automatic failover for data recovery overview . . 908
Source and target node-replication servers. . . . 909
Replication server configurations 909
Policy management for node replication . . . 910
Node replication processing 911
Replication rules 911
Replication state 915
Replication mode 918
Replication of deduplicated data 919
Client node attributes that are updated during replication 920
Node replication restrictions 921
Task tips for node replication 923
Change replication rules. 923
Add and remove client nodes for replication 924
Manage replication servers 924
Validate a configuration and preview results 925
Manage replication processing. 925
Monitor replication processing and verify results 926
Planning for node replication 927
Determining server database requirements for node replication 928
Estimating the total amount of data to be replicated. 929
Estimating network bandwidth required for replication 929
Calculating the time that is required for replication 930
Selecting a method for the initial replication . . 931
Scheduling incremental replication after the initial replication 933
Setting up the default replication configuration . . 934
Step 1: Setting up server-to-server communications 936
Step 2: Specifying a target replication server . . 938
Step 3: Configuring client nodes for replication 938
Customizing a node replication configuration . . 940
Changing replication rules 940
Scenario: Converting to node replication from import and export operations 949
Adding and removing client nodes for replication 950
Managing source and target replication servers 953
Verifying a node replication setup before processing 957
Validating a replication configuration 958
Previewing node replication results 958
Managing data replication 959
Replicating data by command 959
Controlling throughput for node replication . . 964
Disabling and enabling node replication . . . 966
Purging replicated data in a file space 971
Replicating client node data after a database restore. 971
Canceling replication processes 973
Monitoring node replication processing and verifying results 973
Displaying information about node replication settings 973
Displaying information about node replication processes 973
Measuring the effectiveness of a replication configuration 974
Measuring the effects of data deduplication on node replication processing. 975
Retaining replication records 976
Recovering and storing client data after a disaster 976
Restoring, retrieving, and recalling data manually from a target replication server . . . 976
Converting client nodes for store operations on a target replication server 977
Removing a node replication configuration . . . 977

Chapter 30. Disaster recovery manager 979

Querying defaults for the disaster recovery plan file 980
Specifying defaults for the disaster recovery plan file 980
Specifying defaults for offsite recovery media management 983
Specifying recovery instructions for your site. . . 985

Specifying information about your server and client node machines 986
Specifying recovery media for client machines . . 988
Creating and storing the disaster recovery plan . . 989
Storing the disaster recovery plan locally . . . 991
Storing the disaster recovery plan on a target server 991
Managing disaster recovery plan files stored on target servers 991
Displaying information about recovery plan files 992
Displaying the contents of a recovery plan file 992
Restoring a recovery plan file 993
Expiring recovery plan files automatically . . . 993
Deleting recovery plan files manually 994
Moving backup media 994
Moving copy storage pool and active-data pool volumes offsite 996
Moving copy storage pool and active-data pool volumes on-site 998
Managing the Disaster Recovery Manager tasks 999
Preparing for disaster recovery 1001
Recovering from a disaster 1003
Server recovery scenario 1004
Client recovery scenario 1007
Recovering with different hardware at the recovery site 1010
Automated SCSI library at the original and recovery sites 1010
Automated SCSI library at the original site and a manual scsi library at the recovery site . . . 1011
Managing copy storage pool volumes and active-data pool volumes at the recovery site . 1012
Disaster recovery manager checklist 1013
The disaster recovery plan file 1018
Breaking out a disaster recovery plan file. . . 1018
Structure of the disaster recovery plan file . . 1018
Example disaster recovery plan file. 1021

Chapter 31. Integrating disaster recovery manager and node replication into your disaster recovery strategy 1041
Plan for a disaster recovery strategy 1042
Tier 0: No disaster recovery capability. 1043
Tier 1: Offsite vaulting from a single production site 1043
Tier 2: Offsite vaulting with a recovery site . . . 1044
Tier 3: Electronic vaulting of critical data 1044
Tier 4: Active data management at peer sites . . 1045
Tier 5: Synchronous replication 1046
Tier 6: Synchronous replication with coupling or clustering applications 1047

Part 7. Appendixes 1049

Appendix A. External media management interface description . . 1051
CreateProcess call 1051
Processing during server initialization. 1052
Processing for mount requests 1052
Processing for release requests 1053
Processing for batch requests. 1053
Error handling 1054
Begin batch request 1054
End batch request 1054
Volume query request 1055
Initialization requests 1056
Volume eject request 1056
Volume release request. 1057
Volume mount request 1058
Volume dismount request 1061

Appendix B. User exit and file exit receivers 1063
Sample user exit declarations. 1063
Sample user-exit program 1065
Readable text file exit (FILETEXTEXIT) format 1066

Appendix C. Tivoli support information 1069
Communities and other learning resources . . . 1069
Searching knowledge bases 1071
Searching the Internet 1071
Using IBM Support Assistant. 1071
Finding product fixes 1072
Receiving notification of product fixes. . . . 1072
Contacting IBM Software Support 1072
Setting up and managing support contracts 1073
Determining the business impact 1073
Describing the problem and gathering background information 1073
Submitting the problem to IBM Software Support 1074

Appendix D. Accessibility features for the Tivoli Storage Manager product family 1075

Notices 1077
Trademarks 1079
Privacy policy considerations. 1079

Glossary 1081
A 1081
B 1083
C 1084
D 1085
E 1087
F 1088
G 1088
H 1089
I 1090
J 1090
K 1090
L 1091
M 1092
N 1093
O 1094

P 1094
Q 1095
R 1096
S 1097
T 1100
U 1100
V 1101
W 1102

Index 1103

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

Publications for the Tivoli Storage Manager family of products are available online. The Tivoli Storage Manager product family includes IBM Tivoli Storage FlashCopy® Manager, IBM Tivoli Storage Manager for Space Management, IBM Tivoli Storage Manager for Databases, and several other storage management products from IBM Tivoli.

To search across all publications or to download PDF versions of individual publications, go to the Tivoli Storage Manager information center at http://pic.dhe.ibm.com/infocenter/tsminfo/v7r1.

You also can find the Tivoli Storage Manager product family information centers and other information centers that contain official product documentation for current and previous versions of Tivoli products at Tivoli Documentation Central. Tivoli Documentation Central is available at http://www.ibm.com/developerworks/community/wikis/home/wiki/Tivoli Documentation Central.

Conventions used in this guide

- Command to be entered on the Oracle Solaris command line:

  ```
  > dsmadmc
  ```

- 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.

© Copyright IBM Corp. 1993, 2013

New for IBM Tivoli Storage Manager Version 7.1

Many features in the Tivoli Storage Manager Version 7.1 server are new for previous Tivoli Storage Manager users.

New in Version 7.1

New features and other changes are available in the Tivoli Storage Manager Version 7.1 server.

Automatic client failover for restore operations from replicated servers

A Tivoli Storage Manager Version 7.1 client can automatically fail over to a target replication server, if the source replication server is unavailable.

To use automatic failover when node replication is in use, both the source and target replication servers and the client must be at V7.1. You can use only one failover server for each replicating node at any time. The failover server is the last server that a node successfully replicated to. The client can recover data from the target replication server, but it cannot store data during failover processing.

Software upgrade for the server database manager

Tivoli Storage Manager Version 7.1 is installed with the latest version of the IBM DB2® database software, V10.5. The DB2 technology provides management functions for the server database.

The previous Tivoli Storage Manager server release, V6.3, was packaged with a DB2 V9.7 database.

If you are upgrading the server, you must ensure that the correct version of DB2 is installed. If you are using a Tivoli Storage Manager V6.1 server, and you want to upgrade to V7.1, you must upgrade the V6.1 server to V6.2 or V6.3. Then, upgrade the server from V6.2 or V6.3 to V7.1. The Tivoli Storage Manager V6.1 server was packaged with a DB2 V9.5 database, which cannot be upgraded to V10.5. The Tivoli Storage Manager V6.2 and V6.3 servers were packaged with a DB2 V9.7 database, which can be upgraded to V10.5.

Deprecated device types

In Tivoli Storage Manager Version 7.1, certain device types are deprecated.

The following device types are deprecated. While you can use these device types with Tivoli Storage Manager Version 7.1, plan to migrate data to new devices before you upgrade to a later release of Tivoli Storage Manager.

- 3490
- 3570
- CARTRIDGE
- OPTICAL
- WORM
- QIC

© Copyright IBM Corp. 1993, 2013

- DTF

For a list of devices and valid device class formats, see the Tivoli Storage Manager Supported Devices website for your operating system:

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

EMC Centera availability on Linux x86_64 systems for Tivoli Storage Manager servers

You can now use EMC Centera for versions of Linux x86_64 that are supported by the Tivoli Storage Manager Version 7.1 server.

For details about Centera-related commands, refer to the *Administrator's Reference*.

File-space level collocation groups

File-space level collocation groups are introduced in Tivoli Storage Manager Version 7.1.

You can group file spaces that belong to a single node, which allows data for these file spaces to be collocated efficiently without requiring separate volumes for each file space. When you use file-space level collocation groups, you can group data for a limited set of file systems, for example, virtual machines. Fewer volumes are required for the data and placement can be coordinated in the server storage hierarchy.

Use the **DEFINE COLLOCMEMBER** command to define members of a file-space collocation group.

For more information about collocation, see *Keeping client files together using collocation* in the *Administrator's Guide*.

For detailed information about the **DEFINE COLLOCMEMBER** command, see the *Administrator's Reference*.

Improved server efficiency when migrating from disk for nodes with a large number of file spaces

With the new functions that are added to migration processing, you can improve the efficiency of the server by using file space level migration. Nodes with multiple large file spaces can take advantage of faster migration processing for random-access storage pools.

For updated information about how collocation settings affect data processing and how the server migrates files, see *How the server selects files to migrate* in the *Administrator's Guide*.

Shared memory for database backup and restore operations

You can now use shared memory to reduce processor load and improve throughput, if the database backup performance is slow.

You can manually configure a Tivoli Storage Manager server, or use the instance configuration wizard, to use shared memory with DB2.

Immediate use of space that is added to the server database

When you add space to the database, new database directories are now available for immediate use and parallel I/O performance is improved.

You can add directories to the database by using the **EXTEND DBSPACE** command. In Version 7.1, the updates for this operation include distributing data across all database directories and then reclaiming unused space and returning it to the system. Because redistribution operations take considerable system resources, plan ahead when you want to add space to the database. You must complete the process while the server is not handling a heavy load.

For detailed requirements, see the **EXTEND DBSPACE** command in the *Administrator's Reference*.

As part of these enhancements, a new server utility, **DSMSERV EXTEND DBSPACE**, is available to perform the same function as the **EXTEND DBSPACE** command, while the server is offline.

Related tasks:

"Increasing the size of the database" on page 662

Server components that are not delivered with Version 7.1

Tivoli Monitoring for Tivoli Storage Manager and the Administration Center are not delivered with Tivoli Storage Manager Version 7.1.

Tivoli Monitoring for Tivoli Storage Manager

You can use Tivoli Monitoring for Tivoli Storage Manager V6.3.4 with any Tivoli Storage Manager V 5.5 or later server. For more information about installing and configuring Tivoli Monitoring for Tivoli Storage Manager, see Technote 1649883 at: http://www.ibm.com/support/docview.wss?uid=swg21649883

Administration Center

The Administration Center is not delivered in V7.1 or later releases. You can use the V6.3.4 Administration Center with any Tivoli Storage Manager V6.3 or later server, including to automatically update backup-archive clients. For more information about the Tivoli Storage Manager Administration Center, see the Tivoli Storage Manager V6.3 Information Center at: http://pic.dhe.ibm.com/infocenter/tsminfo/v6r3/index.jsp

Restriction: The Operations Center cannot be installed on HP-UX or Oracle Solaris systems. However, you can use the Operations Center to manage Tivoli Storage Manager V6.3.4 or later servers that run on HP-UX or Oracle Solaris systems.

New best practices

Learn about new best practices that you can implement for the Tivoli Storage Manager server and client.

Duplicate identification processes and memory requirements

To get the best performance for your data deduplication processes, you can specify an increased number of duplicate identification processes.

When you create a storage pool for data deduplication, you can now specify 0 - 50 parallel processes for server-side duplicate identification.

You can use additional memory to optimize the frequent access of deduplicate extent information that is stored in the Tivoli Storage Manager database.

Server startup and setting ulimits

Before you start the Tivoli Storage Manager server on an IBM AIX®, HP-UX, Linux, or Solaris operating system, verify access rights and user limits, also known as ulimits.

If you do not verify user limits, the server might become unstable or fail to respond. Set the ulimit value as 65536 for the maximum number of open files. Ensure that the system-wide limit is at least the value of the ulimit.

Configuring the server to prevent issues with database backups

In Tivoli Storage Manager Version 7.1, it is no longer necessary to set the API password when you manually configure the server. If you set the API password during the manual configuration process, attempts to back up the database might fail.

Part 1. Tivoli Storage Manager basics

© Copyright IBM Corp. 1993, 2013

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 various vendors, with various 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 include a command-line administrative client and a web-based interface that is called the Operations Center. With Tivoli Storage Manager you can 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-access 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 through a local area network (LAN) or a 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 that is installed and is registered with the server.

Network-attached storage (NAS) file servers can also be client nodes, but when you use 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

© Copyright IBM Corp. 1993, 2013

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 can also be installed on the same system as the 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 (by using NDMP)
The server can use the Network Data Management Protocol (NDMP) to back up and restore file systems that are stored on a network-attached storage (NAS) file server. The data on the NAS file server is backed up to a tape library. The Tivoli Storage Manager software does not need 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. For more information, including details about supported NAS file servers, see Chapter 9, "Using NDMP for operations with NAS file servers," on page 199.

Application client
Application clients allow users to initiate 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 initiate 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 programming interface (API)
The API can help 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 operating systems. 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 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.

For information about supported operating systems for clients, see the IBM Tivoli Storage Manager website 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 that 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 that are based on information that is provided in administrator-defined policies, and manages devices and media that are based on information that is 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.

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 more.

You can have multiple policies and assign the different policies as needed to specific clients, or 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 and 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 471. You can modify this policy and define policies.

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

Back up and restore
The backup process copies data from client workstations to server storage to protect against data-loss. The server retains versions of a file according to policy, and replaces older versions of the file with newer versions. The policy includes the number of versions of files to keep and the retention time for the versions.

A client can restore any version of a file that is kept on the Tivoli Storage Manager server.

Archive and retrieve
The archive process copies data from client workstations to the Tivoli Storage Manager server 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, which is created on the server from the most recently backed-up files.

You can copy a backup set onto compatible portable media, which you can take directly to the client for rapid recovery without the use of a network and communicating with the Tivoli Storage Manager server.

Migration and recall
Migration is a function of the Tivoli Storage Manager for Space

Management program. Migration can free up client storage space by copying files from workstations to the Tivoli Storage Manager server. On the client, the Tivoli Storage Manager for Space Management program replaces the original file with a stub file that points to the original file on the Tivoli Storage Manager server. Files can be recalled to the workstations when needed.

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

A policy determines when files are considered for automatic migration. On UNIX or Linux systems that support the Tivoli Storage Manager for Space Management program, the policies determine whether files must be backed up to the server before it is 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 shows how a policy is part of the Tivoli Storage Manager process for storing client data.

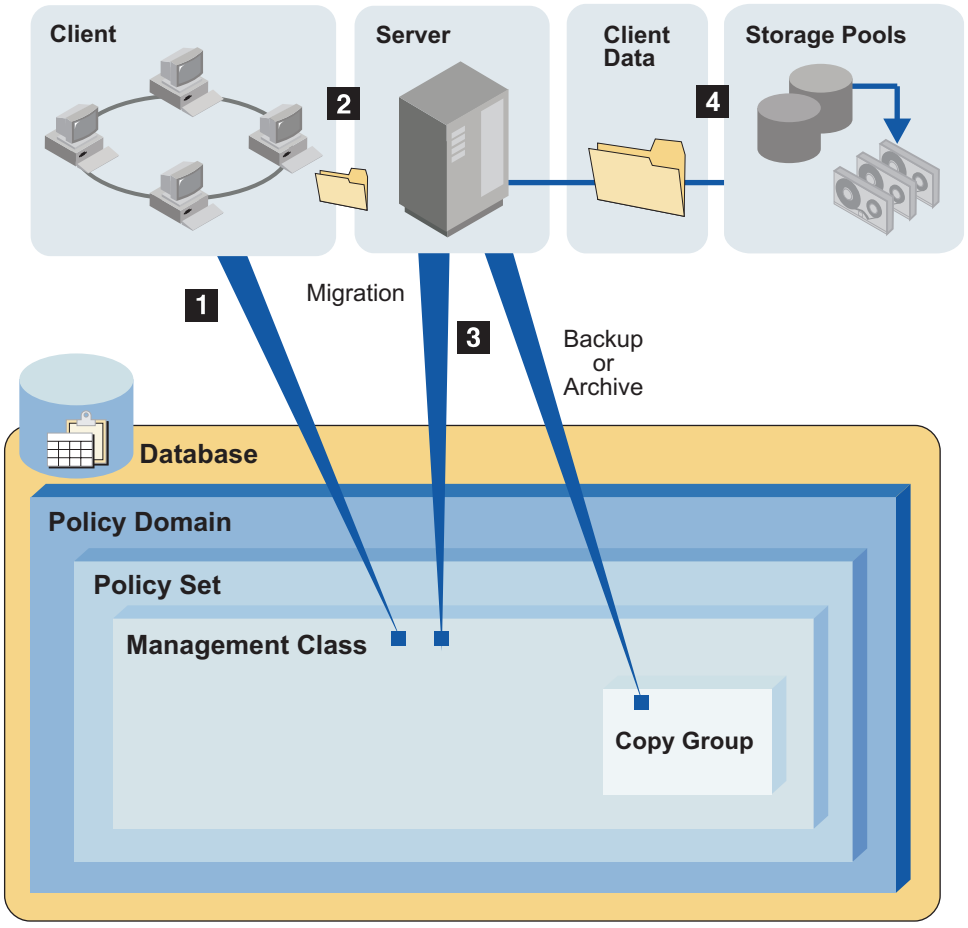


Figure 1. How IBM Tivoli Storage Manager Controls Backup, Archive, and Migration Processes

The steps in the process are as follows:

- 1 A client initiates a backup, archive, or migration operation. The file is bound to a management class. The management class is either the default or one specified for the file in the client options file or client options set.
- 2 Based on information in the management class, the client sends the file and file information to the server. The information that is sent depends on the client operation that was performed.
- 3 The server checks the management class that is bound to the file to determine the destination storage pool where the file is stored. For backed-up and archived files, the destination is stored in the backup and archive copy groups within management classes. For space-managed files, the destination is stored directly in the management class.

  The storage pool can be a group of disk volumes, or tape 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 might 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 the storage pools until they expire and expiration processing occurs, or until they are deleted on the server. A file expires because of criteria that are set in the policy. For example, the criteria include the number of file versions that are allowed and the number of days that have elapsed since a file was deleted from a client-node file system. If data retention protection is activated, an archive object cannot be inadvertently deleted.

For information about assigning storage destinations in copy groups and management classes, and on binding management classes to client files, see Chapter 13, “Implementing policies for client data,” on page 469.

For information about managing the database, see Chapter 19, “Managing the database and recovery log,” on page 633.

For information about storage pools and storage pool volumes, see Chapter 10, “Managing storage pools and volumes,” on page 235.

For information about event-based policy, deletion hold, and data retention protection, see Chapter 13, “Implementing policies for client data,” on page 469.

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 1 shows some examples of the protection options.

Table 1. 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.

Table 1. Examples of meeting your goals with Tivoli Storage Manager (continued)

For this goal...	Do this...
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.
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 *instant archive*.
Provide the ability to restore data to a point in time.	Use the backup-archive client to regularly perform incremental backups (either manually or automatically through schedules). Then do one of the following: • 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 522 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 537 for details.
Save a set of files and directories before making significant changes to them.	Use the backup-archive client to archive the set of files and directories. 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.

Table 1. Examples of meeting your goals with Tivoli Storage Manager (continued)

For this goal...	Do this...
Manage a set of related files, which are not in the same file system, with the same backup, restore, and server policies.	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 *Backup-Archive Clients Installation and User's Guide* for details.
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 IBM Tivoli Storage FlashCopy Manager 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.
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 15, "Scheduling operations for client nodes," on page 553.

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 12.

Table 2 on page 10 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 2. 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 *not* 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 486 and the *Backup-Archive Clients Installation and User's Guide*.
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 *not* 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 488 and the *Backup-Archive Clients Installation and User's Guide*.
Adaptive subfile backup	A backup method that backs up only the *parts* 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. The process works with either the standard progressive incremental backup or with selective backup. Applicable to clients on Windows systems.	Maintains backups of data while minimizing connect time and data transmission for the backup of mobile and remote users.	The base file plus a maximum of one subfile is restored to the client.	See "Enabling clients to use subfile backup" on page 546 and the *Backup-Archive Clients Installation and User's Guide*.
Journal-based backup	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.	Reduces the amount of time required for backup. The files eligible for backup are known before the backup operation begins. Applicable to clients on AIX and Windows systems, except Windows 2003 64-bit IA64.	Journal-based backup has no effect on how files are restored; this depends on the type of backup performed.	See the *Backup-Archive Clients Installation and User's Guide*.

Table 2. Summary of client operations (continued)

Type of operation	Description	Usage	Restore options	For more information
Image backup	Full volume backup. Nondisruptive, on-line backup is possible for Windows clients by using the Tivoli Storage Manager snapshot function.	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.	The entire image is restored.	See "Policy for logical volume backups" on page 517 and the *Backup-Archive Clients Installation and User's Guide*.
Image backup with differential backups	Full volume backup, which can be followed by subsequent differential backups.	Used only for the image backups of NAS file servers, performed by the server using NDMP operations.	The full image backup plus a maximum of one differential backup are restored.	See Chapter 9, "Using NDMP for operations with NAS file servers," on page 199.
Backup using hardware snapshot capabilities	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 FlashCopy Manager product then uses the volume copies to back up the database volumes.	Implements high-efficiency backup and recovery of business-critical applications while virtually eliminating backup-related downtime or user disruption on the database server.	Details depend on the hardware.	See the documentation for Tivoli Storage FlashCopy Manager.
Group backup	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. Applicable to clients on UNIX and Linux systems.	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. A group can be included in a backup set.	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.	See the *Backup-Archive Clients Installation and User's Guide*.

Table 2. Summary of client operations (continued)

Type of operation	Description	Usage	Restore options	For more information
Archive	The process creates a copy of files and stores them for a specific time.	Use for maintaining copies of vital records for legal or historical purposes. **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.	The selected version of the file is retrieved on request.	See "Archive" on page 489 and the *Backup-Archive Clients Installation and User's Guide*.
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 537.

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 2 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 “Protecting client data” on page 870.

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 “Protecting the database and infrastructure setup files” on page 858.

These backups can become part of a disaster recovery plan, created automatically by the disaster recovery manager. See Chapter 30, “Disaster recovery manager,” on page 979.

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 56.

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 353.

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 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 primary storage pools" on page 871.

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 537.

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 402.

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 for server storage

With Tivoli Storage Manager, you can use direct-attached storage devices and network-attached devices for server storage.

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 can 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 39.

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 256.

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 506.

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 described in the following topics.

Interfaces to Tivoli Storage Manager

You can work with many different applications by using Tivoli Storage Manager interfaces.

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*.
- Web interfaces for server administration and for the backup-archive client

  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 Oracle 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 programming interface

  For more information, see the *IBM Tivoli Storage Manager Using the Application Program Interface*.
- Access to information in the server 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 784.

© Copyright IBM Corp. 1993, 2013

Storage configuration and management

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

The Tivoli Storage Manager 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 storage area network (SAN). When you configure and manage the storage, consider the following tasks:

- Selecting the devices and media that will form the server storage, and whether library drives want to 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 by using NDMP to protect data on network-attached storage (NAS) file servers when backups occur 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 by using an external media manager to do the management outside of the Tivoli Storage Manager product.

For an introduction to storage concepts, see Chapter 3, “Storage device concepts,” on page 39.

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 235

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 101

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 “Defining device classes” on page 174.

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 129

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.

About this task

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 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.

About this task

Complete the following steps:

Procedure

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.

About this task

To label tapes with an automated library:

Procedure

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.

About this task

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 may want to ensure that migration occurs at a time that is best for your situation. You can control when migration occurs by setting migration thresholds with the **DEFINE STGPOOL** command.

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

To migrate from a disk storage pool to a tape storage pool, devices must be allocated and tapes must be mounted. Mount messages are directed to the console message queue and to any administrative client that is started with either the mount mode or console mode option.

See "Migrating disk storage pools" on page 269 and the *Administrator's Reference* for more information.

Techniques for managing your storage pools and volumes

Backed-up, archived, and space-managed files are stored in groups of volumes that are called storage pools.

About this task

You can use the following techniques to manage your storage pool data:

Collocation
The server can keep data that belongs to a single client node, group of client nodes, or client file space assigned to as few volumes as possible. When you restore collocated files, fewer media mounts is needed, however, when you back up files from different clients, more mounts are needed.

Reclamation
Reclamation makes the fragmented space on volumes usable again by moving any remaining active files from one volume to another volume, thus making the original volume available for reuse.

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. You can configure the storage pool to simultaneously write to copy storage pools when client data is written to the primary 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 have to position past inactive files that do not have to be restored.

An active-data pool that uses removable media, such as tape, reduces 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 saves 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 can 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 that are associated with the pools. Tivoli Storage Manager can migrate client files through this hierarchy to ensure the most efficient use of 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 235.

Increasing disk storage-pool sizes

When the server is installed, Tivoli Storage Manager creates default disk storage pools. You must define volumes for these storage pools.

About this task

At server installation, the following storage pools are created:

- BACKUPPOOL
- ARCHIVEPOOL
- SPACEMGPOOL

Before defining storage-pool volumes, estimate how much client data you want to store. Consider defining at least as many volumes as you need to hold one day of client backups. To format and define storage pool volumes, use the **DEFINE VOLUME** command.

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 472.
- For automating client operations, see Chapter 15, "Scheduling operations for client nodes," on page 553.
- 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 13, "Implementing policies for client data," on page 469.

For more information about these tasks, see Chapter 16, "Managing schedules for client nodes," on page 561

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 IBM 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 415
- Chapter 12, "Managing client nodes," on page 425

Allowing subfile backups

Restriction: Subfile backups are only applicable to Windows clients.
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 14, "Managing data for client nodes," on page 529.

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 14, "Managing data for client nodes," on page 529.

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 416.

You can ensure that only authorized administrators and client nodes are communicating with the server by requiring passwords. Passwords can authenticate with an LDAP directory server or the Tivoli Storage Manager server. Most password-related commands work for both kinds of servers. The **PASSEXP** and **RESET PASSEXP** commands do not work for passwords that authenticate with an LDAP directory server. You can use the LDAP directory server to give more options to your passwords, independent of the Tivoli Storage Manager server.

Restriction: The Tivoli Storage Manager backup-archive client must be at version 6.4.0 or later to use passwords that authenticate with an LDAP directory server.

Whether you store your passwords on an LDAP directory server, or on the Tivoli Storage Manager server, you can set the following requirements for passwords:

- Minimum 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 stops the client node from accessing the server. The limit can be set on the Tivoli Storage Manager server, and on the LDAP directory server.

Important: The invalid password limit is for passwords that authenticate with the Tivoli Storage Manager server and any LDAP directory servers. Invalid password attempts can be configured on an LDAP directory server, outside of the Tivoli Storage Manager server. But the consequence of setting the number of invalid attempts on the LDAP directory server might pose some problems. For example, when the **REGISTER NODE** command is issued, the default behavior is to name the node administrator the same name as the node. The LDAP server does not recognize the difference between the node "NODE_Q" and the administrator "NODE_Q". The node and the administrator can authenticate to the LDAP server if they have the same password. If the node and administrator have different passwords, the authentication fails for either the node or administrator. If the node or the administrator fail to logon consistently, their IDs are locked. You can avoid this situation by issuing the **REGISTER NODE** command with `USERID=`*`userid`* or `USERID=NONE`.

Tivoli Storage Manager provides several ways to manage security. You can control the authority of administrators. An organization can name a single administrator or distribute the workload among a number of administrators and grant them different levels of authority. For better security when clients connect across a firewall, you can control whether clients can initiate contact with the server for scheduled operations. Server options can be set to keep client and administrative traffic on separate server ports.

Related concepts:

"Managing Tivoli Storage Manager administrator IDs" on page 834

"Managing passwords and logon procedures" on page 843

Related reference:

Chapter 27, "Managing Tivoli Storage Manager security," on page 821

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.

About this task

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 13, "Implementing policies for client data," on page 469.

Creating new policies

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

About this task

For example, it specifies that Tivoli Storage Manager retains up to two backup versions of any file that exists on the client (see Chapter 13, "Implementing policies for client data," on page 469 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.

Procedure

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 policyset newdomain standard
   activate policyset 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.

Results

For more information about the default policy and establishing and managing new policies, see Chapter 13, “Implementing policies for client data,” on page 469.

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 458 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.

Related tasks:

"Scheduling a client operation" on page 554

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 425 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 /engspec/*.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 482.

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 579. For suggestions about the day-to-day tasks required to administer the server, see Chapter 17, “Managing server operations,” on page 579.

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 **SETOPT** command. 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.

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.

About this task

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.

Results

For more information about automating Tivoli Storage Manager operations, see Chapter 18, "Automating server operations," on page 609.

Modifying a maintenance script

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

Before you begin

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**.

About this task

Perform the following tasks to modify a maintenance script:

Procedure

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 19, "Managing the database and recovery log," on page 633.

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 database and storage, and resource usage. By monitoring selected information, you can provide reliable services to users while you are making the best use of available resources.

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:

- Part 5, "Monitoring operations," on page 765
- Chapter 22, "Daily monitoring tasks," on page 767
- "Using SQL to query the IBM Tivoli Storage Manager database" on page 784
- Chapter 26, "Logging IBM Tivoli Storage Manager events to receivers," on page 797

Tivoli Storage Manager server networks

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

About this task

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 691.
- Route commands to multiple servers while logged on to one server. See "Routing commands" on page 716.
- 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 811.
- 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 722 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 30, "Disaster recovery manager," on page 979 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 722 for details.
- To simplify password management, have client nodes and administrators authenticate their passwords on multiple servers using an LDAP directory server. See "Managing passwords and logon procedures" on page 843.

Exporting and importing data

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

About this task

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 21, "Exporting and importing data," on page 731.

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.

About this task

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 that is 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, always encrypt sensitive data by using the Tivoli Storage Manager client or the storage device, unless the storage media is physically secured. See Part 6, "Protecting the server," on page 819 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 30, "Disaster recovery manager," on page 979.

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.

About this task

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
- Keeping the `cert.kdb` file in a safe place to ensure that SSL (Secure Sockets Layer) is secure for authenticating passwords

For information about protecting the server with these measures, see Chapter 28, "Protecting and recovering the server infrastructure and client data," on page 857.

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 28, "Protecting and recovering the server infrastructure and client data," on page 857.

Part 2. Configuring and managing storage devices

To make storage devices available to a server, you must attach the devices, configure device drivers, and create Tivoli Storage Manager objects that represent those devices. To use the devices to store client data, you also create objects representing storage pools and storage-pool volumes.

© Copyright IBM Corp. 1993, 2013

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 40
"Tivoli Storage Manager storage objects" on page 40
"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
"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.

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, "Attaching devices for the server," on page 81
Configure devices to use with Tivoli Storage Manager, using detailed scenarios of representative device configurations.	Chapter 6, "Configuring storage devices," on page 101
Plan, configure, and manage an environment for NDMP operations	Chapter 9, "Using NDMP for operations with NAS file servers," on page 199
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 129
Define and manage device classes.	"Defining device classes" on page 174

© Copyright IBM Corp. 1993, 2013

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
- Removable file devices
- 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 3 on page 63. For details and updates, see the Tivoli Storage Manager device support website:

http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.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, ACSLS, or VTL libraries. The physical library is controlled by the Tivoli Storage Manager server that is configured as a library manager. Tivoli Storage Manager servers that use the SHARED library type are library clients to the library manager server. Shared libraries reference a library manager.

Automated cartridge system library software libraries

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

The StorageTek software performs the following functions:

- Mounts volumes, both private and scratch
- Dismounts volumes
- Returns library volumes to scratch status

The ACSLS software selects an 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 your StorageTek documentation. If you are using an ACSLS library and have a firewall enabled, refer to the *StorageTek Automated Cartridge System Library Software Administrator's Guide, Version 8.1* (April 2012 edition).

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 101

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

"Managing server requests for media" on page 149

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 Oracle 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 101

Virtual tape libraries

A *virtual tape library* (VTL) is a hardware component that can emulate a tape library while using a disk as the underlying storage hardware.

Using a VTL, you can create variable numbers of drives and volumes because they are only logical entities within the VTL. The ability to create more drives and volumes increases the capability for parallelism, giving you more simultaneous mounts and tape I/O.

VTLs use SCSI and Fibre Channel interfaces to interact with applications. Because VTLs emulate tape drives, libraries, and volumes, an application such as Tivoli Storage Manager cannot distinguish a VTL from real tape hardware unless the library is identified as a VTL.

For information about configuring a VTL library, see "Managing virtual tape libraries" on page 114.

349X libraries

A 349X library is a collection of drives in an IBM 3494. Volume mounts and dismounts 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.

Categories in an IBM 3494 library

The library manager that is built into the IBM 3494 library tracks the category number of each volume in the library. A single category number identifies all volumes that are 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 might 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 that shares 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. For more information about private, scratch, and scratch WORM volumes, see "Tivoli Storage Manager volumes" on page 50.

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 the category number for a volume. For Tivoli Storage Manager, you use the **CHECKIN LIBVOLUME** command.

For more information about checking in new volumes, see "Checking new volumes into a library" on page 133.

349X library objects contain only one device type (3590 or 3592) of drives. Thus, if you have 3590 and 3592 drives in your 349X library, you must define two library objects: one for your 3590 drives and one for your 3592 drives. Each of these library objects must have the same device parameter when their paths are defined.

The server reserves two category numbers in each 3494 library that it accesses: one for private volumes and one for scratch volumes. For example, suppose you want to define a library object for your 3592 drives:

```
define library my3494 libtype=349x privatecategory=400 scratchcategory=401
wormscratchcategory=402
```

For this example, the server uses the following categories in the new `my3494` library:

- 400 (X'190') Private volumes
- 401 (X'191') Scratch volumes
- 402 (X'192') WORM scratch volumes

Note: The default values for the categories might 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 might try to use the same volume, and cause a corruption or loss of data.

For information about configuring a 349X library, see:

Chapter 6, "Configuring storage devices," on page 101

External libraries

An external library is a collection of drives that are 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 Oracle StorageTek library used in conjunction with an external library manager (ELM), like Gresham's EDT-DistribuTAPE, is a type of external library. IBM Tape Storage Library Manager (TSLM) is another example.

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 1051.

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 or virtual tape libraries (VTLs), how often a tape drive is cleaned, and whether the drive is online.

Tivoli Storage Manager drives include tape drives that can stand alone or that can be part of an automated library. Supported removable media drives also include removable file devices such as rewritable CDs.

Device class

Each device that is defined to Tivoli Storage Manager is associated with one device class. The device class 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 various removable media types along with FILE, CENTERA, and SERVER.

A device class for a tape 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 systems" 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.

The GENERICTAPE device type is provided to support certain devices that are not supported by the Tivoli Storage Manager server.

For more information about supported removable media device types, see "Defining device classes" on page 174 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 or one server process 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 ACTIVEDATA
  – 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 systems" on page 69.

Files on sequential volumes (CENTERA)

The CENTERA device type defines the EMC Centera storage device. It can be used like any standard storage device from which files can be backed up and archived as needed.

The Centera storage device can also be configured with the Tivoli Storage Manager server to form a specialized storage system that protects you from inadvertent deletion of mission-critical data such as emails, trade settlements, legal documents, and so on.

The CENTERA device class creates logical sequential volumes for use with Centera storage pools. These volumes share many of the same characteristics as FILE type volumes. With the CENTERA device type, you are not required to define library or drive objects. CENTERA volumes are created as needed and end in the suffix "CNT."

Multiple client retrieve sessions, restore sessions, or server processes can read a volume concurrently in a storage pool that is associated with the CENTERA device type. In addition, one client session or one server process can write to the volume while it is being read. Concurrent access improves restore and retrieve performance because two or more clients can have access to the same volume at the same time.

The following server processes can share read access to Centera volumes:

- EXPORT NODE
- EXPORT SERVER
- GENERATE BACKUPSET

The following server processes cannot share read access to Centera volumes:

- AUDIT VOLUME
- DELETE VOLUME

For more information about the Centera device class, see "Defining device classes for CENTERA devices" on page 194. For details about Centera-related commands, refer to the *Administrator's Reference*.

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 722.

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.

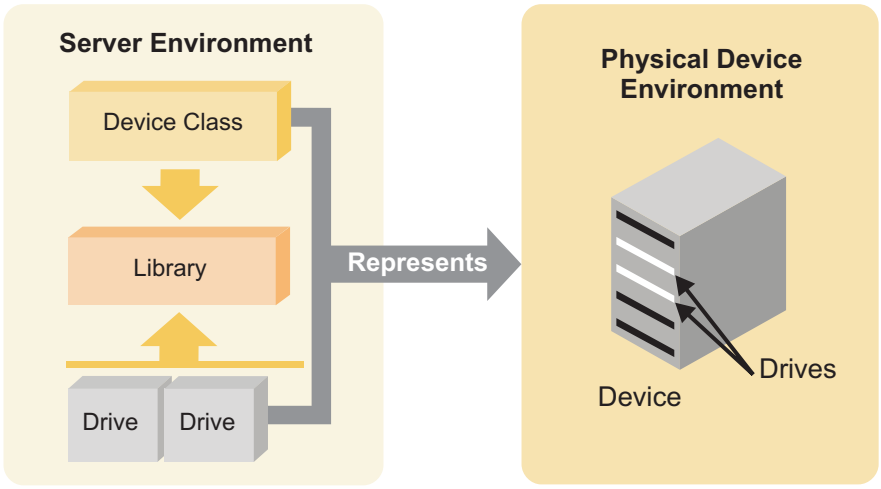


Figure 2. Removable media devices are represented by a library, drive, and device class

- For more information about the drive object, see:
  - "Managing drives" on page 153
  - "Defining drives" on page 170
- For more information about the library object, see:
  - "Managing libraries" on page 151
  - "Defining libraries" on page 169
- For more information about the device class object, see "Defining device classes" on page 174.

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.

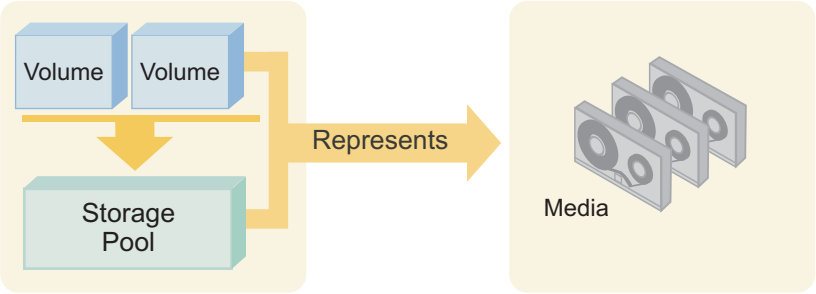


Figure 3. Relationships of storage pool volumes, storage pools, and media

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 251

"Preparing volumes for sequential-access storage pools" on page 252

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.

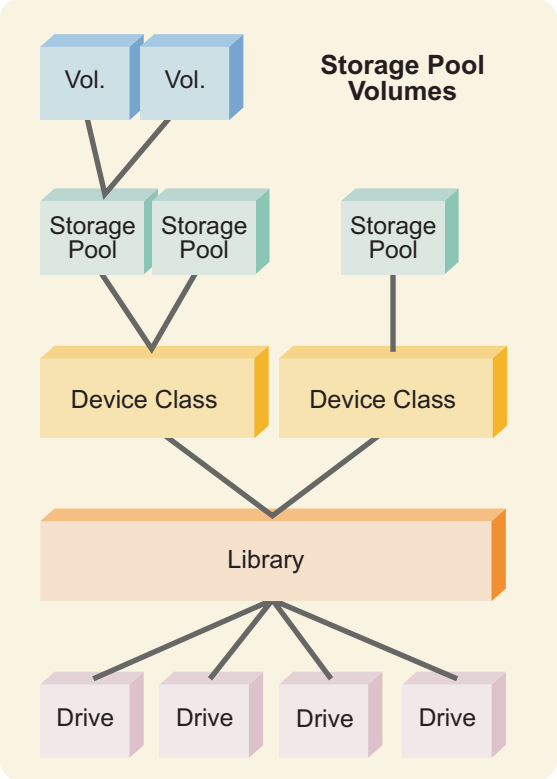


Figure 4. Relationships between storage and device objects

For information about defining storage pool and volume objects, see Chapter 10, "Managing storage pools and volumes," on page 235.

For information about configuring volumes for random access see"Configuring random access volumes on disk devices" on page 76.

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 172
"Managing paths" on page 165

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:

- "Configuring library sharing" on page 110
- "Using virtual volumes to store data on another server" on page 722
- *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 252. 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 145
- 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 133

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 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 118 and "Configuring library sharing" on page 110.

For information about the categories of libraries that are supported by Tivoli Storage Manager, see "Libraries" on page 40.

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).

By using Tivoli Storage Manager in a SAN, you can take advantage of the following functions:

- Sharing storage devices among multiple Tivoli Storage Manager servers.

  Devices that use the GENERICTAPE device type are not included.
- Moving Tivoli Storage Manager client data directly to storage devices (LAN-free data movement) by configuring a storage agent on the client system.

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

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

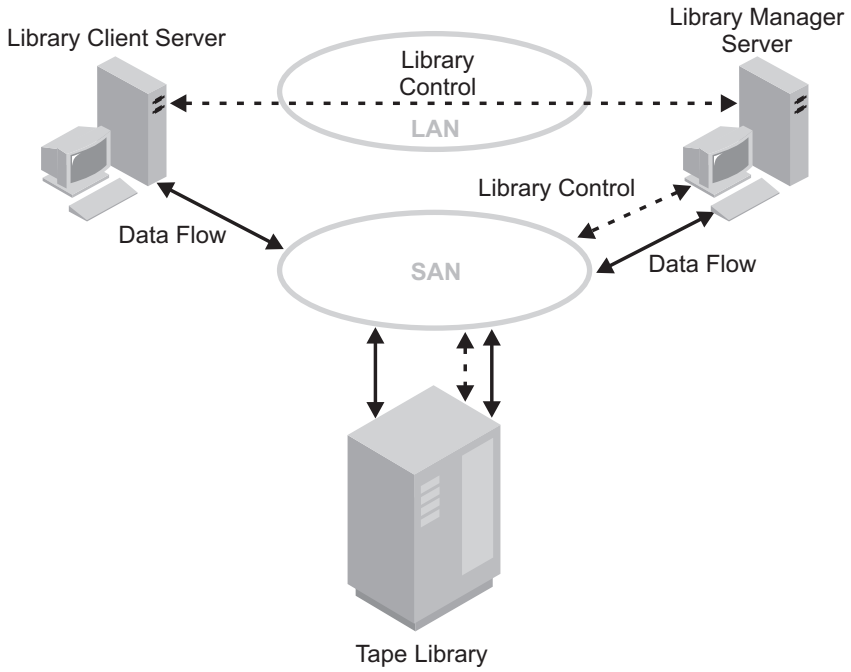


Figure 5. Library sharing in a storage area network (SAN) configuration. The servers communicate over the LAN. The library manager controls the library over the SAN. The library client stores data to the library devices over the SAN.

When Tivoli Storage Manager servers share a library, the server that is defined as the library manager controls device operations. These operations include mount, dismount, volume ownership, and managing library inventory. Tivoli Storage Manager servers that are defined as 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 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 completes a single mount or dismount operation at a time. The library manager completes all mount operations to provide this serialization.

Related tasks:

"Configuring library sharing" on page 110

Chapter 5, "Attaching devices for the server," on page 81

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.

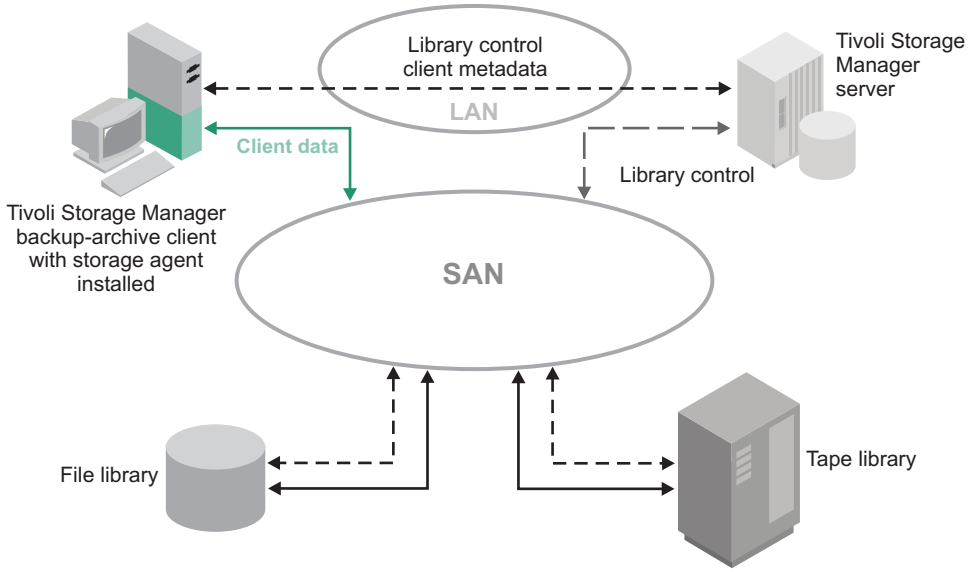


Figure 6. LAN-Free data movement. Client and server communicate over the LAN. The server controls the device on the SAN. Client data moves over the SAN to the device.

LAN-free data movement requires the installation of a storage agent on the client system. 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 that uses 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 handles 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:

- Centera storage devices cannot be targets for LAN-free operations.
- 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.

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 that is 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 onsite disaster.

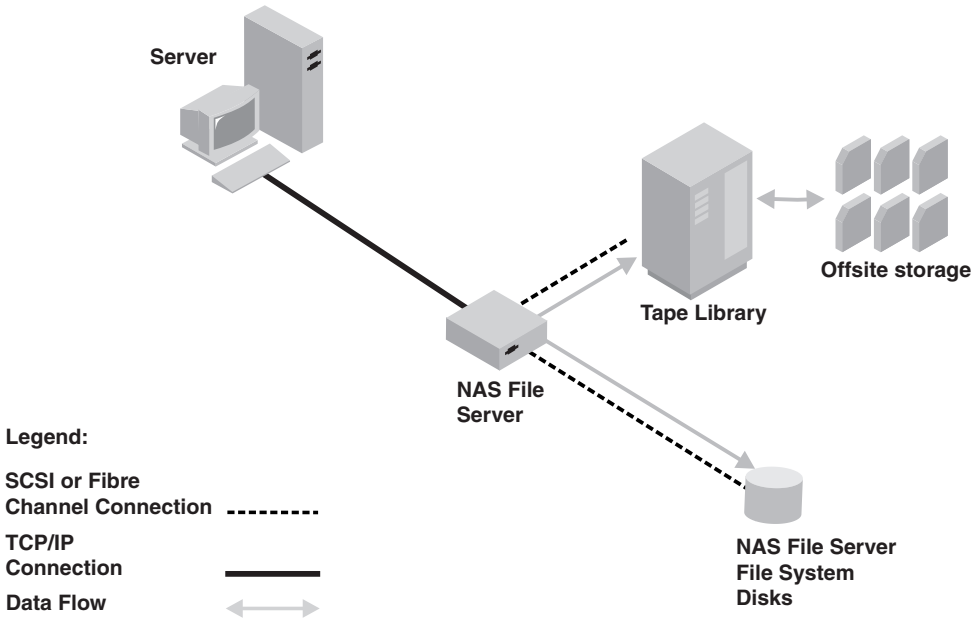


Figure 7. Library device directly attached to a NAS file server

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 on page 56.) 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.

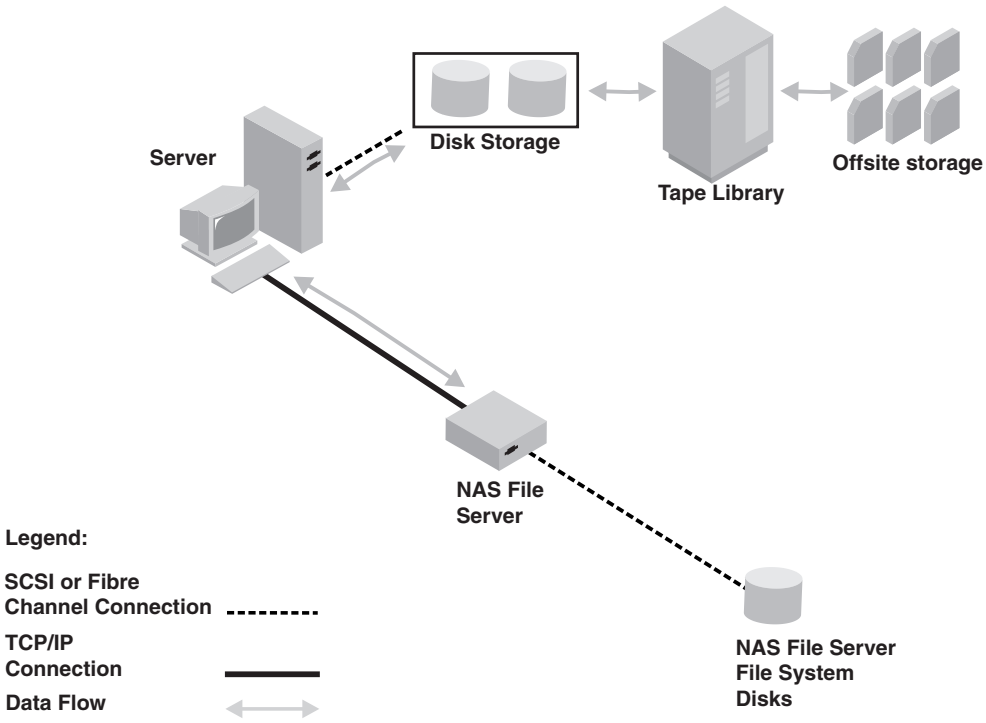


Figure 8. NAS file server toTivoli Storage Manager storage-pool hierarchy

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 about NDMP file-level restores, see "NDMP file-level restoration" on page 57.

Note:

- A Centera storage device cannot be a target for NDMP operations.
- 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 209.

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:

"Example: Configure a SCSI or virtual tape library with multiple drive device types" on page 104

"Example: Configure a 3494 library with multiple drive device types" on page 109

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)
- LTO Ultrium Generation 5 (LTO5) with LTO Ultrium Generation 6 (LTO6)

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.
4. LTO Ultrium Generation 6 drives can only read Generation 4 media. If you are mixing Ultrium Generation 4 with Ultrium Generation 6 drives and media in a single library, you must mark the Generation 4 media as read-only, and all Generation 4 scratch volumes must be checked out.

To learn more about additional considerations when mixing LTO Ultrium generations, see “Defining LTO device classes” on page 188.

When using Tivoli Storage Manager, you cannot mix drives that are 3592 drive generations. Use one of three special configurations. For details, see "Defining 3592 device classes" on page 180.

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 LTO formats. Multiple storage pools and their device classes of different types can point to the same library that 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 different drive generations that are read but not write compatible within the same library, separate storage pools for each must be used.

Library sharing

Library sharing or tape resource sharing allows multiple Tivoli Storage Manager servers to use the same tape library and drives on a storage area network (SAN) and to improve backup and recovery performance and tape hardware asset utilization.

When Tivoli Storage Manager servers share a library, one server is set up as the library manager and controls library operations such as mount and dismount. The library manager also controls volume ownership and the library inventory. Other servers are set up as library clients and use server-to-server communications to contact the library manager and request resources.

Library clients must be at the same or a lower version than the library manager server. A library manager cannot support library clients that are at a higher version. For example, a version 6.2 library manager can support a version 6.1 library client but cannot support a version 6.3 library client.

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. Managing media involves creating a policy to expire data after a certain time or under certain conditions, moving valid data onto new media, and reusing the empty media.

In addition to information about storage pool volumes, volume history contains information about tapes that are 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 that contain client data backups.

Figure 9 shows a typical lifecycle for removable media. The numbers (such as *1*) refer to numbers in the figure.

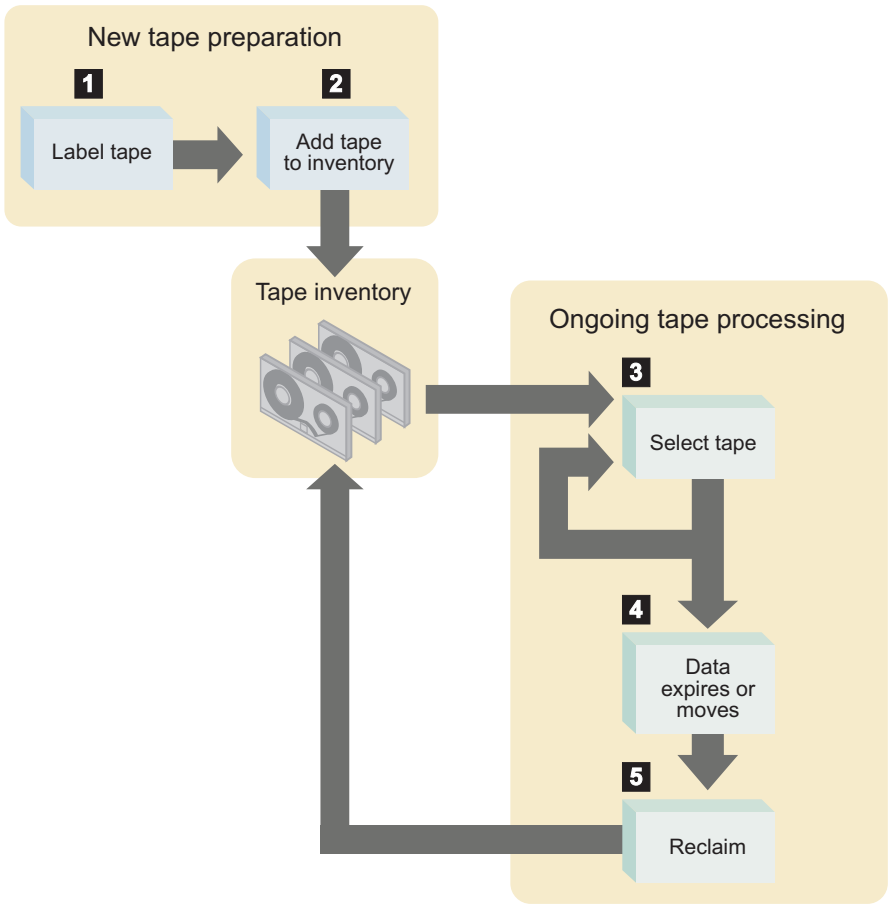


Figure 9. Simplified view of the lifecycle of a tape

1. You label *1* and check in *2* the media. Checking media into a manual library means storing them (for example, on shelves). Checking media into an automated library involves adding them to the library volume inventory.

   For more information, see "Labeling removable media volumes" on page 130
2. If you plan to define volumes to a storage pool associated with a device, 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. It can be either the default management class for the policy set, or the class that is specified by the client in the client's include/exclude list or file.
   - The storage pool that is 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 was reached, when the scratch volumes are selected.

- Whether collocation is enabled for that storage pool. When collocation is enabled, the server tries to place data on separate volumes. The data can come from different client nodes, groups of client nodes, file spaces, or groups of file spaces. For more information, see "Keeping client files together using collocation" on page 353.

Figure 10 shows more detail about the policies and storage pool specifications that govern the volume selection that is described in this step.

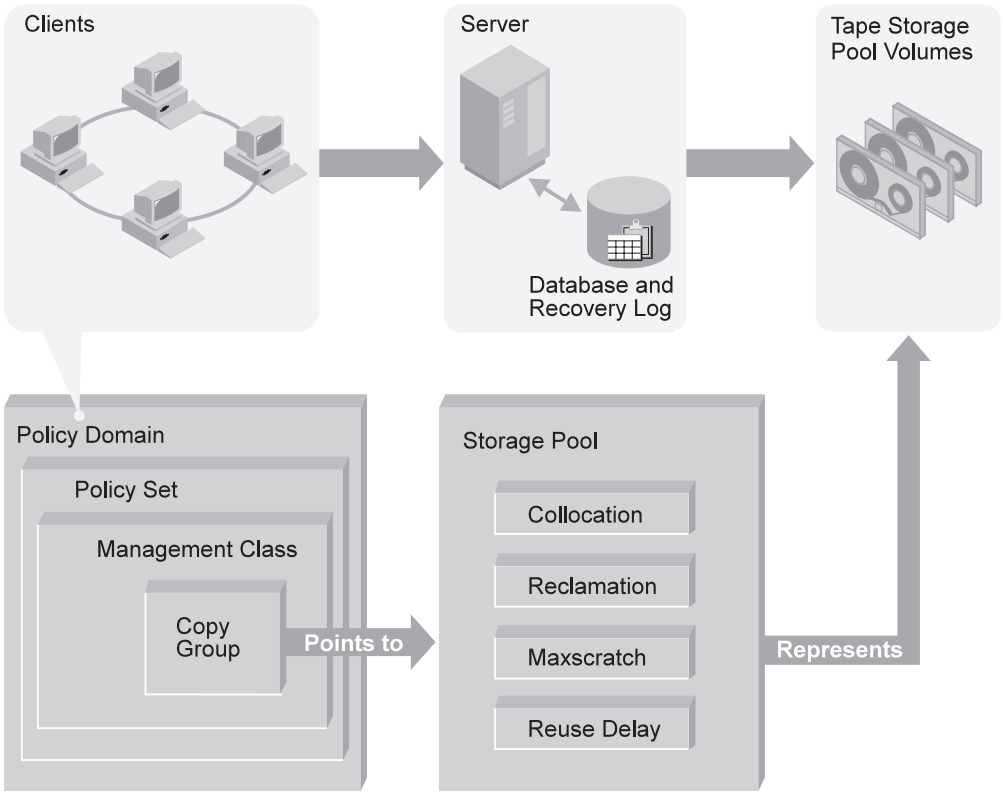


Figure 10. How Tivoli Storage Manager affects media use

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 470.
   - 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 that is 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 the server tries to keep data on a minimal number of removable media in a storage pool. The data can belong to a single client node, a group of client nodes, a file space, or a group of file spaces.

   If the volume becomes empty because all valid data either expires or is moved to another volume, the volume is available for reuse. It is not available for

reuse if a time delay was 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 is at its end-of-life.

   For volumes that you defined (private volumes), check the statistics on the volumes by querying the database. The statistics include these two entries:

   - The number of write passes on a volume (compared with the number of write passes that are recommended by the manufacturer)
   - The number of errors on the volume

   You must move any valid data off a volume that 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.

Definitions for storage devices

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

About this task

You can use the **PERFORM LIBACTION** command to simplify the process when you add devices to SCSI and VTL library types.

Table 3 summarizes the definitions that are required for different device types.

Table 3. Required definitions for storage devices

		Required Definitions			
Device	**Device Types**	**Library**	**Drive**	**Path**	**Device Class**
Magnetic disk	DISK	—	—	—	Yes See note
	FILE See note	—	—	—	Yes
	CENTERA	—	—	—	Yes
Tape	3590 3592 4MM 8MM DLT LTO NAS VOLSAFE GENERICTAPE ECARTRIDGE 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 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.

About this task

To map storage devices to device classes, use the information shown in Table 4.

Table 4. 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 "Defining device classes" on page 174 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.

About this task

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 5.

Table 5. 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 76

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.

About this task

Procedure

1. Determine which drives and libraries are supported by the server. For the most up-to-date list of supported devices and operating-system levels, see the Supported Devices website:

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

2. Determine which storage devices can be selected for use by the server. For example, determine how many tape drives you have that you can 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 that are 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 might 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 about device driver support, see:

   "Selecting a device driver" on page 83

4. Determine how to attach the devices to the server. For more information about attaching devices, see:

   "Attaching an automated library device to your system" on page 82

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 similar devices together 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, 349X, or VTL library. Each automated library device is a separate library.
   - Devices that are controlled by Oracle StorageTek Automated Cartridge System Library Software (ACSLS) must be part of a defined ACSLS 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 by using the storage of another Tivoli Storage Manager server, consider network bandwidth and network traffic. If your network resources constrain your environment, you might have problems with 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 might want to create a new labeling convention for media so that you can distinguish them from media that are 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.

The Table 6 table provides brief descriptions of these options.

Table 6. 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 ACSLS libraries to be locked.
ACSQUICKINIT	Allows a quick or full initialization of the ACSLS library.
ACSTIMEOUTX	Specifies the multiple for the built-in timeout value for ACSLS API.
ASSISTVCRRECOVERY	Specifies whether the server assists an IBM 3590 drive in recovering from a lost or corrupted Vital Cartridge Records (VCR) condition.
CHECKTAPEPOS	Specifies whether the server validates data position on tape. **Tip:** This option might conflict with storage area network (SAN) tape acceleration.
DRIVEACQUIRERETRY	Specifies how many times the server retries the acquisition of a drive in a library when there are no drives available after it acquires a mount point.
NOPREEMPT	Specifies whether the server allows certain operations to preempt other operations for access to volumes and devices. See “Preempting operations” on page 602 for details.
RESOURCETIMEOUT	Specifies how long the server waits for a resource before it cancels the pending acquisition of a resource. **Tip:** For proper management of shared library resources, consider setting the `RESOURCETIMEOUT` option at the same time limit for all servers in a shared configuration. For 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 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.

You can store the following types of data on magnetic disk devices:

- The database and recovery log
- Backups of the database
- Export and import data
- Client data that is backed up, archived, or migrated from client nodes. The client data is stored in storage pools.

See the following sections:

Tasks:
"Configuring random access volumes on disk devices" on page 76
"Configuring FILE sequential volumes on disk devices" on page 77
"Varying disk volumes online or offline" on page 78
"Cache copies for files stored on disk" on page 78
"Freeing space on disk" on page 79
"Scratch FILE volumes" on page 79
"Volume history file and volume reuse" on page 79

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 40.

Requirements for disk systems

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

Review the following Tivoli Storage Manager requirements for disk devices and compare them with information from your disk system vendor. A list of supported disk storage devices is not available. Contact the vendor for your disk system if you have questions or concerns about whether Tivoli Storage Manager requirements are supported. The vendor should be able to provide the configuration settings to meet these requirements.

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

© Copyright IBM Corp. 1993, 2013

system 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 must be 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 that is being retrieved varies depending on the way that an I/O system manages the writing of data. 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 systems at the remote site must also be met.

Database write operations must be nonvolatile for active and archive logs and DISK device class storage pool volumes. Data must be permanently committed to storage that is known toTivoli 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 after 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 systems 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. Data for the I/O operation can be lost if nonvolatile cache is used to safeguard I/O writes to a device and the nonvolatile cache is battery protected. If there is a power loss and power is not restored before the battery is exhausted, then data 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 system.

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 if 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 is lost, the Tivoli Storage Manager server at the remote location must be restored by conventional means by using Tivoli Storage Manager database and storage pool restores.

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.

Comparison of 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 depends on the operating system on which your Tivoli Storage Manager server is running.

Table 7 provides some general information about the characteristics of DISK devices (random access) and FILE devices (sequential access) and the benefits of each.

Table 7. Comparing random access and sequential access disk devices

Function	Random Access (DISK)	Sequential Access (FILE)	Notes®
Storage space allocation and tracking	Disk blocks	Volumes	Space allocation and tracking by blocks uses more database storage space, and requires 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 7. Comparing random access and sequential access disk devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Notes®
Client restore operations	One session per restore	Multiple concurrent sessions access different volumes simultaneously on both the server and the storage agent. Active versions of client backup data is 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 physically mount tapes and does not position past inactive files. For more information, see "Backing up primary storage pools" on page 871, and the information about client restore operations in the *Optimizing Performance guide*.
Available for use in LAN-free backup	Not available	Available for LAN-free backup using IBM General Parallel File System.	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 76. For more information about volumes on FILE devices, see "Configuring FILE sequential volumes on disk devices" on page 77.
Tivoli Storage Manager server caching (after files are 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 280.

Table 7. Comparing random access and sequential access disk devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Notes®
Recovery of disk space	When caching is enabled, the space occupied by cached files is reclaimed on demand by the server. When caching is disabled, the server recovers disk space immediately after all physical files are migrated or deleted from within an aggregate.	The server recovers disk space in a process called *reclamation*, 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.	For more information about reclamation, see "Reclaiming space in sequential-access storage pools" on page 363.
Aggregate reconstruction	Not available; the result is wasted space	Aggregate reconstruction occurs as part of the reclamation process. It is also available by using the **RECONSTRUCT** parameter on the **MOVE DATA** and **MOVE NODEDATA** commands.	An *aggregate* 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 364.
Available for use as copy storage pools or active-data pools	Not available	Available	Copy storage pools and active-data pools provide additional levels of protection for client data. For more information, see "Backing up primary storage pools" on page 871.
File location	Volume location is limited by the trigger prefix or by manual specification	FILE volumes use directories. A list of directories can be specified. If directories correspond with file systems, performance is optimized.	

Table 7. Comparing random access and sequential access disk devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Notes®
Restoring the database to an earlier level	See Notes	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 DSMSERV 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 875. For more information about reuse delay, see "Delaying reuse of volumes for recovery purposes" on page 874. For command syntax, refer to the *Administrator's Reference*.
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 269.
Storage pool backup	Performed by node and file space. 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 236.
Copying active data	Performed by node and file space. 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 236.

Table 7. Comparing random access and sequential access disk devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Notes®
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 896 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 353.
Shredding data	If shredding is enabled, sensitive data is destroyed after it is deleted from a storage pool. Write caching on a random access devices 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 533.
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 281.

File systems and raw logical volumes for random access storage

You can choose to use either files in a file system or raw logical volumes when defining random access storage pool volumes.

Random access storage pool volumes defined as raw logical volumes have the following advantages:

- The formatting of volumes is nearly instantaneous because the creation of a file is not needed.
- Many layers of the operating system can be bypassed, providing faster performance and lower CPU utilization.
- Fewer RAM resources are consumed because file system cache is not used.

One disadvantage of raw logical volumes is that there is no locking mechanism that prevents other applications or another Tivoli Storage Manager server instance from using the volume. This can cause data corruption or other problems. This risk can be minimized by defining the volume to only one instance of Tivoli Storage Manager on a particular system and restricting access to the device files to the operating system user ID used for Tivoli Storage Manager.

Configuring random access volumes on disk devices

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

About this task

Define storage pool volumes on disk drives that reside on the server system, not on remotely mounted file systems. Network attached drives can compromise the integrity of the data that you are writing.

Procedure

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 245 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. For example:

   Define a 21 MB volume for the ENGBACK1 storage pool 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
   ```

   Another option for preparing a volume is to create a raw partition on a formatted disk. Raw partitions generally provide the best performance, however a IBM Tivoli Storage Manager server that is running Solaris 10 does not support ZFS raw partitions. For details, see the format command for the operating system. When creating the partition with the `/etc/format` utility, do not include cylinder 0 (zero) in the partition that is intended for use as a raw partition. After you create the partition, define the volume to Tivoli Storage Manager.

   Note: The server cannot detect if data exists on a raw partition. Existing data can be destroyed if you define a volume to that partition.
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.
   - 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 247.

Related concepts:

"Disk devices" on page 45

Chapter 13, "Implementing policies for client data," on page 469

Related tasks:

"Defining storage pool volumes" on page 252

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.

Procedure

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.

   To store database backups or exports on FILE volumes, this step is all that you must 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.
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.

   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.
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 13, "Implementing policies for client data," on page 469 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 247.

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

   ```
   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 are exceeded.

For more information, see the *Administrator's Reference*.

Related tasks:

"Defining sequential-access disk (FILE) device classes" on page 184

"Step 1: Defining device classes for database backups" on page 859

"Planning for sequential media used to export data" on page 744

"Defining storage pool volumes" on page 252

"Preparing volumes for sequential-access storage pools" on page 252

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.

About this task

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:

```
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 is 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 that are 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.

Related tasks:

"Caching in disk storage pools" on page 280

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 information from the database about any client files that are no longer valid according to the policies you have set. For example, suppose that 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.
- Set the EXPINTERVAL server option and specify the interval so that expiration processing runs periodically.

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.

Related concepts:

"Securing sensitive client data" on page 533

Related reference:

"Running expiration processing to delete expired files" on page 506

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 that are used for these operations in the *volume history*. Tivoli Storage Manager does not allow you to reuse these volumes until you delete the volume information from the volume history.

To reuse volumes that were previously used for database backup or export, use the **DELETE VOLHISTORY** command.

Note: With Tivoli Storage Manager Extended Edition, the disaster recovery manager (DRM) function automatically deletes volume information during processing of the **MOVE DRMEDIA** command.

Related tasks:

"Protecting the volume history file" on page 866
Chapter 30, "Disaster recovery manager," on page 979

Chapter 5. Attaching devices for the server

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

About this task

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 to your system"
"Attaching an automated library device to your system" on page 82
"Selecting a device driver" on page 83
"Installing and configuring device drivers" on page 87

Attaching a manual drive to your system

You can attach a manual drive to your system to store your data on tapes.

About this task

Complete the following steps to attach a manual drive to your system:

Procedure

- For Fibre Channel (FC), Fibre Channel over Ethernet (FCoE), Serial Attached SCSI (SAS), or iSCSI adapters, complete the following steps:
  1. Install an adapter card and the associated drivers.
  2. Attach the device to the server system hardware by following the manufacture's instructions.
  3. Install the device drivers for attached tape devices.
- For SCSI adapters, complete the following steps:
  1. Install the SCSI adapter card, and the associated drivers.
  2. Determine the SCSI IDs that are available on the adapter card to which you are attaching the device. Find one unused SCSI ID for each drive.
  3. Set the SCSI ID for the drive to the unused SCSI IDs.
  4. Set switches on the back of the device or set the IDs on the operator's panel.

     For each device that is connected in a chain to a single SCSI bus through a Fibre Channel adapter card, you must configure it to have a unique SCSI ID.

     On Linux platforms, if the SCSI IDs for each device are not set sequentially, and there is a gap in the sequence, the system only sees the first device.
  5. Turn off the system before you attach a device to prevent damage to the hardware.
  6. Attach a terminator to the last device in the chain of devices that are connected on one SCSI adapter card.

© Copyright IBM Corp. 1993, 2013

Results

For more information about selecting a device driver, see "Selecting a device driver" on page 83.

Attaching an automated library device to your system

You can attach and automated library device to your system to store your data on tapes.

About this task

Before you attach an automated library device, consider the following restrictions:

- Attached devices must be on their own Host Bus Adapter (HBA).
- An HBA must not be shared with other devices types (disk, CDROM, and so on).
- For multiport Fibre Channel HBAs, attached devices must be on their own port. These ports must not be shared with other device types.
- IBM tape drives have some special requirements on HBA and associated drivers. For more information about devices, see the Tivoli Storage Manager Supported Devices website.

Procedure

- To use the Fibre Channel (FC) adapter card, complete the following steps:
  1. Install the FC adapter card and associated drivers.
  2. Install the appropriate device drivers for attached medium changer devices.

     For more information about selecting a device driver, see "Selecting a device driver" on page 83.
- To use the SCSI adapter card, complete the following steps:
  1. Install the SCSI adapter card and associated drivers.
  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.
  3. Set the SCSI ID for the drives to the unused SCSI IDs.
  4. Set switches on the back of the device or set the IDs on the operator's panel. For each device that is connected in a chain to a single SCSI bus, you must configure it to have a unique SCSI ID. If each device does not have a unique SCSI ID, serious system problems can arise.
  5. Turn off your system before you attach a device to prevent damage to the hardware.
  6. Attach the device to your server system hardware, by following the manufacturer's instructions.
  7. Attach a terminator to the last device in the chain of devices that are connected on one SCSI adapter card.

Setting the library mode

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

About this task

The appropriate mode is usually called *random* mode; however, terminology can 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 does 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. A library that is configured in *sequential* mode is not detected by the system device driver as a library changer device.

Selecting a device driver

To use devices with Tivoli Storage Manager you must install the appropriate device driver.

IBM device drivers

IBM device drivers are available for most IBM labeled devices.

You can download IBM device drivers from the Fix Central website:

1. Go to the Fix Central website: http://www.ibm.com/support/fixcentral/.
2. Select **System Storage** for the Product Group menu.
3. Select **Tape systems** for the System Storage menu.
4. Select **Tape drivers and software** for the Tape systems menu.
5. Select **Tape device drivers** for the Tape drivers and software menu.
6. Select your operating system for the Platform menu.

For the most up-to-date list of devices and operating-system levels that are supported by IBM device drivers, see the Tivoli Storage Manager Supported Devices website at: http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html.

Tivoli Storage Manager device drivers

Tivoli Storage Manager provides device drivers or you can use native device drivers for your operating system to work with non IBM devices.

The Tivoli Storage Manager device driver is installed with the server.

The Tivoli Storage Manager device driver uses persistent reservation for some tape drives. See technote 1470319 at http://www.ibm.com/support/docview.wss?uid=swg21470319 for details.

- For the following tape devices, you can choose whether to install the Tivoli Storage Manager device driver or the native device driver for your operating system.
  All DLT and SDLT (including IBM 7337)
  4MM
  8MM
  DLT
  StorageTek SD3, 9490, 9840, 9940, and T10000
  Non IBM LTO
- All SCSI-attached libraries that contain tape drives from the list must use the Tivoli Storage Manager changer driver.

Device drivers that are acquired from other vendors can be used if they are supplied by the hardware vendor and are associated with the GENERICTAPE device class. Using a device class other than GENERICTAPE with a device driver acquired from another vendor is not recommended. Generic device drivers are not supported in write-one read-many (WORM) device classes.

For more information, see the **DEFINE DEVCLASS** - GENERICTAPE command in the *Administrator's Reference*.

Device special file names

A special file name for a device is required for the Tivoli Storage Manager server to work with tape, medium changer, or removable media devices.

During device configuration, one or more device special file names are created. For an IBM 349X, the device name is the symbolic name that you entered in the configuration file for the library. The special file is a symbolic link into the `/devices` tree of the file system. The symbolic link destination has the form:

`/devices/.../mt@x,0:mt`

where *x* is the SCSI ID of the device.

The following Table 8 table shows the form of the names. In the examples, *x* denotes a positive integer, 0 or greater.

Table 8. Device examples

Description	Device example	Logical file name
Tape drives that can be used by the Tivoli Storage Manager device driver	`/dev/rmt/xmt`	*x*`mt`
SCSI-attached libraries that can be used by the Tivoli Storage Manager device driver	`/dev/rmt/xlb`	*x*`lb`
IBM 3581, 3583, 3584, and 3590 tape drives, which can be used by the IBMtape device driver	`/dev/rmt/xst`	*x*`st`
IBM 3590 B11 with Automatic Cartridge Facility (ACF) feature, which can be used by the IBMtape device driver	`/dev/rmt/xsmc`	*x*`smc`
Mount point to use on REMOVABLEFILE device type (CD-ROM)	`/dev/cdx`	`cd`*x*

Correcting device special file names with SANDISCOVERY

The SANDISCOVERY server option is available to automatically correct a special file name for a device if it has changed due to changes in a SAN environment.

To allow both root and non-root users to perform SAN discovery, a special utility module, dsmqsan, is invoked when a SAN-discovery function is launched. The module performs as root, giving SAN-discovery authority to non-root users. While SAN discovery is in progress, dsmqsan runs as root.

The dsmqsan module is installed by default when the Tivoli Storage Manager server is installed. It is installed with owner root, group system, and mode 4755. The value of the SETUID bit is on. If, for security reasons, you do not want non-root users to run SAN-discovery functions, set the bit to off. If non-root users are having problems running SAN-discovery functions, check the following:

- The SETUID bit. It must be set to on.
- Device special file permissions and ownership. Non-root users need read/write access to device special files (for example, to tape and library devices).
- The SANDISCOVERY option in the server options file. This option must be set to ON.
- Increase the value of the SANDISCOVERYTIMEOUT option.

The dsmqsan module works only for SAN-discovery functions, and does not provide root privileges for other Tivoli Storage Manager functions.

Tivoli Storage Manager server device utilities

You can use device utilities for tasks related to configuring storage devices for the Tivoli Storage Manager server.

autoconf (Auto configure devices)

Use the **autoconf** utility to configure devices for use with the Tivoli Storage Manager server.

The **autoconf** utility performs the following tasks:

- Loads the driver to the kernel
- Creates the necessary files for the Tivoli Storage Manager device driver
- Creates device information files for libraries and tape devices

The **autoconf** utility is included in the device driver package and is installed to the /opt/tivoli/tsm/devices/bin directory.

Options

/t Enables tracing for the **autoconf** utility.

/? Displays information about the **autoconf** utility and its parameters.

Example: Configure devices by using the autoconf utility

Run **autoconf** to configure Tivoli Storage Manager devices:

```
> /opt/tivoli/tsm/devices/bin/autoconf
```

Related tasks:

"Reinstalling or upgrading the Tivoli Storage Manager device driver" on page 91

rmstdev (Detect and delete device special files)

Use the **rmstdev** utility to detect and delete device special files that are created by the Oracle Solaris st device driver. If these files exist on the system and you are using the Tivoli Storage Manager or IBM® tape device driver, data can be overwritten.

Run the rmstdev utility as the root user after each system reboot unless the server is started automatically by using the rc.dsmserv or dsmsta.rc utility. If you are not a root user, the rmstdev utility operates only in preview mode and you cannot issue `rmstdev -d` to delete files.

The rmstdev utility is included in the Tivoli Storage Manager server package and is installed to the `/opt/tivoli/tsm/server/bin` directory.

Options

/d Deletes st device special files that correspond to devices that are configured by the Tivoli Storage Manager device driver or the IBM® tape device driver.

/t Enables tracing for **rmstdev**.

/? Displays information about **rmstdev** and its parameters.

Example: Preview a list of st device special files

Run **rmstdev** to list all st device special files corresponding to Tivoli Storage Manager devices that are detected on the system:

```
> /opt/tivoli/tsm/devices/bin/rmstdev
```

Related tasks:

"Preventing tape labels from being overwritten" on page 94

Displaying information about devices with tsmdlist

Use the **tsmdlist** utility to view device names and other information about medium changer, and tape devices that are controlled by the Tivoli Storage Manager device driver.

After devices are configured, you can run the **tsmdlist** utility to display device information. The utility saves this information in output files that you can retrieve. The output files are named lbinfo for medium changer devices, and mtinfo for tape devices. After a device is added or reconfigured, you can update these output files by running the **tsmdlist** utility again. If you configure devices by using the **autoconf** utility, **tsmdlist** runs automatically after the devices are configured.

The `tsmdlist` utility and the output files it generates are in the devices bin directory, which is `/opt/tivoli/tsm/devices/bin` by default. Before you run the `tsmdlist` utility, make sure that either the Tivoli Storage Manager server is stopped or that all device activities are stopped. If a device is in use by the Tivoli Storage Manager server when the `tsmdlist` utility runs, a device busy error is issued.

Options

/t Displays trace messages for the **tsmdlist** utility.

/? Displays usage information about **tsmdlist** and its parameters.

Example: Display information about all devices

Display information about all devices that were configured by the Tivoli Storage Manager device driver:

```
tsmdlist
```

Installing and configuring device drivers

In order to use devices with Tivoli Storage Manager, you must install the correct device driver.

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

Install an IBM tape device driver to use an IBM SCSI tape device.

About this task

See the following documentation for instructions about installing and configuring IBM tape device drivers:

- *IBM Tape Device Drivers Installation and User's Guide*: http://www.ibm.com/support/docview.wss?uid=ssg1S7002972
- *IBM Tape Device Drivers Programming Reference*: http://www.ibm.com/support/docview.wss?uid=ssg1S7003032

After you complete the installation procedure in the *IBM Tape Device Drivers Installation and User's Guide*, the IBM device driver creates a set of special file names for each device. To see the special file names, issue the following command:

```
ls -la /dev/rmt
```

The output shows that the special files are symbolic links into the `/devices` tree of the file system. These files have the form:

```
/dev/rmt/xst
/dev/rmt/xsmc
```

Where `x` is a number. The file name might have more characters (n, b, c) at the end to indicate different operating characteristics, but these characters 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/rmt/xst`) or a library (`/dev/rmt/xsmc`).

Note: This applies to the IBM device driver only and the device type of this class must NOT be **GENERICTAPE**.

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 multiport FC Host Bus Adapter, or to different single 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.

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.

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*.

Multipath I/O is enabled automatically for medium changers, but not for tape drives. To enable multipath I/O for all configured tape drives on the system, follow these steps:

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.

When multipath I/O is enabled for a logical device, it remains enabled until the device is deleted or the support is unconfigured.

To obtain the names of special files, use the ***ls -l*** command. As shown in the following example, primary paths and alternate paths are identified by "Primary" and "Alt_path_*x*," where *x* is the number of the alternate path. (The output has been reformatted for readability.)

```
Inst#  Special File  Device   ...  Device Physical Path                          Path Type
686  /dev/rmt/12smc 03584L32  ... /devices/pci@If,2000/QLOGC,qla@1/IBMtape@1,1  Primary
688  /dev/rmt/14smc 03584L32  ... /devices/pci@If,2000/QLOGC,qla@1/IBMtape@2,1  Alt_path_1
685  /dev/rmt/0st   03592J1A  ... /devices/pci@1f,2000/QLOGC,qla@1/IBMtape@1,0  Primary
697  /dev/rmt/5st   03592J1A  ... /devices/pci@1f,2000/QLOGC,qla@1/IBMtape@7,0  Alt_path_1
```

The name of the special file associated with the primary path of this medium changer is /dev/rmt/12smc. Specify /dev/rmt/12smc as the value of the **DEVICE** parameter in the **DEFINE PATH** command for this medium changer. The name of the special file associated with the primary path for this tape drive is /dev/rmt/0st. Specify /dev/rmt/0st as the value of the **DEVICE** parameter in the **DEFINE PATH** command for this drive.

To display path-related details about a specific device on the system, enter the **tapelist** command, and then select option **54 Query Device Paths**.

Installing device drivers for IBM 349x libraries

Install the IBM device driver for an IBM 349X tape library.

About this task

See the following documentation for instructions about installing and configuring IBM tape device drivers:

- *IBM Tape Device Drivers Installation and User's Guide*: http://www.ibm.com/support/docview.wss?uid=ssg1S7002972
- *IBM Tape Device Drivers Programming Reference*: http://www.ibm.com/support/docview.wss?uid=ssg1S7003032

For an IBM 349X Tape Library Dataserver, install the IBM Tape Library Driver, which consists of a daemon and other components. To determine the device name that is needed when defining a path for the library, check what you entered in the /etc/ibmatl.conf file for the library's symbolic name.

Configuring Tivoli Storage Manager device drivers for vendor acquired SCSI-attached devices

You can configure Tivoli Storage Manager device drivers in two different ways. If you must selectively configure attached devices, you can run individual commands for tape drives and libraries. Or, you can use the auto configuration command to configure all devices that are attached to your system.

About this task

If you are configuring a tape library, you must set up device drivers for both the drives in the library and the library itself.

Restriction: Tivoli Storage Manager /mt device drivers do not support the Oracle Solaris mt command.

To install the Tivoli Storage Manager device drivers for tape drives and libraries, complete the following steps:

1. Ensure that the devices are connected to the system and powered on and active.

2. Ensure that the device driver package (TIVsmSdev) is installed.
3. Edit the following files:
   - For tape drives: `/usr/kernel/drv/mt.conf`
   - For libraries: `/usr/kernel/drv/lb.conf`

   Add one stanza (as shown in the example at the top of the file) to the device configuration file for each device's SCSI target ID LUN combination or WWN of the port to which the device is attached. This allows the device driver to probe the system that you want to use for supported devices. For more information about editing configuration files, see "Editing configuration files" on page 92.
4. Log in as root user.

   To add all devices, enter the following command:

   ```
   > /opt/tivoli/tsm/devices/bin/autoconf
   ```

   Or to add a single device type, use the following commands.
   - To add a tape drive, enter the following command:

     ```
     > /usr/sbin/add_drv -v -m '* 0666 bin bin' mt
     ```
   - To add a library, enter the following command:

     ```
     > /usr/sbin/add_drv -v -m '* 0666 bin bin' lb
     ```
5. Determine the special file names:
   - For tape drives, the device special files are saved in the `/dev/rmt` directory. Their names have the form `/dev/rmt/xmt`, where x is a number. To determine the relationship between a physical device and the special file that was created for it, issue the following command:

     ```
     > ls -l /dev/rmt/*mt
     ```

     The output shows that the special file is really a symbolic link into the `/devices` tree of the file system. The symbolic link destination has the form`/devices/.../mt@x,0:mt,` where x is the SCSI ID of the device. You can use this information to know which of the `/dev/rmt/xmt` special file names to provide to the server when you issue a **DEFINE PATH** command.
   - For tape libraries, the device special files are displayed in the `/dev/rmt` directory. Their names have the form `/dev/rmt/xlb`, where x is a number. To determine the relationship between a physical device and the special file that was created for it, issue the following command:

     ```
     > ls -l /dev/rmt/*lb
     ```

     The output shows that the special file is really a symbolic link into the `/devices` tree of the file system. The symbolic link destination has the form `/devices/.../lb@x,y:lb`, where x is the SCSI ID of the device and where y is the LUN (logical unit number) of the device. You can use this information to know which of the `/dev/rmt/xlb` special file names to provide to the server when you issue a **DEFINE PATH** command.

Reinstalling or upgrading the Tivoli Storage Manager device driver

When the Tivoli Storage Manager device driver is reinstalled, upgraded, or reloaded, or when changes are made to add a device, it is possible for the drive or library device name to change.

About this task

For example, a DLT drive with the device name */dev/rmt/3mt* may change to */dev/rmt/5mt*. Tivoli Storage Manager SAN Device Mapping will automatically update device special file names after the server is restarted if the feature is supported and the device has been defined to the server. Since the SAN Device Mapping is only supported by certain operating systems and HBAs, check the Tivoli Storage Manager Supported HBAs for San Device Mapping Technote at http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html before reinstalling or upgrading the Tivoli Storage Manager device driver. If the support is not available for your system, the device special file name will have to be updated manually. Complete the following steps to manually update device special file names.

1. Save the old device name. If the device is configured through autoconf, the device information has already been saved in the device information files--mtinfo, lbinfo, and opinfo. If the Tivoli Storage Manager device driver is loaded manually through the *add_drv* command, ensure that all devices controlled by the Tivoli Storage Manager device driver are not being used. Run TSMDLIST in the */opt/tivoli/tsm/devices/bin* directory to collect device information into mtinfo, lbinfo, and opinfo. Rename the files using the following commands:

   ```
   mv /opt/tivoli/tsm/devices/bin/mtinfo /opt/tivoli/tsm/devices/bin/
   mtinfo.org
   mv /opt/tivoli/tsm/devices/bin/lbinfo /opt/tivoli/tsm/devices/bin/
   lbinfo.org
   mv /opt/tivoli/tsm/devices/bin/opinfo /opt/tivoli/tsm/devices/bin/
   opinfo.org
   ```

2. Once the Tivoli Storage Manager device driver has been reinstalled or upgraded, or the device configuration file has been updated, reconfigure the device through autoconf using the following command:

   ```
   /opt/tivoli/tsm/devices/bin/autoconf
   ```

3. Compare the device names in the old device information files with those in the new device information files.

   If the new device name in the Special File column is different from the previous name, you will need to update the device path and provide the new device name for the DEVICE parameter. The following example shows the device name for the HP Ultrium 3 drive has been changed from */dev/rmt/0mt* to */dev/rmt/1mt* after a new device, Sony SDX-550V, was added to the tape configuration file:

```
#cat /opt/tivoli/tsm/devices/bin/mtinfo.org
Inst Special File Vendor Product        Serial     Tar- LUN  World Wide Name  Firm
                                                   get                        ware
---- ------------ ------ -------------- ---------- ---  ---- ---------------- ----
1    /dev/rmt/0mt HP     Ultrium 3-SCSI HU1044827C 1    0    50060b00002c203e G54W

#cat /opt/tivoli/tsm/devices/bin/mtinfo
Inst Special File Vendor Product        Serial     Tar- LUN  World Wide Name  Firm
                                                   get                        ware
---- ------------ ------ -------------- ---------- ---  ---- ---------------- ----
4    /dev/rmt/0mt SONY   SDX-550V       0001000052 4    0    08004606c00f4274 0100
1    /dev/rmt/1mt HP     Ultrium 3-SCSI HU1044827C 1    0    50060b00002c203e G54W
```

4. Start the Tivoli Storage Manager server, and update the path for the device. If library device names are not updated, library initialization will fail. Drive will be taken offline if their device paths are not updated. Use the following commands to update the paths.

   For a drive:

   UPDate PATH *source_name destination_name* SRCType=SERVer DESTType=DRive LIBRary=*library_name* **DEVIce=new_drive_name**

   For a library:

   UPDate PATH *source_name destination_name* SRCType=SERVer DESTType=LIBRary **DEVIce=new_library_name**

Editing configuration files

For SCSI-attached devices, the configuration files must contain one stanza for each SCSI-attached device that the Tivoli Storage Manager server is to use. You can edit the `mt.conf` file for tape drives and the `lb.conf` file for library devices.

About this task

- For SCSI-attached tape drives, add the following line to the `mt.conf` file:

  ```
  name="mt" class="scsi" target=$X$ lun=$Y$;
  ```

- For SCSI-attached libraries, add the following line to the `lb.conf` file:

  ```
  name="lb" class="scsi" target=$X$ lun=$Y$;
  ```

Replace the X string with the SCSI ID of the tape or library drive. This ID is a number in the range of 0-6 (7 is reserved for the SCSI adapter and cannot be used for devices) and possibly 8-15, depending on the type of the SCSI adapter. In a configuration where devices are attached to multiple initiators with the same SCSI ID, only one target or logical unit number (LUN) ID must be specified. The device driver attaches to all targets and LUNs that are the same if they are on different adapters. Replace the Y string with the SCSI LUN of the device. This number is usually zero. All other fields must be left as they are shown in the preceding example.

For example, if the SCSI ID number of your tape drive is six, and the LUN ID is zero, add the following line to the `mt.conf` file:

```
name="mt" class="scsi" target=6 lun=0;
```

For a library, add the following line to the `lb.conf` file:

```
name="lb" class="scsi" target=6 lun=0;
```

Note:

1. The **add_drv** command probes all devices as specified in the device .conf file and creates separate device special files for each entity.
2. An extra line is required in the lb.conf, mt.conf, and op.conf files so that pseudo devices with the special file names /devices/pseudo/mt@16383:tsmmtctl, /devices/pseudo/lb@16383:tsmlbctl, and /devices/pseudo/mt@16383:tsmopctl are created. If these pseudo devices are not created, the **ddtrace** command can fail to get device driver trace.

   Add the following line to the mt.conf file:

   ```
   name="mt" parent="pseudo" instance=16383;
   ```

   Add the following line to the lb.conf file:

   ```
   name="lb" parent="pseudo" instance=16383;
   ```

For Fibre Channel attached devices, the configuration file must contain one stanza for each Fibre Channel attached device that the Tivoli Storage Manager server is to use. You can edit the mt.conf file for tape drives and the lb.conf file for library devices.

For a tape drive, add the following line to the mt.conf file:

```
name="mt" parent="fp" lun=0 fc-port-wwn="xxx";
```

For a library, add the following line to the lb.conf file:

```
name="lb" parent="fp" lun=0 fc-port-wwn="yyy";
```

Where xxx, and yyy are the 16-digit numbers that represent the Fibre Channel port worldwide name for the devices.

Note:

1. The **add_drv** command probes all devices as specified in the .conf file and creates separate device special files for each entity.
2. An extra line is required in the lb.conf, mt.conf, and op.conf files so that pseudo devices with the special file names /devices/pseudo/mt@16383:tsmmtct, /devices/pseudo/lb@16383:tsmlbctl, and /devices/pseudo/lb@16383:tsmopctl are created. If these pseudo devices are not created, the **ddtrace** command can fail to get device driver trace.

   Add the following line to the mt.conf file:

   ```
   name="mt" parent="pseudo" instance=16383;
   ```

   Add the following line to the lb.conf file:

   ```
   name="lb" parent="pseudo" instance=16383;
   ```

   Add the following line to the op.conf file:

   ```
   name="op" parent="pseudo" instance=16383;
   ```
3. The /devices/pseudo/mt@1638:tsmmtctl and /devices/pseudo/lb@1638:tsmlbctl must be enabled in the mt.conf and lb.conf files. Otherwise, the SAN device mapping is not able to determine the special file names for Tivoli Storage Manager devices.
4. For Solaris 10 update 10 or higher releases, more changes are required. Check the lb.conf, mt.conf, and op.conf files for detailed information.

Preventing tape labels from being overwritten

Tivoli Storage Manager tape labels can be overwritten by an application that uses a generic or native device driver. If tape labels are overwritten, data can be lost.

About this task

The Tivoli Storage Manager device driver and the Oracle Solaris generic SCSI tape driver (`st` driver) can coexist in the kernel at the same time. They can also control and operate the same tape device that is attached on the system, which can cause conflicts over how the device is managed.

Note: For Solaris 10 update 10 or higher releases, the SCSI tape driver cannot coexist with the Tivoli Storage Manager driver.

If the Oracle Solaris generic SCSI tape driver (`st` driver) controls devices that are also used by the Tivoli Storage Manager device driver or the IBMtape device driver, Tivoli Storage Manager internal tape labels can be overwritten and data can be lost. By default, the `st` driver rewinds tapes at the end of an operation unless the non-rewind option is specified in the device special file name.

The auto rewind operation relocates the tape header position to the beginning of the tape. If the tape remains loaded in the drive, the next non-Tivoli Storage Manager write operation overwrites the Tivoli Storage Manager tape label because the label is at the beginning of the tape.

To prevent Tivoli Storage Manager data from being overwritten, the `rmstdev` utility locates and deletes all device special files that are created by the `st` driver and that correspond to devices configured by either the Tivoli Storage Manager device driver or the IBMtape device driver. This utility runs at startup when the server or storage agent is started automatically. If you do not start the Tivoli Storage Manager server or storage agent automatically, run `rmstdev -d` as the root user before you start the server or storage agent.

If `rmstdev` does not run when the server is started and `st` device special files are detected for any Tivoli Storage Manager tape drives, the drives are put into the polling state and the following error message is issued every 10 minutes:

```
The server has detected that <tape drive> can also be accessed by other applications
through the st device special file <device special file name>. You must run the
rmstdev utility with -d as the root user to delete the st device special file.
```

After one hour, the drive is taken offline if the problem still exists.

Error messages are also displayed when the **DEFINE PATH** command is issued and Tivoli Storage Manager detects `st` device special files. The operation fails and the drive is not defined. To prevent any conflict that can occur, log on as the root user and run `/opt/tivoli/tsm/server/bin/rmstdev -d` to delete the `st` special files. You do not have to halt the server to run the `rmstdev` utility. You can then redefine a path to the drive.

Related reference:

"rmstdev (Detect and delete device special files)" on page 86

Installing the Tivoli Storage Manager device driver to Solaris Zones

The Tivoli Storage Manager device driver supports the Solaris Zone feature, available with Solaris Version 10.

About this task

You can install the Tivoli Storage Manager device driver in the global zone or any non-global zone. When you install from the global zone, the full device driver package is installed to the global zone. When you install from a non-global zone, only device driver tools are installed to non-global zones on the system. The operation is the same in both types of zones. The Tivoli Storage Manager device driver level in the global zone must match the highest level of the server running on the same host system in any zone. Only one version of Tivoli Storage Manager device driver is supported in the global zone at a time. Versions of the device driver tools running in non-global zones must be at the same level or lower than the level of the device driver installed in the global zone.

If you are installing the device driver package to a non-global zone, devices must be mapped by a root user from the global zone to the non-global zone before they can be used. Devices installed to the global zone are available for immediate use.

If you are using the Tivoli Storage Manager device driver or the IBMtape device driver to control tape drives in the global zone, you can run the rmstdev utility to detect and delete device special files created by the Oracle Solaris st device driver. If these files are not deleted, conflicts can occur if more than one device driver tries to access a drive and data can be overwritten.

The Tivoli Storage Manager device driver package includes the device mapping commands devadd, devlist, and devrem for Solaris Version 10. These commands allow you to add, list, or remove devices for specific Solaris Zones.

Adding devices to non-global zones

Use the devadd command to map devices from the global zone to non-global zones.

About this task

The devadd command can only be used in the global zone. When you issue the command, prompts with menus ask you to select a non-global zone, a mapping option, and, for a single device, the device special file name.

The devadd command will map all targeted devices or a single targeted device from the global zone to the targeted non-global zone. After mapping is complete, devices in the targeted non-global zone are ready for immediate use. Special file names for mapped devices in the targeted non-global zone are the same as in the global zone. You do not need to reboot the targeted non-global zone to use the devices mapped to it.

Adding all tape drives in the global zone to a non-global zone

To add all tape drives in the global zone to a non-global zone, complete the following steps.

Procedure

1. Issue the devadd command. A list of zones is displayed:

```
(root@sunflame) /opt/tivoli/tsm/devices/bin>devadd

ID NAME            STATUS         PATH
0 global           running        /
1 sunflame1        running        /zones/sunflame1
2 sunflame2        running        /zones/sunflame2
```

2. Enter a local zone name for a device to be mapped, for example, sunflame2. A list of device mapping options is displayed:

```
1) one tape drive            2) all tape drives
3) one medium changer        4) all medium changers
5) one optical disk drive    6) all optical disk drives
7) all types of devices      8) quit this program
```

3. Enter a device mapping option, for example, 2.

Results

All tape drives have been mapped to the non-global zone sunflame2.

Adding a single tape drive from the global zone to a non-global zone

To add a single tape drive from the global zone to a non-global zone, complete the following steps.

Procedure

1. Issue the devadd command. A list of zones is displayed:

```
(root@sunflame) /opt/tivoli/tsm/devices/bin>devadd

ID NAME            STATUS         PATH
0 global           running        /
1 sunflame1        running        /zones/sunflame1
2 sunflame2        running        /zones/sunflame2
```

2. Enter a local zone name for a device to be mapped, for example, 1. A list of device mapping options is displayed:

```
1) one tape drive            2) all tape drives
3) one medium changer        4) all medium changers
5) one optical disk drive    6) all optical disk drives
7) all types of devices      8) quit this program
```

3. Enter a device mapping option, for example, 1. A list of available Tivoli Storage Manager devices in the global zone is displayed:

```
Major: 203 Minor: 28 File: /dev/rmt/43mt
Major: 203 Minor: 30 File: /dev/rmt/44mt
Major: 203 Minor: 31 File: /dev/rmt/45mt
```

4. Enter the special file name of a mapping device:

```
/dev/rmt/45mt
```

The device /dev/rmt/45mt has been mapped to the sunflame1 non-global zone.

Displaying devices configured in zones

Use the devlist command to list Tivoli Storage Manager devices in different zones.

About this task

The devlist command can be used in any zone. When devlist is issued from the global zone, all devices configured in the global zone and all devices mapped to each non-global zone are displayed. If devlist is issued in a non-global zone, only Tivoli Storage Manager devices mapped to the non-global zone will be displayed.

Displaying all devices in all zones

To display all devices configured in the global and non-global zones, issue the devlist command from the global zone.

About this task

```
(root@sunflame) /opt/tivoli/tsm/devices/bin>devlist

======== The global zone ========

TSM medium changer devices: None.

TSM tape drive devices:
/dev/rmt/43mt
/dev/rmt/44mt
/dev/rmt/45mt

TSM optical disk drive devices: None.

====== The sunflame1 local zone ========

TSM medium change devices: None.

TSM tape drive devices:
/zones/sunflame1/dev/rmt/43mt
/zones/sunflame1/dev/rmt/44mt
/zones/sunflame1/dev/rmt/45mt

TSM optical disk drive devices: None.

====== The sunflame2 local zone ========

TSM medium change devices: None.

TSM tape drive devices: None.

TSM optical disk drive devices: None.
```

Displaying devices in a non-global zone

To display all devices configured in a specific non-global zone, issue the devlist command from the non-global zone:

About this task

```
(root@sunflame1) /opt/tivoli/tsm/devices/bin>devlist

======== The sunflame1 local zone ========

TSM medium change devices: None

TSM tape drive devices:
/dev/rmt/43mt
/dev/rmt/44mt
/dev/rmt/45mt

TSM optical disk drive devices: None
```

Removing devices from non-global zones

Use the devrem command to remove devices from non-global zones.

About this task

The devrem command can only be used in a non-global zone. When you issue the command, prompts with menus ask you to select a non-global zone and device removal option. The devrem command removes all targeted devices or a single device from the targeted non-global zone. When removal is complete, the deleted devices are no longer available, and all special files associated with them are removed from the non-global zone.

Removing a single device from a non-global zone

To remove a single tape drive device from a non-global zone, complete the following steps.

Procedure

1. Issue the devrem command. A list of zones is displayed:

```
(root@sunflame) /opt/tivoli/tsm/devices/bin>devrem

   ID NAME            STATUS         PATH
   0 global           running        /
   1 sunflame1        running        /zones/sunflame1
   2 sunflame2        running        /zones/sunflame2
```

2. Enter a local zone name for a device to be removed, for example, sunflame1. A list of device removal options is displayed:

```
   1) one tape drive             2) all tape drives
   3) one medium changer         4) all medium changers
   5) one optical disk drive     6) all optical disk drives
   7) all types of devices       8) quit this program
```

3. Enter a device removal option, for example, 1. A list of Tivoli Storage Manager tape drives is displayed:

```
   TSM tape drive devices in this sunflame1 local zone are:
   crwxr-xr-x   1 root     root     203, 28 Jul 22 15:53 /dev/rmt/43mt
   crwxr-xr-x   1 root     root     203, 30 Jul 22 15:53 /dev/rmt/44mt
   crwxr-xr-x   1 root     root     203, 31 Jul 22 15:53 /dev/rmt/45mt
```

4. Enter the special file name for the tape drive you want to remove, for example:
   `/dev/rmt/45mt`

The tape drive /dev/rmt/45mt has been removed from the sunflame1 local zone.

Removing all devices from a non-global zone

To remove all devices from a non-global zone, complete the following steps.

Procedure

1. Issue the devrem command. A list of zones is displayed:

```
(root@sunflame) /opt/tivoli/tsm/devices/bin>devrem
   ID NAME            STATUS         PATH
   0 global           running        /
   1 sunflame1        running        /zones/sunflame1
   2 sunflame2        running        /zones/sunflame2
```

2. Enter the local zone name where you want to remove all devices, for example, sunflame1. A list of device removal options is displayed:

```
   1) one tape drive            2) all tape drives
   3) one medium changer        4) all medium changers
   5) one optical disk drive    6) all optical disk drives
   7) all types of devices      8) quit this program
```

3. Enter 7, for all device types.

Results

All Tivoli Storage Manager devices have been removed from the sunflame1 local zone.

Installing the Centera SDK for Centera shared libraries

Beginning with Tivoli Storage Manager Version 5.5, Centera shared libraries are not installed with the server. In order to use Centera with Tivoli Storage Manager, the Centera SDK must be installed. You can download the Centera SDK libraries from the EMC website.

About this task

Perform the following steps when you set up the Tivoli Storage Manager server to access Centera:

Procedure

1. Install the Tivoli Storage Manager server.
2. If you are upgrading from a previous level of Tivoli Storage Manager, delete the Centera SDK libraries from the directory where the server was installed. For each platform delete the following files:

Table 9. Centera SDK library files to delete

Operating system	Files to delete
AIX and Linux x86_64	In the `/usr/local/Centera_SDK/lib/64` directory: libFPCore64.a libFPLibrary64.a libFPParser64.a libFPStreams64.a libFPUtils64.a libFPXML64.a libPAI_module64.a
HP-UX and Oracle Solaris	In the `/opt/Centera_SDK/lib/64` directory: libFPCore64.a libFPLibrary64.a libFPParser64.a libFPStreams64.a libFPUtils64.a libFPXML64.a libPAI_module64.a

3. Contact your EMC representative to obtain the installation packages and instructions to install the Centera SDK Version 3.2 or later.
4. Install the Centera SDK. During the installation, take note of the directory where the Centera SDK is installed.
   a. Unzip and untar the package in a working directory.
   b. Inside the installation directory is an installation script. It copies the libraries to a default directory:

      AIX and Linux x86_64: `/usr/local/Centera_SDK/lib/64`

      HP-UX and Oracle Solaris: `/opt/Centera_SDK/lib/64`
5. After the SDK is installed, set the following environment variables to the directory where the SDK was installed. This is necessary to allow Tivoli Storage Manager to locate the SDK.

   AIX: LIBPATH

   HP-UX: SHLIB_PATH

   Linux x86_64: LD_LIBRARY_PATH

   Oracle Solaris (64–bit only): LD_LIBRARY_PATH_64
6. Check the permissions of each library file. Ensure that each library file has read permissions to the Tivoli Storage Manager instance user ID.
7. Start the Tivoli Storage Manager server and set up the policy, device class, and storage pools for Centera.

Chapter 6. Configuring storage devices

Before you can use a removable media device with Tivoli Storage Manager, you must plan for, attach, and configure the device by defining the objects that represent the device to the Tivoli Storage Manager server.

About this task

Device configuration requires an understanding of Tivoli Storage Manager storage objects. For an introduction to storage objects, see "Tivoli Storage Manager storage objects" on page 40.

Procedure

1. Plan for the device by reviewing your storage requirements and hardware environment.

   For planning information, 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, see Chapter 5, "Attaching devices for the server," on page 81.
3. Define libraries, drives, paths, device classes, storage pools, and storage volume objects to Tivoli Storage Manager.

   For more information, see, "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.
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.
6. Register clients to the domain associated with the policy that you defined or updated in the preceding step. For more information, see Chapter 13, "Implementing policies for client data," on page 469.

What to do next

After you attach and define 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 516.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Storage pool hierarchies" on page 256.

© Copyright IBM Corp. 1993, 2013

Configuring libraries for use by one server

To use a library or libraries for storage from one Tivoli Storage Manager server, you must first set up the device on the server system.

Procedure

1. For 3494 libraries, set the library ID or name.

   Set the symbolic name for the library in the configuration file for the library device driver (`/etc/ibmatl.conf`).

   These procedures are described in the *IBM Tape Device Drivers Installation and User's Guide* at http://www.ibm.com/support/docview.wss?uid=ssg1S7002972.
2. Physically attach devices to the server hardware.
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 library to Tivoli Storage Manager.
5. Configure the library:
   a. Define the library.
   b. Define a path from the server to the library.
   c. Define the drives in the library.

      Note:

      For SCSI and VTL libraries, you can use the **PERFORM LIBACTION** to define drives and paths for a library instead of completing Steps c and d separately.

      For ACSLS and 3494 libraries: 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, or two generations of drives of the same device type, define two libraries. Then define the drives and device classes for each library.
   d. Define a path from the server to each drive.
   e. Define a Tivoli Storage Manager device class.

      Device classes specify the recording formats for drives and classify them according to type. Use the default value, **FORMAT=DRIVE** as the recording format only if all the drives associated with the device class can read and write to all of the media. For example, if you have a mix of Ultrium Generation 3 and Ultrium Generation 4 drives, but only have Ultrium Generation 3 media, you can specify **FORMAT=DRIVE** because both the Generation 4 and Generation 3 drives can read from and write to Generation 3 media.
   f. Define a storage pool.

      Consider the following key choices for defining storage pools:

      - Scratch volumes are 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. Also, specify **MAXSCR=0** when you define the storage pool so that scratch volumes are not used.

- 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 that belong to a group of client nodes, a single client node, a client file space, or a group of client file spaces 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 353 and "How collocation affects reclamation" on page 374.

g. Check in and label 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 that you might have later.

   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. You can use the **CHECKIN LIBVOLUME** command to check in volumes that are already labeled. Or, if you want to label and check in volumes with one step, issue the **LABEL LIBVOLUME** command.

   Note: If your library has drives of multiple device types, and you defined two libraries to the Tivoli Storage Manager server, the two libraries represent one physical library. You must check in media separately to each defined library. Ensure that you check in volumes to the correct Tivoli Storage Manager library.

6. Verify your device definitions to ensure that everything is configured correctly. Use the **QUERY** command to review information about each storage object.

   When you issue **QUERY DRIVE**, verify that the device type for the drive is what you are expecting. If a path is not defined, the drive device type is listed as UNKNOWN and if the wrong path is used, GENERIC_TAPE or another device type is shown. This step is especially important when you are using mixed media.

Example: Configure a SCSI or virtual tape library with a single drive device type

Configure a VTL or SCSI library that contains two LTO tape drives.

About this task

Assume that you want to attach an automated SCSI library that contains 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 this configuration, both drives in the library are the same device type. Define one device class. The procedure is the same for both SCSI and VTLs, except for the step to define the library. For SCSI libraries, define the library with **libtype=scsi**. For VTLs, define the library with **libtype=vtl**.

Procedure

1. Define a SCSI library named AUTODLTLIB.

   ```
   define library autoltolib libtype=scsi
   ```

   Note: If the library has a bar code reader and you would like to automatically label tapes before they are checked in, you can set the **AUTOLABEL** parameter to YES. For example:

   ```
   define library autoltolib libtype=scsi autolabel=yes
   ```

2. Define a path from the server to the library.

   ```
   define path server1 autoltolib srctype=server desttype=library
   device=/dev/rmt/3lb
   ```

3. Define the drives in the library. Both drives belong to the AUTODLTLIB library.

   ```
   define drive autoltolib drive01
   define drive autoltolib drive02
   ```

   Tip: You can use the **PERFORM LIBACTION** command to define drives and paths for a library in one step.

4. Define a path from the server to each drive.

   ```
   define path server1 drive01 srctype=server desttype=drive
   library=autoltolib device=/dev/rmt/4mt
   define path server1 drive02 srctype=server desttype=drive
   library=autoltolib device=/dev/rmt/5mt
   ```

   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.

5. Define a device class named AUTODLT_CLASS for the two drives in the AUTODLTLIB library.

   ```
   define devclass autolto_class library=autodltlib devtype=lto
   ```

6. Define a storage pool named AUTOLTO_POOL associated with the device class named AUTOLTO_CLASS.

   ```
   define stgpool autolto_pool autolto_class maxscratch=20
   ```

7. Label and check in library volumes.

   ```
   label libvolume autoltolib search=yes labelsource=barcode checkin=scratch
   ```

8. Verify your definitions by issuing the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   query stgpool
   query libvolume
   ```

Example: Configure a SCSI or virtual tape library 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.

About this task

Assume that you want to attach an automated SCSI library that contains 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 by SCSI cables.

In this configuration, the drives are different device types. 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.

The procedure is the same for both SCSI and VTLs, except for the step to define the library. For SCSI libraries, define the library with **libtype=scsi**. For VTLs, define the library with **libtype=vtl**.

Procedure

1. Define a SCSI library named MIXEDLIB.

   ```
   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/rmt/3lb
   ```

3. Define the drives in the library. Both drives belong to the MIXEDLIB library.

   ```
   define drive mixedlib dlt1
   define drive mixedlib lto1
   ```

4. Define a path from the server to each drive. The **DEVICE** parameter specifies the device driver's name for the drive, which is the device special file name.

   ```
   define path server1 dlt1 srctype=server desttype=drive
   library=mixedlib device=/dev/rmt/4mt
   define path server1 lto1 srctype=server desttype=drive
   library=mixedlib device=/dev/rmt/5mt
   ```

   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.

5. Define device classes.

   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
   ```

6. Define storage pools that are associated with the device classes.

   ```
   define stgpool lto_pool lto_class maxscratch=20
   define stgpool dlt_pool dlt_class maxscratch=20
   ```

7. Label and check in library volumes.

   ```
   label libvolume mixedlib search=yes labelsource=barcode checkin=scratch
   ```

8. Verify your definitions by issuing the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   query stgpool
   query libvolume
   ```

Example: Configure an ACSLS library with a single drive device type

The parameter ACSID specifies the number that the Automatic Cartridge System System Administrator (ACSSA) assigned to the library. Issue the QUERY ACS command to your ACSLS system to determine the number for your library ID.

About this task

In this example, both drives in the ACSLS library are the same device type.

Procedure

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.

3. Define a path from the server to each drive.

   ```
   define path server1 drive01 srctype=server desttype=drive
    library=acslib device=/dev/rmt/0mt

   define path server1 drive02 srctype=server desttype=drive
    library=acslib device=/dev/rmt/1mt
   ```

4. Define a device class named ACS_CLASS to classify the two drives in the ACSLIB library.

   ```
   define devclass acs_class library=acslib devtype=ecartridge
   ```

5. Define a storage pool named ACS_POOL and associate it with the device class ACS_CLASS to use the devices that are defined there.

   ```
   define stgpool acs_pool acs_class maxscratch=20
   ```

6. Label and check in library volumes.

   ```
   label libvolume acslib search=yes overwrite=no checkin=scratch
   ```

7. To check what you have defined, issue the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   query stgpool
   query libvolume
   ```

Example: Configure an ACSLS library with multiple drive device types

The following example shows how to set up an ACSLS library with a mix of two T10000C drives and two T10000A drives.

Procedure

1. Define two ACSLS libraries named T10000CLIB and T10000ALIB. Both libraries use the same ACSID.

   ```
   define library T10000Clib libtype=acsls acsid=1
   define library T10000Alib 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 T10000C drives to T10000CLIB.

     ```
     define drive T10000Clib T10000C_drive1 acsdrvid=1,2,3,1
     define drive T10000Clib T10000C_drive2 acsdrvid=1,2,3,2
     ```

   - Define the T10000A drives to T10000ALIB.

     ```
     define drive T10000Alib T10000A_drive3 acsdrvid=1,2,3,3
     define drive T10000Alib T10000A_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.

3. Define a path from the server to each drive. Ensure that you specify the correct library.

   - For the T10000C drives:

     ```
     define path server1 T10000C_drive1 srctype=server desttype=drive
      library=T10000Clib device=/dev/rmt/0mt

     define path server1 T10000C_drive2 srctype=server desttype=drive
      library=T10000Clib device=/dev/rmt/1mt
     ```

   - For the T10000A drives:

     ```
     define path server1 T10000A_drive3 srctype=server desttype=drive
      library=T10000Alib device=/dev/rmt/2mt

     define path server1 T10000A_drive4 srctype=server desttype=drive
      library=T10000Alib device=/dev/rmt/3mt
     ```

   The **DEVICE** parameter gives the device special file name for the drive.

4. Classify the drives in the two libraries according to type by defining one device class for each type of drive. Because there are separate libraries, you can enter a specific recording format or you can enter DRIVE.

```
define devclass T10000C_class library=T10000Clib devtype=ecartridge
 format=T10000C

define devclass T10000A_class library=T10000Alib devtype=ecartridge
 format=T10000A
```

5. Create the storage pools to use the devices in the device classes that you just defined. For example, define storage pools named T10000C_POOL associated with the device class T10000C_CLASS and T10000A_POOL associated with the device class T10000A_CLASS:

```
define stgpool T10000C_pool T10000C_class maxscratch=20

define stgpool T10000A_pool T10000A_class maxscratch=20
```

6. Label and check in library volumes.

```
label libvolume T10000Clib search=yes overwrite=no checkin=scratch
label libvolume T10000Alib search=yes overwrite=no checkin=scratch
```

7. To check what you have defined, enter the following commands:

```
query library
query drive
query path
query devclass
query stgpool
query libvolume
```

Example: Configure a 3494 library with a single drive device type

Configure a 3494 library that contains two IBM 3592 tape drives. Both drives in the library are the same device type.

Procedure

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).

3. Define the drives in the library.

```
define drive 3494lib drive01
define drive 3494lib drive02
```

Both drives belong to the 3494LIB library.

4. Define a path from the server to each drive.

```
define path server1 drive01 srctype=server desttype=drive
 library=3494lib device=/dev/rmt/0mt
define path server1 drive02 srctype=server desttype=drive
 library=3494lib device=/dev/rmt/1mt
```

5. Define a device class named 3494_CLASS to classify the two 3592 drives according to type in the 3494LIB library.

```
define devclass 3494_class library=3494lib devtype=3592
```

6. Define a storage pool named 3494_POOL that is associated with the device class named 3494_CLASS.

```
define stgpool 3494_pool 3494_class maxscratch=20
```

7. Label and check in library volumes.

```
label libvolume 3494lib search=yes checkin=scratch
```

8. Verify your definitions by issuing the following commands:

```
query library
query drive
query path
query devclass
query stgpool
query libvolume
```

Example: Configure a 3494 library with multiple drive device types

The following example shows how to set up a 3494 library that contains two IBM TS1140 tape drives and two IBM TS1120 tape drives. The drives are different device types.

Procedure

1. Define two libraries, one for each type of drive. For example, define a library named TS1140LIB1 and a library named TS1120LIB2. Specify scratch and private categories explicitly. If you accept the category defaults for both library definitions, different types of media are assigned to the same categories.

   ```
   define library TS1140lib1 libtype=349x scratchcategory=301 privatecategory=300
   define library TS1120lib2 libtype=349x scratchcategory=401 privatecategory=400
   ```

2. Define a path from the server to each library:

   ```
   define path server1 TS1140lib1 srctype=server desttype=library device=library1
   define path server1 TS1120lib2 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`).

3. Define the drives, ensuring that they are associated with the appropriate libraries.

   - Define the TS1140 drives to TS1140LIB1.

     ```
     define drive TS1140lib1 TS1140_drive1
     define drive TS1140lib1 TS1140_drive2
     ```

   - Define the TS1120 drives to TS1120LIB2.

     ```
     define drive TS1120lib2 TS1120_drive3
     define drive TS1120lib2 TS1120_drive4
     ```

   Note: Tivoli Storage Manager does not prevent you from associating a drive with the wrong library.

4. Define a path from the server to each drive. Ensure that you specify the correct library. The **DEVICE** parameter gives the device special file name for the drive.

   - For the TS1140 drives:

     ```
     define path server1 TS1140_drive1 srctype=server desttype=drive
      library=TS1140lib1 device=/dev/rmt/0mt
     define path server1 TS1140_drive2 srctype=server desttype=drive
      library=TS1140lib1 device=/dev/rmt/1mt
     ```

   - For the TS1120 drives:

     ```
     define path server1 TS1120_drive3 srctype=server desttype=drive
      library=TS1120lib2 device=/dev/rmt/2mt
     define path server1 TS1120_drive4 srctype=server desttype=drive
      library=TS1120lib2 device=/dev/rmt/3mt
     ```

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 or you can enter DRIVE.

```
define devclass TS1140_class library=TS1140lib1 devtype=3592 format=3592-4

define devclass TS1120_class library=TS1120lib2 devtype=3592 format=3592-2
```

6. Create the storage pools to use the devices in the device classes you just defined. For example, define a storage pool named TS1140POOL associated with the device class TS1140_CLASS, and TS1120POOL associated with the device class TS1120_CLASS:

```
define stgpool TS1140pool TS1140_class maxscratch=20

define stgpool TS1120pool TS1120_class maxscratch=20
```

7. Label and check in library volumes.
   - You can check in volumes that are already labeled:

     ```
     checkin libvolume TS1140lib1 search=yes status=scratch checklabel=no
     checkin libvolume TS1120lib2 search=yes status=scratch checklabel=no
     ```

   - Or, you can label and check in volumes at the same time:

     ```
     label libvolume TS1140lib1 search=yes checkin=scratch
     label libvolume TS1120lib2 search=yes checkin=scratch
     ```

8. To check what you have defined, enter the following commands:

```
query library
query drive
query path
query devclass
query stgpool
query libvolume
```

Configuring library sharing

Multiple Tivoli Storage Manager servers can share storage devices by using a SAN.

About this task

With LAN-free data movement, Tivoli Storage Manager client systems can directly access storage devices that are defined to a Tivoli Storage Manager server. Storage agents are installed and configured on the client systems to perform the data movement. See "Configuring IBM Tivoli Storage Manager for LAN-free data movement" on page 125

To set up library sharing, you must define one Tivoli Storage Manager server as the library manager for your shared library configuration. Then, you must define additional Tivoli Storage Manager servers as library clients that communicate and request storage resources from the library manager. The library manager server must be at the same version or a newer version as the server or servers that are defined as library clients.

The following tasks are required for Tivoli Storage Manager servers to share library resources on a SAN:

Procedure

1. Set up server-to-server communications.

   To share a storage device on a SAN, define servers to each other using the cross-define function. See "Setting up communications among servers" on page 682 for details.

2. Define a shared library and set up devices on the server systems.

   Use the procedure that is described in "Configuring libraries for use by one server" on page 102 to define a library for use in the shared environment.

Modify the procedure to define the library as shared, by specifying the **SHARED=YES** parameter for the **DEFINE LIBRARY** command.

3. Define the library manager server.
4. Define the shared library on the library client server.
5. From the library manager server, define paths from the library client to each drive that the library client can access. The device name must reflect the way that the library client system recognizes the device. A path from the library manager to each drive must be defined in order for the library client to use the drive.

   It is best practice for any library-sharing configuration to have all drive path definitions that are defined for the library manager also defined for each library client. For example, if the library manager defines three drives, the library client must also define three drives. If you want to limit the number of drives that a library client can use at a time, use the **MOUNTLIMIT** parameter of the device class on the library client instead of limiting the drive path definitions for the library client.
6. Define device classes for the shared library.

   A good practice is to make the device class names the same on both servers to avoid confusion when you define multiple device classes with the same device type and library parameters. Some operations, such as database backup, use the device class name to identify the data for backup.

   The device class parameters that are specified on the library manager override the parameters that are specified for the library client whether the device class names are the same on both servers or not. If the device class names are different, the library manager uses the parameters that are specified in a device class that matches the device type that is specified for the library client.
7. Define a storage pool for the shared library.
8. Repeat the steps to define a library client for additional library client servers in your configuration.

Example: Configure library sharing for SCSI or virtual tape libraries

Use the sample procedure to set up a VTL or SCSI library sharing environment.

About this task

In this example, a library manager server named ASTRO and a library client named JUDY are configured. To help clarify where each step is performed, the commands are preceded by the server name from which the command is issued. Most commands are issued from the library client.

The procedure is the same for both SCSI and VTLs, except for the step to define the library. For SCSI libraries, define the library with **libtype=scsi**. For VTLs, define the library with **libtype=vtl**.

Procedure

1. To set up ASTRO as the library manager server, define a shared SCSI library named SANGROUP. For example:

   ```
   astro> define library sangroup libtype=scsi shared=yes
   ```

Then complete the rest of the steps as described in "Example: Configure a SCSI or virtual tape library with a single drive device type" on page 103 to configure the library.

Tip: You can use the **PERFORM LIBACTION** command to define drives and paths for a library in one step.

2. Define ASTRO as the library manager server by issuing the **DEFINE SERVER** command.

```
judy> define server astro serverpassword=secret hladdress=192.0.2.24
 lladdress=1777 crossdefine=yes
```

3. Define the shared library SANGROUP by issuing the **DEFINE LIBRARY** command. You must use the library manager server name in the **PRIMARYLIBMANAGER** parameter, and use LIBTYPE=SHARED.

```
judy> define library sangroup libtype=shared primarylibmanager=astro
```

   Ensure that the library name is the same as the library name on the library manager.

4. Define paths from the library manager, ASTRO, to two drives in the shared library by issuing the **DEFINE PATH** command.

```
astro> define path judy drivea srctype=server desttype=drive
 library=sangroup device=/dev/rmt/6mt
astro> define path judy driveb srctype=server desttype=drive
 library=sangroup device=/dev/rmt/7mt
```

5. Define all device classes that are associated with the shared library.

```
judy> define devclass tape library=sangroup devtype=lto
```

   The following parameters for the device class definition must be the same on the library client as on the library manager:

   - **LIBRARY**
   - **DRIVEENCRYPTION**
   - **WORM**
   - **FORMAT**

6. Define a storage pool named BACKTAPE for the shared library to use. Issue the **DEFINE STGPOOL** command.

```
judy> define stgpool backtape tape maxscratch=50
```

What to do next

Repeat the procedure to define more library clients to your library manager.

Example: Configure library sharing for ACSLS libraries

Use the sample procedure to set up an ACSLS library sharing environment.

About this task

In this example, a library manager server named GLENCOE and a library client named WALLACE are configured. To help clarify where each step is performed, the commands are preceded by the server name from which the command is issued. Most commands are issued from the library client.

Procedure

1. To set up GLENCOE as the library manager server, define a shared ACSLS library named MACGREGOR. For example:

```
glencoe> define library macgregor libtype=acsls shared=yes
```

Then complete the rest of the steps as described in "Example: Configure an ACSLS library with a single drive device type" on page 106 to configure the library.

2. Define GLENCOE as the library manager server by issuing the **DEFINE SERVER** command.

```
wallace> define server glencoe serverpassword=secret hladdress=9.115.3.45
 lladdress=1580 crossdefine=yes
```

3. Define the shared library MACGREGOR, by issuing the **DEFINE LIBRARY** command. You must use the library manager server name in the **PRIMARYLIBMANAGER** parameter, and use LIBTYPE=SHARED.

```
wallace> define library macgregor libtype=shared primarylibmanager=glencoe
```

4. Define paths from the library manager, GLENCOE, to two drives in the shared library by issuing the **DEFINE PATH** command.

```
glencoe> define path wallace drivea srctype=server desttype=drive
 library=macgregor device=/dev/rmt/6mt
glencoe> define path wallace driveb srctype=server desttype=drive
 library=macgregor device=/dev/rmt/7mt
```

5. Define all device classes that are associated with the shared library.

```
wallace> define devclass tape library=macgregor devtype=lto
```

The following parameters for the device class definition must be the same on the library client as on the library manager:

- **LIBRARY**
- **DRIVEENCRYPTION**
- **WORM**
- **FORMAT**

6. Define a storage pool named LOCHNESS for the shared library to use. Issue the **DEFINE STGPOOL** command.

```
wallace> define stgpool lochness tape maxscratch=50
```

What to do next

Repeat the procedure to define more library clients to your library manager.

Example: Configure library sharing for 3494 libraries

Use the sample procedure to set up a 3494 library sharing environment.

About this task

In this example, a library manager server named 3494MGR and a library client named 3494CLI are configured. To help clarify where each step is performed, the commands are preceded by the server name from which the command is issued. Most commands are issued from the library client.

Procedure

1. To set up 3494MGR as the library manager server, define a shared 3494 library named 3494SAN. For example:

```
3494mgr> define library 3494san libtype=349x shared=yes
```

Then complete the rest of the steps as described in "Example: Configure a 3494 library with a single drive device type" on page 108 to configure the library.

2. Define 3494MGR as the library manager server by issuing the **DEFINE SERVER** command.

```
3494cli> define server 3494mgr serverpassword=secret hladdress=9.115.3.45
 lladdress=1580 crossdefine=yes
```

3. Define the shared library 3494SAN, and identify the library manager:

   Note: Ensure that the library name agrees with the library name on the library manager.

```
3494cli> define library 3494san libtype=shared primarylibmanager=3494mgr
```

4. Define paths from the library manager, 3494MGR, to two drives in the shared library by issuing the **DEFINE PATH** command.

```
3494mgr> define path client drivea srctype=server desttype=drive
 library=3494san device=/dev/rmt/4mt
3494mgr> define path client driveb srctype=server desttype=drive
 library=3494san device=/dev/rmt/5mt
```

5. Define the device classes that are associated with the shared library.

```
3494cli> define devclass 3494_class library=3494san devtype=3590
```

6. Define a storage pool named BACKTAPE that will use the shared library.

```
3494cli> define stgpool backtape 3494_class maxscratch=50
```

What to do next

Repeat this procedure to define additional servers as library clients.

Managing virtual tape libraries

A virtual tape library (VTL) does not use physical tape media. When using VTL storage, you can exceed the capabilities of a physical tape library. The ability to define many volumes and drives can provide greater flexibility for the storage environment.

Considerations for using virtual tape libraries

There are some considerations for defining a library as a virtual tape library (VTL), including enhancements for performance and setup of your hardware.

About this task

Defining a VTL to the Tivoli Storage Manager server can help improve performance because the server handles mount point processing for VTLs differently than real tape libraries. The physical limitations for real tape hardware are not applicable to a VTL, affording options for better scalability.

You can use a VTL for any virtual tape library when the following conditions are true:

- There is no mixed media involved in the VTL. Only one type and generation of drive and media is emulated in the library.
- Every server and storage agent with access to the VTL has paths that are defined for all drives in the library.

If either of these conditions are not met, any mount performance advantage from defining a VTL library to the Tivoli Storage Manager server can be reduced or negated.

VTLs are compatible with earlier versions of both library clients and storage agents. The library client or storage agent is not affected by the type of library that is used for storage. If mixed media and path conditions are true for a SCSI library, it can be defined or updated as LIBTYPE=VTL.

Storage capacity for virtual tape libraries

Because virtual tape libraries (VTLs) do not have the physical limitations that real tape hardware does, their capacity for storage is more flexible.

The concept of storage capacity in a virtual tape library is different from capacity in physical tape hardware. In a physical tape library, each volume has a defined capacity, and the library's capacity is defined in terms of the total number of volumes in the library. The capacity of a VTL, alternatively, is defined in terms of total available disk space. You can increase or decrease the number and size of volumes on disk.

This variability affects what it means to run out of space in a VTL. For example, a volume in a VTL can run out of space before reaching its assigned capacity if the total underlying disk runs out of space. In this situation, the server can receive an end-of-volume message without any warning, resulting in backup failures.

When out-of-space errors and backup failures occur, disk space is usually still available in the VTL. It is hidden in volumes that are not in use. For example, volumes that are logically deleted or returned to scratch status in the Tivoli Storage Manager server are only deleted in the server database. The VTL is not notified, and the VTL maintains the full size of the volume as allocated in its capacity considerations.

To help prevent out-of-space errors, ensure that any SCSI library that you update to LIBTYPE=VTL is updated with the `RELABELSCRATCH` parameter set to YES. The `RELABELSCRATCH` option enables the server to overwrite the label for any volume that is deleted and to return the volume to scratch status in the library. The `RELABELSCRATCH` parameter defaults to YES for any library defined as a VTL.

Drive configuration for virtual tape libraries

Drive configuration in a virtual tape library (VTL) is variable, depending on the needs of your environment.

Most VTL environments use as many drives as possible to maximize the number of concurrent tape operations. A single tape mount in a VTL environment is typically faster than a physical tape mount. However, using many drives increases the amount of time that the Tivoli Storage Manager server requires when a mount is requested. The selection process takes longer as the number of drives that are defined in a single library object in the server increases. Virtual tape mounts can take as long or longer than physical tape mounts depending on the number of drives in the VTL.

For best results when you create drives, check with your VTL vendor about device-specific recommendations. If more than 300-500 drives for each VTL are required, you can logically partition the VTL into multiple libraries and assign drives to each library. Operating system and SAN hardware configurations could impose limitations on the number of devices that can be utilized within the VTL library.

Adding a virtual tape library to your environment

Define a virtual tape library (VTL) to take advantage of mount performance and scalability advantages.

About this task

VTLs are identified by using the **DEFINE LIBRARY** command and specifying LIBTYPE=VTL. Because a VTL library functionally interacts with the server in the same way that a SCSI library does, it is possible to use the **UPDATE LIBRARY** command to change the library type of a SCSI library that is already defined. You do not have to redefine the library.

Example

The following examples show how to add a VTL library to your environment.

Add a new VTL library

If you have a new VTL library and want to use the VTL enhancements that are available in Tivoli Storage Manager Version 6.3, define the library as a VTL to the server:

```
define library chester libtype=vtl
```

This sets up the new VTL library and enables the RELABELSCRATCH option to relabel volumes that have been deleted and returned to scratch status.

Update an existing SCSI library to a VTL

If you have a SCSI library and you want to change it to a VTL, use the **UPDATE LIBRARY** command to change the library type:

```
update library calzone libtype=vtl
```

You can only issue this command when the library being updated is defined with LIBTYPE=SCSI.

Revert a real tape library from the VTL library type

If you define a SCSI tape library as a VTL and want to change it back to the SCSI library type, update the library by issuing the **UPDATE LIBRARY** command:

```
update library chester libtype=scsi
```

Defining or deleting all drives and paths for a single library

Use the **PERFORM LIBACTION** command to set up a single SCSI or virtual tape library (VTL) with one step.

About this task

If you are setting up or modifying your hardware environment and must create or change large numbers of drive definitions, the **PERFORM LIBACTION** command can make this task much simpler. You can define a new library and then define all drives and paths to the drives. Or, if you have an existing library that you want to delete, you can delete all existing drives and their paths in one step.

The PREVIEW parameter allows you to view the output of commands before they are processed to verify the action that you want to perform. If you are defining a library, a path to the library must already be defined if you want to specify the PREVIEW parameter. You cannot use the PREVIEW and DEVICE parameters together.

The **PERFORM LIBACTION** command can only be used for SCSI and VTL libraries. If you are defining drives and paths for a library, the SANDISCOVERY option must be supported and enabled. The tape library must be able to return the drive serial number address association.

Procedure

To set up a VTL library named ODIN, complete these steps:

1. Define the library.
   ```
   define library odin libtype=vtl
   ```
2. Define two drives and their paths for your new library, ODIN.
   ```
   perform libaction odin action=define device=/dev/rmt/3lb prefix=dr
   ```

   The server then issues the following commands:
   ```
   define path tsmserver odin srct=server destt=library device=/dev/rmt/3lb
    define drive odin dr0
   define path tsmserver dr0 srct=server destt=drive library=odin
    device=/dev/rmt/1mt define drive odin dr1
   define path tsmserver dr1 srct=server destt=drive library=odin
    device=/dev/rmt/2mt
   ```

Migrating a shared IBM 3494 library to a library manager

If you are sharing an IBM 3494 library among Tivoli Storage Manager servers by using the 3494SHARED server option, you can instead share the library by using a library manager and library clients.

About this task

To facilitate a smooth migration and to ensure that all tape volumes that are being used by the servers are associated with the correct servers, perform the following procedure.

Procedure

1. Complete the following steps on on each server that is sharing the 3494 library:
   a. Update the storage pools by 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 DISABLESCHEDS YES option line if it is not active
      3) Activate the EXPINTERVAL option line if it is not active and set the value to 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.

3. Do the following on the remaining servers (the library clients):
   a. Save the volume history file.
   b. Check out all of the volumes in the library inventory. Use the **CHECKOUT LIBVOLUME** command and set the **REMOVE** parameter to NO.
   c. Set up a library client.
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 and set the **STATUS** to SCRATCH.
5. Halt all the servers.
6. Edit the `dsmserv.opt` file and comment out the following lines in the file:
   ```
   DISABLESCHEDS 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.

About this task

Tivoli Storage Manager uses the capability of the 3494 library manager to partition a library between multiple Tivoli Storage Manager servers. Library partitioning differs from library sharing on a SAN in that 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 in the library is defined to only one server. Each server can then access only the drives that are assigned to it. As a result, library partitioning does not allow dynamic sharing of drives or tape volumes because they are pre-assigned to different servers that use different names and category codes.

To configure static partitioning of drives, complete the following steps. For details and examples of commands that are issued in the procedure, see "Configuring libraries for use by one server" on page 102.

Procedure

1. Set up the library on the system.
2. Define 3494 library devices to the first Tivoli Storage Manager server by completing the following steps:
   a. Define the 3494 library on the server.
   b. Define a path from the server to the library.
   c. Define the drives in the library that are partitioned to the server.
   d. Define a path from the server to each of the drives.

e. Define Tivoli Storage Manager device classes to classify drives according to type.
f. Define a storage pool and associated it with the device class that you created.
g. Verify your device definitions by issuing QUERY commands for each storage object.

3. Repeat the substeps in Step 2 on page 118 to define library devices for the second server.

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.

Procedure

Complete the following steps to export the data from one server and import it to another.

- On 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 by issuing the following command.
     ```
     export node user1 filedata=all devclass=file vol=cdr03
     ```
     This command results in a file named `/home/user1/CDR03` that contains the export data for node USER1. 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.
- On Server B:
  1. Follow the manufacturer's instructions to attach the device to your server.
  2. Oracle Solaris automatically mounts the CD as a removable file system, but the mount options are not correct. Unmount the CD with the following command:

```
umount /cdrom/cdrom0
```

3. Issue this command on your system to mount the CD.

```
mount -F hsfs -o ro,nomaplcase,notraildot /vol/dev/dsk/c0t6d0/cdr03
 /cdrom/cdr03
```

–F hsfs
Specifies that the media has a CD file system

-o Specifies the options on the mount

ro Specifies a read-only file system

nomaplcase
Specifies that file names do *not* map to lower case

notraildot
Specifies that file names do not require a trailing dot when no extension is present

/vol/dev/dsk/c0t6d0/cdr03
Specifies the physical description of the CD on the system

c0 Specifies SCSI controller 0

t6 Specifies SCSI ID 6

d0 Specifies SCSI LUN ID 0

cdr03
Specifies the volume label of the CD

/cdrom/cdr03
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. You must issue the EJECT CDROM command to remove a CD from the drive.

4. 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" on page 121. When you define the drive, the server verifies that a valid file system is present.
5. Define a manual library named CDROM:

```
define library cdrom libtype=manual
```

6. Define the drive in the library:

```
define drive cdrom cddrive
```

7. Define a path from the server to the drive at mount point /cdrom/cdrom0:

```
define path serverb cddrive srctype=server desttype=drive
 library=cdrom device=/cdrom/cdrom0
```

8. Define a device class with a device type of REMOVABLEFILE. The device type must be REMOVABLEFILE.

```
define devclass cdrom devtype=removablefile library=cdrom
```

9. 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
```

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.

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.

Procedure

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 1051 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 172.
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 353 and “How collocation affects reclamation” on page 374.

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 check media into a library 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 30, “Disaster recovery manager,” on page 979.

Migrating media to external media manager control

It is not recommended to 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 you define externally managed devices to the server.

Deleting Tivoli Storage Manager storage pools from externally managed libraries

Before you delete storage pools associated with externally managed libraries, first delete any volumes that are associated with the Tivoli Storage Manager library. For more information, see "Deleting storage pool volumes that contain data" on page 411.

Refer to media manager documentation 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

To configure mounted devices manually, you must first set up the device on the server system.

Procedure

1. Physically attach the device to the server hardware. See "Attaching a manual drive to your system" on page 81.
2. Install and configure the appropriate device driver for the device. See "Selecting a device driver" on page 83 for details.
3. Determine the device name that you need to define the device to Tivoli Storage Manager.

Defining devices as part of a manual library

To manually mount tapes, you must define drives as part of a *manual* library.

About this task

In the following example, two DLT drives are attached to the server system and defined as part of a manual library:

Procedure

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 170 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/rmt/mt1
define path server1 drive02 srctype=server desttype=drive
 library=manualdlt device=/dev/rmt/mt2
```

For more about device special file names, see:

"Device special file names" on page 84

For more information about paths, see "Defining paths" on page 172.

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 device classes" on page 176.

5. Verify your definitions by issuing the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   ```

   See "Requesting information about libraries" on page 151, "Requesting information about drives" on page 153, "Obtaining information about device classes" on page 195, and "Requesting information about paths" on page 165.

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 complete 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. When 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 353 and "How collocation affects reclamation" on page 374.

   See "Defining storage pools" on page 241.

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.

About this task

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:

Procedure

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 252.

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.

About this task

As part of the configuration, a storage agent is installed on the client system. 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

Tivoli Storage Manager supports 349X, ACSLS, VTL, and SCSI tape libraries as well as FILE libraries for LAN-free data movement. The configuration procedure that you follow depends on the type of environment that you implement. However, in all cases you must complete the following steps:

Procedure

1. Verify the network connection.
2. Establish communications among client, storage agent, and Tivoli Storage Manager.
3. Install and configure software on client systems.
4. Configure devices on the server for the storage agent to access.
5. If you are using shared FILE storage, install and configure IBM TotalStorage SAN File System or IBM General Parallel File System .

   Restriction: If a General Parallel File System volume is formatted by an AIX server, the Windows system uses TCP/IP to transfer data and not the storage area network.
6. Define paths from the storage agent to drives.
7. Start the storage agent and verify the LAN-free configuration.

What to do next

For more information on configuring Tivoli Storage Manager for LAN-free data movement, see *Installing and configuring tape-library and file-device-sharing* environments in the *Storage Agent User's Guide*.

Validating your LAN-free configuration

After you configure a Tivoli Storage Manager client for LAN-free data movement, you can verify the configuration and server definitions by using the **VALIDATE LANFREE** command.

About this task

The **VALIDATE LANFREE** command allows you to determine which destinations for a node that is using a specific storage agent are capable of LAN-free data movement. The command output can also help identify if there is a problem with an existing LAN-free configuration. You can evaluate the policy, storage pool, and path definitions for a node and storage agent that the node is using to ensure that an operation is working properly.

The command output displays which management class destinations for an operation type are not LAN-free capable, and provides a brief explanation about why. It also reports the total number of LAN-free destinations.

Procedure

To determine if there is a problem with the client node FRED, that is using the storage agent FRED_STA, issue the **VALIDATE LANFREE** command. For example:

```
validate lanfree fred fred_sta
```

SAN discovery functions for non-root users

To configure Tivoli Storage Manager for LAN-free data movement, you can use the **QUERY SAN** command to obtain information about devices that can be detected on a SAN.

To allow both root and non-root users to perform SAN discovery, a special utility module, `dsmqsan`, is invoked when a SAN-discovery function is launched. The module performs as root, giving SAN-discovery authority to non-root users. While SAN discovery is in progress, `dsmqsan` runs as root.

The `dsmqsan` module is installed by default when the Tivoli Storage Manager server is installed. It is installed with owner root, group system, and mode 4755. The value of the SETUID bit is on. If, for security reasons, you do not want non-root users to run SAN-discovery functions, set the bit to off. If non-root users are having problems running SAN-discovery functions, check the following:

- The SETUID bit must be set to on.
- Device special file permissions and ownership. Non-root users need read/write access to device special files, for example, to tape and library devices.
- The `SANDISCOVERY` option in the server options file must be set to ON

The `dsmqsan` module works only for SAN-discovery functions, and does not provide root privileges for other Tivoli Storage Manager functions.

Impacts of device changes on the SAN

The SAN environment can shift dramatically due to device or cabling changes. This dynamically changing nature of the SAN can cause static definitions to fail or become unpredictable.

Device IDs that are assigned by the SAN and known to the server or storage agent can be altered due to bus resets or other environmental changes. For instance, the server might 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, for example, a new device is added, 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 helps recover from changes to devices on the SAN by using serial numbers to confirm the identity of devices it contacts.

When you define a 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 and can use it 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, attempting 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 that is 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 affect Tivoli Storage Manager. The following are the number ranges for messages that are related to serial numbers:

- ANR8952 through ANR8958
- ANR8961 through ANR8968
- ANR8974 through ANR8975

Restriction: Some devices cannot report 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 must respond to operator requests and manage libraries, drives, disks, paths, and data movers.

About this task

Tasks
"Preparing removable media"
"Labeling removable media volumes" on page 130
"Checking new volumes into a library" on page 133
"Controlling access to volumes" on page 140
"Reusing tapes in storage pools" on page 141
"Reusing volumes used for database backups and export operations" on page 143
"Managing volumes in automated libraries" on page 144
"Managing server requests for media" on page 149
"Managing libraries" on page 151
"Managing drives" on page 153
"Managing paths" on page 165
"Managing data movers" on page 166

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

You must label removable media before it can be used.

About this task

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.

Procedure

Complete the following steps to prepare a volume for use:

1. Label the volume. Any tape 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.

© Copyright IBM Corp. 1993, 2013

For details, see "Defining storage pool volumes" on page 252.

If the storage pool can contain scratch volumes (**MAXSCRATCH** is set to a non-zero value), skip this step.

Labeling removable media volumes

Tape volumes must be labeled before the server can use them. You can use the **LABEL LIBVOLUME** command, or you can use the **AUTOLABEL** parameter with the **DEFINE LIBRARY** and **UPDATE LIBRARY** commands.

About this task

You can use parameters with the **LABEL LIBVOLUME** command to 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 for the following tasks:
  - Prompt for volume names in SCSI libraries
  - Read the barcode label for each cartridge in SCSI, 349X, and automated cartridge system library software (ACSLS) libraries
- Whether to check in the volume for the following tasks:
  - Add the volume to the scratch pool
  - 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 idle volumes that are mounted. If necessary, use the **DISMOUNT VOLUME** command to dismount the idle volume to make that drive available.

By default, the **LABEL LIBVOLUME** command does not overwrite an existing label. However, if you want to overwrite an existing label, you can specify the `OVERWRITE=YES` option.

Attention:

- By overwriting a volume label, you destroy all of the data that is on the volume. Use caution when you overwrite volume labels to avoid deleting valid data.
- The labels on VolSafe volumes can be overwritten only one time. Therefore, use the **LABEL LIBVOLUME** command only one time for VolSafe volumes. You can guard against overwriting the label by using the `OVERWRITE=NO` option with the **LABEL LIBVOLUME** command.

When you use the **LABEL LIBVOLUME** command, you can identify the volumes to be labeled in one of the following ways:

- Explicitly name one volume.
- Enter a range of volumes by using the **VOLRANGE** parameter.
- Use the **VOLLIST** parameter to specify a file that contains a list of volume names or to explicitly name one or more volumes.

For automated libraries, you are prompted to insert the volume in the entry/exit slot of the library. If no I/O convenience station is available, insert the volume into an empty slot. For manual libraries, you are prompted to load the volume directly into a drive.

For information about the **AUTOLABEL** parameter, see "Labeling new volumes using AUTOLABEL" on page 132.

Labeling volumes in a manual drive

To label volumes in a manual drive, issue the LABEL LIBVOLUME command.

About this task

Suppose that you want to label a few new volumes by using a manual tape drive that is defined as the following:

/dev/rmt/5mt

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/rmt/5mt

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.

About this task

Perform the following steps to label volumes one-at-a-time:

Procedure

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.

About this task

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.

About this task

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.

About this task

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 tsmlibname 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.

About this task

The server attempts to label volumes in certain categories to prevent the destruction of data on volumes that are being used by other systems that are connected to the 349X 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 that are already checked into the library.

If you want to label all of the volumes that are in the INSERT category in an IBM TotalStorage 3494 Tape Library, use 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.

About this task

Each volume that is used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries but that are used by the same server.

You can use the **CHECKIN LIBVOLUME** command, or **LABEL LIBVOLUME** command with the CHECKIN option to inform the server that a new volume is available in an automated library. 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:

- Do not mix volumes with barcode labels with volumes without barcode labels in a library. Barcode scanning can take a long time for unlabeled volumes.

- Tivoli Storage Manager accepts only tapes that are 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 website. If you are using non- IBM storage devices and media, consult your tape-cartridge distributor.
- Any volume that has a barcode that begins with `CLN` is treated as a cleaning tape.
- You must use the `CHECKLABEL=YES` option with the **CHECKIN LIBVOLUME** command when you check in VolSafe volumes into a library. This is true for both automated cartridge system library software (ACSLS) and SCSI libraries.
- If you use the `AUTOLABEL=YES` parameter on the **DEFINE LIBRARY** command, you do not need to label tapes before you check them in.
- If a volume has an entry in volume history, you cannot check it in as a scratch volume.

When you check in a volume, you must supply the name of the library and the status of the volume (private or scratch). 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 might need to label more volumes. As volumes are used, you might also need to increase the number of scratch volumes that are 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 that you define must be labeled and checked in. See "Defining storage pool volumes" on page 252.

To check in a volume, you can specify the name of the volume with the **CHECKIN LIBVOLUME** command. To check in several of volumes at the same time, you can use the search capability or the **VOLRANGE** parameter of the **CHECKIN LIBVOLUME** command.

When you use the **CHECKIN LIBVOLUME** command, you might be required to supply some or all of the following information:

Library name
Specifies the name of the library where the storage volume is to be stored.

Volume name
Specifies the volume name to be checked in.

Status Specifies the status that is assigned to the storage volume to be checked in. If the volume that you are checking in is already defined in a storage pool or exists in the volume history file, you must specify `STATUS=PRIVATE`. The `PRIVATE` 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 reads sequential media labels of volumes or use a barcode reader during the **CHECKIN LIBVOLUME** command processing.

Swap Specifies whether Tivoli Storage Manager initiates a swap operation when an empty slot is not available during **CHECKIN LIBVOLUME** command processing.

Mount wait
Specifies the maximum length of time, in minutes, to wait for a volume to be mounted.

Search
Specifies whether Tivoli Storage Manager searches the library for volumes that have not been checked in.

Device type
Specifies the device type for the volume to be checked in. Use this parameter only for 349X libraries in which the drives do not have paths that are defined.

For more information, see the following topics:

- "Checking volumes into a SCSI library one-at-a-time"
- "Checking in volumes in library slots" on page 136
- "Checking in volumes in library entry or exit ports" on page 137
- "Checking media labels" on page 137
- "Allowing swapping of volumes when the library is full" on page 137

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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 129.

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.

About this task

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 you use WORM media.

You can use the following types of WORM media with Tivoli Storage Manager:

- 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

External and manual libraries use separate logical libraries to segregate their media. You must ensure that the correct media is loaded.

Tips:

- A storage pool can consist of either WORM or RW media, but not both.
- To avoid wasting tape after a restore or import operation, avoid using WORM tapes for database backup or export operations.

Related concepts:

"Selecting a device driver" on page 83

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 prelabeled WORM media with the LTO or ECARTRIDGE device class. You cannot use WORM media in IBM LTO-4 drives, HP LTO-4, Oracle StorageTek T10000B, or Oracle StorageTek T10000C 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 (ACSLS) 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.

About this task

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.

About this task

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 primary storage pools” on page 871.

Reusing tapes in storage pools

Tape reuse is controlled by expiration and reclamation processing. You can run either of these processes automatically or manually.

About this task

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 506.

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 363.

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 368.

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.

About this task

Over time, media ages, and some of the backup data that is on it might 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 470
- "Running expiration processing to delete expired files" on page 506
- "File expiration and expiration processing" on page 473

Reusing media - reclamation processing

Data on tapes might 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 363.

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 that are completed 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 that are 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 363.

Ensuring media are available for the tape rotation

Over time, the demand for volumes might 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 141.

For automated libraries, see "Managing server requests for media" on page 149.

Write-once-read-many (WORM) drives can waste media when Tivoli Storage Manager cancels transactions because volumes are not available to complete the backup. After Tivoli Storage Manager writes to WORM volumes, the space on the volumes cannot be reused, even if the transactions are canceled (for example, if a backup is canceled because of a shortage of media in the device).

To minimize wasted WORM media:

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.

If most backups are small files, controlling the transaction size can affect how WORM platters are used. Smaller transactions mean that less space is wasted if a transaction such as a backup must be canceled. Transaction size is controlled by a server option, `TXNGROUPMAX`, and a client option, `TXNBYTELIMIT`.

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.

About this task

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 30, "Disaster recovery manager," on page 979.

For information about the volume history file, see "Protecting the volume history file" on page 866.

Maintaining a supply of scratch volumes

You must set the maximum number of scratch volumes high enough for the expected usage.

About this task

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 141.

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 148.

Maintaining a supply of volumes in a library containing WORM media

For libraries containing write-once-read-many (WORM) media, prevent cancellation of data storage transactions by maintaining a supply of scratch or new private volumes in the library. Canceled transactions can cause wasted WORM media.

About this task

Tivoli Storage Manager cancels (rolls back) a transaction if volumes, either private or scratch, are not available to complete the data storage operation. After Tivoli Storage Manager begins a transaction by writing to a WORM volume, the written space on the volume cannot be reused, even if the transaction is canceled.

For example, if a client starts to back up data and does not have sufficient volumes in the library, Tivoli Storage Manager cancels the backup transaction. The WORM volumes to which Tivoli Storage Manager had already written for the canceled backup are wasted because the volumes cannot be reused. Suppose that you have WORM platters that hold 2.6 GB each. A client starts to back up a 12 GB file. If Tivoli Storage Manager cannot acquire a fifth scratch volume after filling four volumes, Tivoli Storage Manager cancels the backup operation. The four volumes that Tivoli Storage Manager already filled cannot be reused.

To minimize cancellation of transactions:

- Ensure that you have enough volumes available in the library to manage expected client operations such as backup.
  - Verify that you set the maximum number of scratch volumes for the storage pool that is associated with the library to a high enough number.
  - Check enough scratch or private volumes into the library to manage the expected load.
- If your clients tend to store files of smaller sizes, controlling the transaction size can affect how WORM platters are used. Smaller transactions waste less space if a transaction such as a backup must be canceled. The TXNGROUPMAX server option and the TXNBYTELIMIT client option control transaction size.

See “How the server groups files before storing” on page 258 for information.

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.

About this task

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 (ACSLS) 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 133.

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.

About this task

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.

About this task

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:

Procedure

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.

About this task

Perform the following steps to manage a full library:

Procedure

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.*
5. 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.

Results

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.

About this task

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:

```
audit library tapelib checklabel=barcode
```

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.

About this task

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:

Procedure

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.

Results

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 143.

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 10 shows the interaction between library clients and the library manager in processing Tivoli Storage Manager operations.

Table 10. How SAN-enabled servers processTivoli 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.

About this task

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.

Procedure

The following table provides information about how to handle different server media tasks.

Task	Details
Use 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. To start an administrative client in mount mode, use the `dsmadmc -mountmode` command.
Mount volumes for manual libraries	You can monitor mount requests for manual libraries on the Tivoli Storage Manager server console or through an administrative client in mount mode or console mode. A designated operator must respond to the mount requests by putting in tape volumes, as requested. You cannot use the Tivoli Storage Manager **REPLY** command to respond to these messages.
Receive messages for automated libraries	You can see mount messages and messages about problems with automated libraries on administrative clients in mount mode or console mode. 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.
Obtain information about pending operator requests	To get information about pending operator requests, issue the **QUERY REQUEST** command or check the mount message queue on an administrative client that is started in mount mode. When you issue the **QUERY REQUEST** command, Tivoli Storage Manager displays requested actions and the amount of time that is remaining before the requests time out.
Reply to operator requests	When the server requires an explicit reply to a completed mount request, use the **REPLY** command. The **request_number** parameter specifies the request identification number that tells the server which one of the pending operator requests is 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 operator request requires a device to be used for the mount, the second parameter for this command is a device name.
Cancel 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. For most requests that are 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. 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. If you want to mark the requested volume as `UNAVAILABLE`, specify the **PERMANENT** parameter. This action 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.

Task	Details
Respond to a request for volume check-in	If the server cannot find a particular volume to mount 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. If the requested volume is available, put the volume in the library and check it in using the normal procedures. See "Checking new volumes into a library" on page 133. If the requested volume 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. Use the **QUERY REQUEST** command to obtain the ID of the request that you want to cancel. If you do not respond to the checkin request from the server within the mount-wait period that is specified for the device class for the storage pool, the server marks the volume as unavailable.
Determine which volumes are mounted	For a report of all volumes that are 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.
Dismount idle volumes	After a volume becomes idle, the server keeps it mounted for a time that is specified by the mount retention parameter for the device class. Using a mount retention value can reduce the access time if volumes are used repeatedly. 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 mounted. For information about setting mount retention times, see "Controlling the amount of time that a volume remains mounted" on page 179.

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.

About this task

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    Library    Private    Scratch    WORM Scratch     External
Name       Type       Category   Category   Category         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.

About this task

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.

About this task

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=serverr desttype=library device=/dev/rmt/1lb
  ```

  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=2lb,3lb,4lb
  ```

  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.

About this task

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 165.

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.

About this task

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   Drive    Device     On Line
Name      Name     Type
--------  -------  ---------  -------
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.

About this task

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 is located in a SCSI or virtual tape library (VTL)
- 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/rmt/0mt
```

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 "Managing server requests for media" on page 149.

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.

About this task

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 later
- IBM and HP LTO generation 4 and later
- Oracle StorageTek T10000B
- Oracle StorageTek T10000C

Drives must be able to recognize the correct format. With Tivoli Storage Manager, you can use the following encryption methods:

Table 11. Encryption methods supported

	Application method	**Library method**	**System method**
3592 generation 2 and later	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
Oracle StorageTek T10000B	Yes	No	No
Oracle StorageTek T10000C	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 drives that are 3592 generation 2 and later" on page 182
- "Encrypting data using LTO generation 4 tape drives" on page 190
- "Enabling ECARTRIDGE drive encryption" on page 193 and "Disabling ECARTRIDGE drive encryption" on page 194

Validating data during read/write operations to tape

To validate data and identify data that is corrupted, you can use a feature called *logical block protection*. If you use logical block protection, Tivoli Storage Manager inserts a cyclic redundancy check (CRC) value at the end of each logical block of data to be written to tape.

About this task

With logical block protection, you can identify errors that occur while data is being written to tape and while data is transferred from the tape drive to Tivoli Storage Manager through the storage area network. Drives that support logical block protection validate data during read and write operations. The Tivoli Storage Manager server validates data during read operations.

If validation by the drive fails during write operations, it can indicate that data was corrupted while being transferred to tape. The Tivoli Storage Manager server fails the write operation. You must restart the operation to continue. If validation by the drive fails during read operations, it can indicate that the tape media is corrupted. If validation by the Tivoli Storage Manager server fails during read operations, it can indicate that data was corrupted while being transferred from the tape drive and the server tries the operation again. If validation fails consistently, the Tivoli Storage Manager server issues an error message that indicates hardware or connection problems.

If logical block protection is disabled on a tape drive, or the drive does not support logical block protection, the Tivoli Storage Manager server can read protected data. However, the data is not validated.

Logical block protection is superior to the CRC validation that you can specify when you define or update a storage pool definition. When you specify CRC validation for a storage pool, data is validated only during volume auditing operations. Errors are identified after data is written to tape.

Restriction: You cannot use logical block protection for sequential data such as backup sets and database backups.

Drives that support logical block protection

Logical block protection is available only for 3592, LTO, and ECARTRIDGE device types. Capable 3592 drives include IBM TS1130, TS1140, and later generations. Capable LTO drives include IBM LTO-5 and supported LTO-6 drives. Capable Oracle StorageTek drives include those with the T10000C format.

The following table shows the media and the formats that you can use with drives that support logical block protection.

Drive	Tape media	Drive formats
IBM TS1130	3592 Generation 2	3592-3 and 3592-3C
IBM TS1140	3592 Generation 2 3592 Generation 3	Generation 2: 3592-3 and 3592-3C Generation 3: 3592-4 and 3592-4C
IBM LTO-5	LTO-5	ULTRIUM5 and ULTRIUM5C
IBM LTO-6	LTO-6 LTO-5	ULTRIUM6 and ULTRIUM6C ULTRIUM5 and ULTRIUM5C
Oracle T10000C	Oracle StorageTek T10000 T2	T10000C and T10000C-C

Tip: If you have a 3592, LTO, or Oracle StorageTek drive that is not capable of logical block protection, you can upgrade the drive with firmware that provides logical block protection.

Logical block protection is only available for drives that are in MANUAL, SCSI, 349*x*, and ACSLS libraries. Logical block protection is not available for drives that are in external libraries. For the most current information about support for logical block protection, see http://www.ibm.com/support/docview.wss?uid=swg21568108.

To use logical block protection for write operations, all the drives in a library must support logical block protection. If a drive is not capable of logical block protection, volumes that have read/write access are not mounted. However, the server can use the drive to mount volumes that have read-only access. The protected data is read and validated by the Tivoli Storage Manager server if logical block protection is enabled for read/write operations.

Enabling and disabling logical block protection

You can specify logical block protection for read and write operations, or only for write operations. You can also disable logical block protection if it is enabled. By default, logical block protection is disabled because of performance effects that result from CRC validation on the server and the tape drive.

About this task

Read/write operations to empty or filling volumes depend on whether the volumes have logical block protection. Protected and unprotected data blocks cannot be mixed on the same volume. If you change the setting for logical block protection, the change applies only to empty volumes. Filling and full volumes maintain their status of logical block protection until they are empty and ready to

be refilled. For example, suppose that you disable logical block protection. If the server selects a volume that is associated with a device class that has logical block protection, the server continues writing protected data to the volume.

To enable logical block protection, specify the **LBPROTECT** parameter on the **DEFINE DEVCLASS** or the **UPDATE DEVCLASS** command for the 3592, LTO, and ECARTRIDGE device types:

Procedure

- To enable logical block protection, specify a value of READWRITE or WRITEONLY for the **LBPROTECT** parameter.

  For example, to specify logical block protection during read/write operations for a 3592 device class named 3592_lbprotect, issue the following command:

  ```
  define devclass 3592_lbprotect library=3594 lbprotect=readwrite
  ```

  Tips:

  - If you update the value of the **LBPROTECT** parameter from NO to READWRITE or WRITEONLY and the server selects a filling volume without logical block protection for write operations, the server issues a message each time the volume is mounted. The message indicates that data will be written to the volume without logical block protection. To prevent this message from displaying or to have Tivoli Storage Manager only write data with logical block protection, update the access of filling volumes without logical block protection to read-only.
  - To reduce the performance effects, do not specify the **CRCDATA** parameter on the **DEFINE STGPOOL** or **UPDATE STGPOOL** command.
  - When data is validated during read operations by both the drive and by the Tivoli Storage Manager server, it can slow server performance during restore and retrieval operations. If the time that is required for restore and retrieval operations is critical, you can change the setting of the **LBPROTECT** parameter from READWRITE to WRITEONLY to increase the restore or retrieval speed. After data is restored or retrieved, you can reset the **LBPROTECT** parameter to READWRITE.

- To disable logical block protection, specify a value of NO for the **LBPROTECT** parameter.

  Restriction: If logical block protection is disabled, the server does not write to an empty tape with logical block protection. However, if a filling volume with logical block protection is selected, the server continues to write to the volume with logical block protection. To prevent the server from writing to tapes with logical block protection, change access of filling volumes with logical block protection to read-only. When data is read, the CRC on each block is not checked by either drive or the server.

  If a disaster occurs and the disaster recovery site does not have drives that support logical block protection, you must set the **LBPROTECT** parameter to NO. If the tape drives are used for write operations, you must change the volume access for volumes with protected data to read-only to prevent the server from using the volumes.

  If the server is to enable logical block protection, the server issues an error message that indicates that the drive does not support logical block protection.

What to do next

To determine whether a volume has logical block protection, issue the **QUERY VOLUME** command and verify the value in the field `Logical Block Protection`.

Read/write operations to volumes with logical block protection

Read/write operations to empty or filling volumes depend on whether the volumes have logical block protection. Protected and unprotected data blocks cannot be mixed on the same volume.

If you use the **UPDATE DEVCLASS** command to change the setting for logical block protection, the change applies only to empty volumes. Filling and full volumes maintain their status of logical block protection until they are empty and ready to be refilled.

For example, suppose that you change the value of the **LBPROTECT** parameter from READWRITE to NO. If the server selects a volume that is associated with the device class and that has logical block protection, the server continues writing protected data to the volume.

Remember:

- Before you select the volume, the Tivoli Storage Manager server does not verify whether the volume has logical block protection.
- If a drive does not support logical block protection, the mounts of volumes with logical block protection for write operations fail. To prevent the server from mounting the protected volumes for write operations, change the volume access to read-only. Also, disable logical block protection to prevent the server from enabling the feature on the tape drive.
- If a drive does not support logical block protection, and logical block protection is disabled, the server reads data from protected volumes. However, the data is not validated by the server and the tape drive.

To determine whether a volume has logical block protection, issue the **QUERY VOLUME** command and verify the value in the field `Logical Block Protection`.

Tip: Consider updating the access of filling volumes to read-only if you update the value of the **LBPROTECT** parameter in one of the following ways:

- READWRITE or WRITEONLY to NO
- NO to READWRITE or WRITEONLY

For example, suppose that you change the setting of the **LBPROTECT** parameter from NO to READWRITE. If the server selects a filling volume without logical block protection for write operations, the server issues a message each time the volume is mounted. The message indicates that data will be written to the volume without logical block protection. To prevent this message from being displayed or to have Tivoli Storage Manager only write data with logical block protection, update the access of filling volumes without logical block protection to read-only.

Storage pool management

To mix protected and unprotected data in a library, you must create different device classes and different storage pools to separate the data. The device classes associated with the protected data can specify logical block protection for read/write operations or only for write operations.

Suppose, for example, that you have a 3584 library that has LTO-5 drives and that you want to use for protected and unprotected data. To define the required device classes and storage pools, you can issue the following commands.

```
define library 3584 libtype=scsi
define devclass lbprotect library=3584 devicetype=lto lbprotect=readwrite
define devclass normal library=3584 devicetype=lto lbprotect=no
define stgpool lbprotect_pool lbprotect maxscratch=10
define stgpool normal_pool normal maxscratch=10
```

Cleaning drives

The server can control cleaning tape drives in SCSI libraries and offers partial support for cleaning tape drives in manual libraries.

About this task

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 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.

Tip: Use library based cleaning for automated tape libraries that support this function.

Device manufacturers that include library-cleaning functions recommend its use to prevent premature wear on the read/write heads of the drives.

Drives and libraries from manufacturers differ in how they manage cleaner cartridges, and how they report the presence of a cleaner cartridge in a drive. The device driver might 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 not known to applications, therefore, Tivoli Storage Manager might not always detect the cleaner cartridges in drives and might not be able to determine when cleaning begins.

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 might be to prevent the library drive cleaning from functioning effectively. If this happens, use Tivoli Storage Manager to control drive cleaning. You can set the frequency to match the cleaning recommendations from the manufacturer.

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.

Library based cleaning provides several advantages for automated tape libraries that support this function:

- Library based cleaning lowers the burden on the Tivoli Storage Manager administrator to manage cleaning cartridges.
- It can improve cleaning cartridge usage rates. Most tape libraries track the number of cleans left based on the hardware indicators. Tivoli Storage Manager uses a raw count.
- Unnecessary cleaning is reduced. Modern tape drives do not need cleaning at fixed intervals and can detect and request when cleaning is required.

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 might 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 might not always detect cleaner cartridges in drives and might not be able to determine when cleaning begins.

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 that is 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 cannot 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 configure server-controlled drive cleaning in an automated library, you can specify how often you want the drives to be cleaned.

About this task

Complete the following steps to configure server-controlled drive cleaning in an automated library:

Procedure

1. Define or update the drives in a library, by using the **CLEANFREQUENCY** parameter in the **DEFINE DRIVE** and **UPDATE DRIVE** commands.

   The **CLEANFREQUENCY** parameter sets how often you want the drive to be cleaned. Consult the documentation that accompanies the drives for recommendations on cleaning frequency.

Remember: The **CLEANFREQUENCY** parameter is not valid for externally managed libraries, for example, 3494 libraries or Oracle StorageTek libraries that are managed under automated cartridge system library software (ACSLS).
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 your drive manufacturer documentation for cleaning recommendations. If the documentation gives recommendations for cleaning frequency in terms of hours of use, convert to a gigabytes value by completing the following steps:

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 number of 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 that is recommended by the product documentation will not overclean the drives.
- The `CLEANFREQUENCY=ASNEEDED` parameter value does not work for all tape drives. You can determine whether a drive supports this function at http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html. 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 volume inventory with the **CHECKIN LIBVOLUME** command.

```
checkin libvolume autolib1 cleanv status=cleaner cleanings=10 checklabel=no
```

After the cleaner cartridge is checked in, the server mounts the cleaner cartridge in a drive when the drive needs cleaning. The server uses the cleaner cartridge for the number of cleanings specified. See "Checking in cleaner cartridges" and "Operations with cleaner cartridges in a library" on page 163 for more information.

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.

About this task

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:

```
ANR89141I 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:

```
ANR89141I 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 147.

Deleting drives

You can delete drive definitions by issuing the DELETE DRIVE command.

About this task

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 "Managing server requests for media" on page 149. 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.

About this task

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.

About this task

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/rmt/1lb
  ```

- **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/rmt/1mt
```

Deleting paths

You can delete an existing path definition by issuing the DELETE PATH command.

About this task

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.

About this task

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.

About this task

For example, to update the data mover for the node named NAS1 to change the IP address, issue the following command:

```
update datamover nas1 hladdress=9.67.97.109
```

Deleting data movers

You can delete an existing data mover definition by issuing the DELETE DATAMOVER command.

About this task

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. Tips for defining devices

Learn more details about defining the Tivoli Storage Manager storage objects.

About this task

Defining devices and paths

The following topics describe how to define libraries, drives, and paths to Tivoli Storage Manager.

About this task

See “Managing libraries” on page 151, “Managing drives” on page 153, and “Managing paths” on page 165 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.

About this task

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 84.

```
define path server1 robotmount srctype=server desttype=library
 device=/dev/rmt/0lb
```

For more information about paths, see “Defining paths” on page 172.

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
```

© Copyright IBM Corp. 1993, 2013

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.

About this task

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 127.

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.

About this task

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 is located.

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 or virtual tape library (VTLs). 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 enable 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 enable 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 might not be able to automatically detect the serial number. In this case, the server does not record a serial number for the device, and is not 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 127.

Element address

The element address of the drive. The ELEMENT parameter applies only to drives in SCSI or VTL 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. The server can 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 might 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 website 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=/rmt/1mt
```

For more information about paths, see "Defining paths" on page 172.

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.

About this task

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:

For example, if you have a SCSI type library named AUTODLTLIB that has a device name of /dev/rmt/3lb, define the path to the server named ASTRO1 by doing the following:

```
define path astro1 autodltlib srctype=server desttype=library
device=/dev/rmt/3lb
```

If you have a drive, DRIVE01, that resides in library AUTODLTLIB, and has a device name of /dev/rmt/4mt, define it to server ASTRO1 by doing the following:

```
define path astro1 drive01 srctype=server desttype=drive library=autodltlib
 device=/dev/rmt/4mt
```

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.

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

LAN-free access to 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.

Learn more from the IBM General Parallel File System Information Center at: http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfsbooks.html.

Defining device classes

A device class defines a device type for a set of volumes that can be created in a storage pool. Device classes are also important for storing database backups and for exporting and importing data.

About this task

Sequential-access device types include tape, 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 want to specify different attributes for different storage pools that use the same type of tape drive. Variations might 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 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 by using the **DELETE DEVCLASS** and **QUERY DEVCLASS** commands.

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 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 device classes” on page 176
“Defining 3592 device classes” on page 180
“Device classes for devices not supported by the Tivoli Storage Manager server” on page 183
“Defining device classes for removable media devices” on page 183

Tasks
“Defining sequential-access disk (FILE) device classes” on page 184
“Defining LTO device classes” on page 188
“Defining SERVER device classes” on page 191
“Defining device classes for StorageTek VolSafe devices” on page 192
“Defining device classes for CENTERA devices” on page 194
“Obtaining information about device classes” on page 195
“How Tivoli Storage Manager fills volumes” on page 196

For details about commands and command parameters, see the *Administrator's Reference*.

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

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

See “Mixed device types in libraries” on page 57 and “Mixed device types in libraries” on page 57 for more 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

With Tivoli Storage Manager you can use tape devices, magnetic disk devices, removable media devices, and virtual volumes.

The following tables list the valid devices, media types, and Tivoli Storage Manager device types.

For details and updates, see the following website: http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html

Table 12. 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
Oracle StorageTek SD-3, 9490, 9840, 9940, and T10000 drives	Tape cartridges	ECARTRIDGE

Table 12. Tape devices (continued)

Examples	Media type	Device type
Tape drives that can be used by the operating system device drivers	Tape cartridges	GENERICTAPE
IBM 3580	LTO Ultrium cartridges	LTO
Tape drives that can be used by the NAS file server for backups	Unknown	NAS
Oracle StorageTek 9840 drives	Write-once read-many (WORM) tape cartridges	VOLSAFE

Table 13. Magnetic disk devices

Examples	Media type	Device type
Sequential-access disk	File system or storage volumes	FILE
EMC Centera	File system or storage volumes	CENTERA

Table 14. 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 15. Virtual volumes

Examples	Media type	Device type
Tivoli Storage Manager target server	Storage volumes or files that are archived on another Tivoli Storage Manager server	SERVER

Defining tape device classes

Device class definitions for tapes include parameters that allow you to control storage operations.

Specifying the estimated capacity of tape volumes

Tivoli Storage Manager also uses estimated capacity to determine when to begin reclamation storage pool volumes.

About this task

For tape device classes, the default values selected by the server depend on the recording format that is used to write data to the volume. You can either accept the default for a 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 196.

Specifying recording formats for tape media

You can specify the recording format that is used by Tivoli Storage Manager when writing data to tape media.

About this task

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 "Example: Configure a SCSI or virtual tape library with multiple drive device types" on page 104.

The recording format that Tivoli Storage Manager uses for a 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.

About this task

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 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.

About this task

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 “Preempting operations” on page 602.

  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.

About this task

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 601
- "Managing server requests for media" on page 149

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.

About this task

To control the wait time for a drive to become available for a mount request, use the **MOUNTWAIT** parameter when you define or update a device class.

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 192

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.

About this task

The following table shows read-and-write interoperability for drive generations.

Drives	Generation 1 format	Generation 2 format	Generation 3 format	Generation 4 format
Generation 1	Read and write	n/a	n/a	n/a
Generation 2	Read and write	Read and write	n/a	n/a
Generation 3	Read only	Read and write	Read and write	n/a
Generation 4	n/a	Read only	Read and write	Read and write

If you must mix generations of drives in a library, use one of the methods in the following table to prevent or minimize the potential for problems.

Important: Mixed drive generations is not the optimal configuration. If possible, limit the drives in a library to a single generation.

Mixing generations of drives
(349X, ACSLS, SCSI libraries) If your library contains two drive generations, force all drives to use the format of the earlier generation. For example, if your library contains generation 3 and generation 4 drives, force all the generation 4 drives to use the generation 3 format. The result is that all the drives can read and write to all the media. **Remember:** If you force a drive generation to write in the format of an earlier drive generation, both drive generations can verify labels and read media written in the format of the earlier drive generation. For example, if you library contains generation 2 and generation 3 drives, both drive generations can verify labels and read media written in the generation 2 format. However, this configuration does not allow the generation 3 drives to read or write in their optimal format. If your library contains three drive generations, the latest drive generation in your library can only read media from the earliest format, but cannot write with it. For example, if your library contains generation 2, generation 3, and generation 4 drives, the generation 4 drives can only read the generation 2 format. In this configuration, mark all media previously written in generation 2 format to read-only.
(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.

Mixing generations of drives
(SCSI libraries only) Define a new storage pool and device class for the latest drive generation. For example, suppose you have a storage pool and device class for 3592-2. The storage pool will contain all the media written in generation 2 format. Suppose that the value of the FORMAT parameter in the device class definition is set to 3952-2 (not DRIVE). You add generation 3 drives to the library. Complete the following steps: 1. In the new device-class definition for the generation 3 drives, set the value of the FORMAT parameter to 3592-3 or 3592-3C. Do not specify DRIVE. 2. In the definition of the storage pool associated with generation 2 drives, update the MAXSCRATCH parameter to 0, for example: `update stgpool genpool2 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 188.) 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 or later 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 or later 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 or later 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.

About this task

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 257.

Encrypting data with drives that are 3592 generation 2 and later:

With Tivoli Storage Manager, you can use the following types of drive encryption with drives that are 3592 generation 2 and later: 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 drives that are 3592 generation 2 and later. Use this parameter to ensure Tivoli Storage Manager compatibility with hardware encryption settings for empty volumes.

About this task

- 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:

Procedure

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
   ```

Results

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 530.

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.

Device classes for devices not supported by the Tivoli Storage Manager server

To use a tape device that is not supported by the Tivoli Storage Manager server, you must define a device class whose device type is GENERICTAPE.

For a manual library with multiple drives of device type GENERICTAPE, ensure that the device types and recording formats of the drives are compatible. Because the devices are controlled by the operating system device driver, the Tivoli Storage Manager server is not aware of the following:

- The actual type of device: 4 mm, 8 mm, digital linear tape, and so forth. For example, if you have a 4 mm device and an 8 mm device, you must define separate manual libraries for each device.
- The actual cartridge recording format. For example, if you have a manual library defined with two device classes of GENERICTAPE, ensure the recording formats are the same for both drives.

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.

About this task

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, */dev/cdx* (*x* is a number that is assigned by your operating system)

For more information, see:

"Removable file device configuration" on page 119

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.

About this task

To define a FILE device class, use the DEVTYPE=FILE parameter in the device class definition.

Do not use raw partitions with a device class type of FILE.

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 for 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 that is associated with a FILE-type device class. In addition, one client session or one server process 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 ACTIVEDATA
- 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.

About this task

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.

About this task

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.

About this task

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 systems" 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.

About this task

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

Procedure

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.

Results

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.

About this task

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.

About this task

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 “Preempting operations” on page 602.

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.

About this task

If you are considering mixing different generations of LTO media and drives, be aware of the following restrictions:

Table 16. 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.

About this task

For more information about using drive encryption, refer to "Encrypting data on tape" on page 530.

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.

About this task

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:

Procedure

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.

About this task

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 722.

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.)

Procedure

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/rmt/0mt
    library=volsafelib
   ```

4. Define a device class:

```
define devclass volsafeclass library=volsafelib devtype=ecartridge
format=drive worm=yes
```

Results

For more information about VolSafe media, see "Write-once, read-many tape media" on page 138.

Enabling ECARTRIDGE drive encryption

The DRIVEENCRYPTION parameter specifies whether drive encryption is enabled or can be enabled for DRIVE, T10000B, T10000B-C, T10000C, and T10000C-C formats. This parameter ensures Tivoli Storage Manager compatibility with hardware encryption settings for empty volumes.

About this task

Tivoli Storage Manager supports the Application method of encryption with Oracle StorageTek T10000B or T10000C 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 or VolSafe 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:

Procedure

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 531

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.

Defining device classes for CENTERA devices

To use a Centera device, you must define a device class whose device type is CENTERA.

Concurrent access to Centera volumes

Concurrent access improves performance while restoring or retrieving data because two or more clients can access the same volume at the same time.

Multiple client retrieve sessions, restore sessions, or server processes can read a volume concurrently in a storage pool that is associated with the CENTERA device type. In addition, one client session or one server process can write to the volume while it is being read.

The following server processes can share read access to Centera volumes:

- EXPORT NODE
- EXPORT SERVER
- GENERATE BACKUPSET

The following server processes cannot share read access to Centera volumes:

- AUDIT VOLUME
- DELETE VOLUME

Server operations not supported by Centera

Centera storage devices do not support some Tivoli Storage Manager server operations.

The following server operations are not supported:

- Data-movement operations:
  - Moving node data into or out of a Centera storage pool.
  - Migrating data into or out of a Centera storage pool.
  - Reclaiming a Centera storage pool.
  - LAN-free data movement or Network Data Management Protocol (NDMP) operations. Centera storage pools cannot be the target or source of data for either of these operations.
- Backup operations:
  - Backing up a Centera storage pool.
  - Using a Centera device class to back up a database.
  - Backing up a storage pool to a Centera storage pool.
- Restore operations:
  - Restoring data from a copy storage pool or an active-data pool to a Centera storage pool.
  - Restoring volumes in a Centera storage pool.
- Other:
  - Exporting data to a Centera device class or importing data from a Centera device class. However, files stored in Centera storage pools can be exported and files being imported can be stored on Centera.

- Using a Centera device class for creating backup sets; however, files stored in Centera storage pools can be sent to backup sets.
- Defining Centera volumes.
- Using a Centera device class as the target of volume history, device configuration, trace logs, error logs, or query output files.
- Using a Centera device class as the target for a virtual volume operation.
- Data deduplication.
- Copying active versions of backup data either to or from a Centera storage pool.

Controlling the number of concurrently open mount points for Centera devices

You can control the number of mount points that can be opened concurrently for access by server storage and retrieval operations. Any attempts to access more mount points than indicated by the mount limit causes the requester to wait.

About this task

When you select 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, the Tivoli Storage Manager server is currently performing a client backup request to an output volume and another request from another client to restore data from the same volume. The backup request is preempted and the volume is released for use by the restore request. For more information, see “Preempting operations” on page 602.

To control the number of mount points concurrently open for Centera devices, use the **MOUNTLIMIT** parameter when you define the device class or update its definition.

Obtaining information about device classes

You can choose to view a standard or detailed report for a device class.

About this task

Task	Required Privilege Class
Request information about device classes	Any administrator

To display a standard report on device classes, enter:

```
query devclass
```

Figure 11 on page 196 provides an example of command output.

```
Device       Device        Storage    Device     Format     Est/Max    Mount
Class        Access           Pool    Type                  Capacity   Limit
Name         Strategy        Count                             (MB)
---------    ----------    -------    -------    ------    --------   -----
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 11. Example of a standard device class report

To display a detailed report on the GEN1 device class, enter:

```
query devclass gen1 format=detailed
```

Figure 12 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 12. 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 418 and "Registering nodes with the server" on page 416.

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 13 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 380.

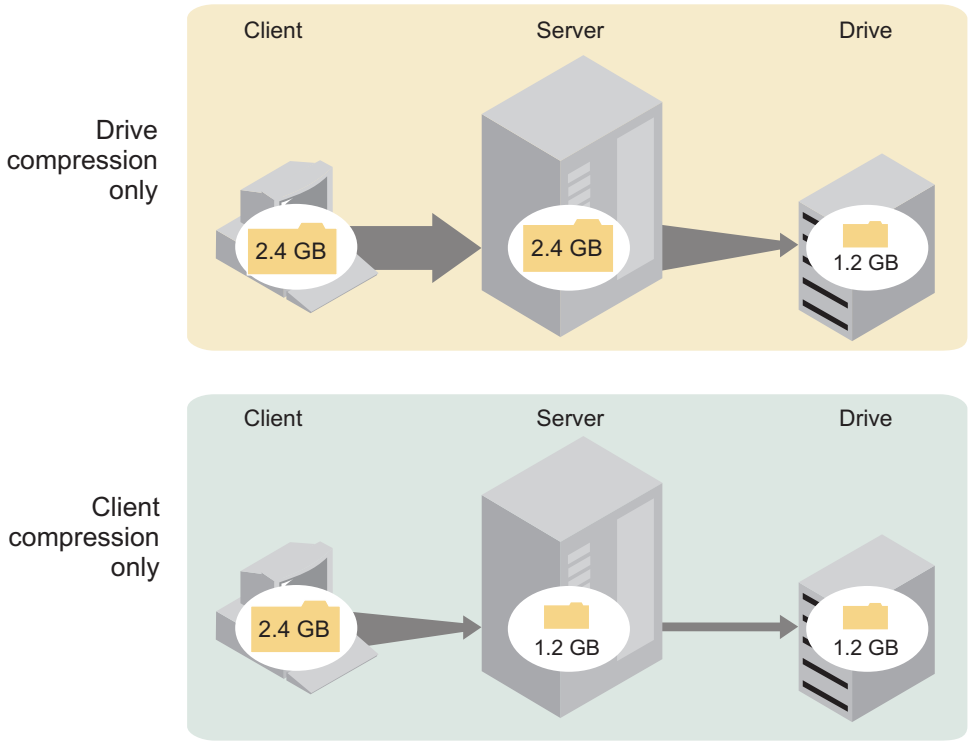


Figure 13. Comparing compression at the client and compression at the device

For how to set up compression on the client, see "Node compression considerations" on page 418 and "Registering nodes with the server" on page 416.

Chapter 9. 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 207
"Determining the location of NAS backup" on page 209
"Configuring a tape library for NDMP operations" on page 213
"Configuring Tivoli Storage Manager policy for NDMP operations" on page 208
"Registering NAS nodes with the Tivoli Storage Manager server" on page 219
"Defining a data mover for the NAS file server" on page 220
"Defining paths to libraries for NDMP operations" on page 223
"Defining paths to libraries for NDMP operations" on page 223
"Defining paths for NDMP operations" on page 220
"Labeling and checking tapes into the library" on page 224
"Scheduling NDMP operations" on page 224
"Defining virtual file spaces" on page 224
"Tape-to-tape copy to back up data" on page 225
"Tape-to-tape copy to move data" on page 225
"Backing up and restoring NAS file servers using NDMP" on page 226
"Backing up NDMP file server to Tivoli Storage Manager server backups" on page 228
"Managing tables of contents" on page 205
"NDMP operations management" on page 202
"Managing NAS file server nodes" on page 203
"Managing data movers used in NDMP operations" on page 204
"Storage pool management for NDMP operations" on page 205

NDMP requirements

You must meet certain requirements when you use 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."

© Copyright IBM Corp. 1993, 2013

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 run tests to ensure that 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 three types of libraries for operations that use NDMP. The libraries that are supported are SCSI, ACSLS (automated cartridge system library software), and VTL (virtual tape library). 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. 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 be directly connected only 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.

VTL library

A virtual tape library that is supported by the Tivoli Storage Manager server. This type of library can be attached directly either to the Tivoli Storage Manager server or to the NAS file server. A virtual tape library is essentially the same as a SCSI library but is enhanced for virtual tape library characteristics and allows for better mount performance.

Note: If you are defining a VTL library, your environment must not include any mixed-media. Paths must be defined between all drives in the library and all defined servers, including storage agents, that use the library. If either of these characteristics are not true, the overall performance can degrade to the same levels as the SCSI library type; especially during times of high stress.

349X library

A 349X library can be directly connected only 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 running NDMP operations can be a library manager for either an ACSLS, SCSI, VTL, 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 runs 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, VTL, 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 continues to support all NDMP backup and restore operations with a NAS device that runs NDMP version 3. The Tivoli Storage Manager server negotiates the highest protocol level (either Version 3 or Version 4) with the NDMP server when it establishes an NDMP connection. If you experience any issues with Version 4, you might want to try 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 Oracle Solaris system)
- web client

Server Interfaces:

- Server console
- Command line on the administrative client

Tip: All examples in this chapter use server commands.

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 229 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.

About this task

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.

Procedure

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.

About this task

For example, to rename NASNODE1 to NAS1 you must perform the following steps:

Procedure

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.

Procedure

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.

About this task

For example, if you shut down a NAS file server for maintenance, you might want to take the data mover offline.

Procedure

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.

Procedure

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.

You cannot designate a Centera storage pool as a target pool of NDMP operations.

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 were defined for NDMP operations in traditional Tivoli Storage Manager operations. Be especially careful when you assign 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 tables of contents

You can use several commands to manage different aspects of your data contents.

About this task

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 can 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 473 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.

Preventing long-running, inactive NDMP connections from closing

To prevent firewalls from closing network data-management protocol (NDMP) connections that are long-running but inactive, you can enable Transmission Control Protocol (TCP) keepalive on the NDMP control connections.

About this task

The Tivoli Storage Manager server initiates control connections to network-attached storage (NAS) devices during NDMP backup or restore operations. These control connections might remain open and inactive for an extended amount of time. For example, suppose that two NDMP operations are started for the same NAS device. The control connection for one NDMP operation might remain open but inactive if the operation requires a resource, for example, a tape drive or sequential volume, that is being used by the other NDMP operation.

Some firewall software is configured to automatically close network connections that are inactive for a specified length of time. If a firewall exists between a Tivoli Storage Manager server and a NAS device, it is possible that the firewall can close NDMP control connections unexpectedly and cause the NDMP operation to fail.

The Tivoli Storage Manager server provides a mechanism, TCP keepalive, that you can enable to prevent long-running, inactive connections from being closed. If TCP keepalive is enabled, small packets are sent across the network at predefined intervals to the connection partner.

Restriction: To prevent errors, do not enable TCP keepalive in certain types of environments. One example is environments that do not have firewalls between the Tivoli Storage Manager server and a NAS device. Another example is environments with firewalls that tolerate long-running, inactive connections. Enabling TCP keepalive in these types of environments can cause an idle connection to be inadvertently closed if the connection partner temporarily fails to respond to TCP keepalive packets.

Enabling TCP keepalive

To enable TCP keepalive, which keeps NDMP connections open, use the NDMPENABLEKEEPALIVE server option.

Procedure

Add the option to the server options file dsmserv.opt:

```
ndmpenablekeepalive yes
```

Example

To update the server option, you can use the **SETOPT** command.

Specifying connection idle time for TCP keepalive

HP-UX and Oracle Solaris operating systems do not provide Tivoli Storage Manager with a method for setting the connection idle time before the first TCP keepalive packet is sent. To set the idle time on HP-UX and Oracle Solaris operating systems, you must issue an operating system command.

Procedure

To specify connection idle time, issue the following operating system command:

```
ndd -set /dev/tcp tcp_keepalive_interval milliseconds
```

For example, to specify an idle time of 900,000 milliseconds (15 minutes) before the first TCP keepalive packet is transmitted, issue the following command:

```
ndd -set /dev/tcp tcp_keepalive_interval 900000
```

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.

Procedure

1. Set up the tape library and media. See "Configuring a tape library for NDMP operations" on page 213, where the following steps are described in more detail.
   a. Attach the SCSI or virtual tape library (VTL) 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, VTL, 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 208.
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 219.
4. Define a data mover for the NAS file server. See "Defining a data mover for the NAS file server" on page 220.

5. Define a path from either the Tivoli Storage Manager server or the NAS file server to the library. See "Defining paths to libraries for NDMP operations" on page 223.
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 paths for NDMP operations" on page 220.
7. Check tapes into the library and label them. See "Labeling and checking tapes into the library" on page 224.
8. Set up scheduled backups for NAS file servers. This step is optional. See "Scheduling NDMP operations" on page 224.
9. Define a virtual file space name. This step is optional. See "Defining virtual file spaces" on page 224.
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 225.
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 225.

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.

About this task

See "Configuring policy for NDMP operations" on page 518 for more information. Complete the following steps to configure Tivoli Storage Manager policy for NDMP operations:

Procedure

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 policyset 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 219.

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 459 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 17 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 17. 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.

Table 17. Comparing methods for backing up NDMP data (continued)

Property	NDMP: Filer to server	NDMP: Filer to attached library	Backup-archive client to 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.
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

Table 17. Comparing methods for backing up NDMP data (continued)

Property	NDMP: Filer to server	NDMP: Filer to attached library	Backup-archive client to server
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
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.

About this task

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 14 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.

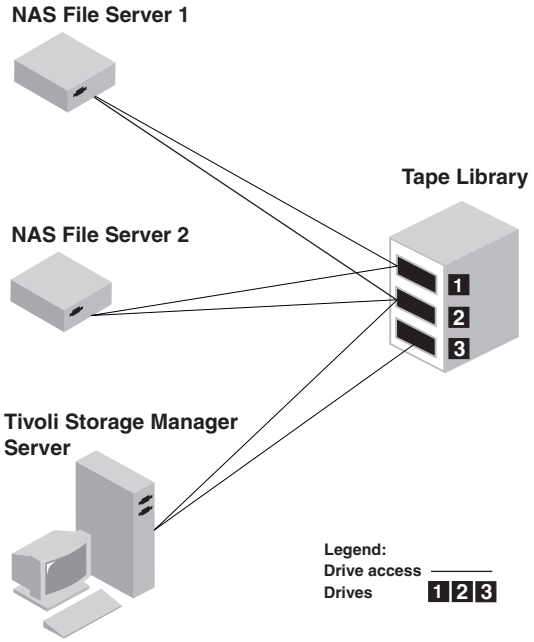


Figure 14. Tivoli Storage Manager drive usage example

To create the configuration shown in Figure 14, perform the following steps:

Procedure

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.

Results

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 paths for NDMP operations" on page 220 for more information.

Configuring a tape library for NDMP operations

You can configure a tape library to backup a network-attached storage (NAS) device to tape.

About this task

Perform the following steps to set up tape libraries for NDMP operations:

Procedure

1. Connect the library and drives to be used for NDMP operations.
   a. Connect the SCSI library. Before you set up a SCSI tape library for NDMP operations, determine whether you want to attach your library robotics control to the Tivoli Storage Manager server or to the NAS file server. See "Tape libraries and drives for NDMP operations" on page 211. Connect the SCSI tape library robotics to the Tivoli Storage Manager server or to the NAS file server. Refer to your device manufacturer documentation for instructions.

      If the library is 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.

      If the library is 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 you library device by issuing the **DEFINE LIBRARY** command. The library must be a single device type, not a mixed device. Issue one of the following commands to define the library depending on the type of device that you are configuring:

   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 your NDMP device by issuing the **DEFINE DEVCLASS** command.

Tip: A device class that is defined with a device type of NAS is not explicitly associated with a specific drive type, for example, LTO. However, as best practice, define a separate device classes for different drive types.

In the **DEFINE DEVCLASS** command, use the following parameters and values:

- Specify DEVTYPE=NAS.
- Specify MOUNTRETENTION=0 . It 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 with an estimated capacity is 40 GB for the media, issue the following command:

```
define devclass nasclass devtype=nas library=naslib mountretention=0
 estcapacity=40g
```

4. Define a storage pool for NDMP media by issuing the **DEFINE STGPOOL** command. When NETAPPDUMP, CELERRADUMP, or NDMPDUMP is designated as the type of storage pool, managing the storage pools that are produced by NDMP operations is different from managing storage pools that contain media for traditional Tivoli Storage Manager backups. Tivoli Storage Manager operations use storage pools that are 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 best practice, 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 or 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 are defined for NDMP operations in traditional Tivoli Storage Manager operations. Be especially careful when you assign 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 can fail.

5. Optionally, 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.

   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 about defining volumes, see "Configuring random access volumes on disk devices" on page 76.

For more information about connecting libraries, see Chapter 5, "Attaching devices for the server," on page 81.

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.

About this task

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 18. 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 4 (ACSLS library)	May be limited by ACSLS 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.

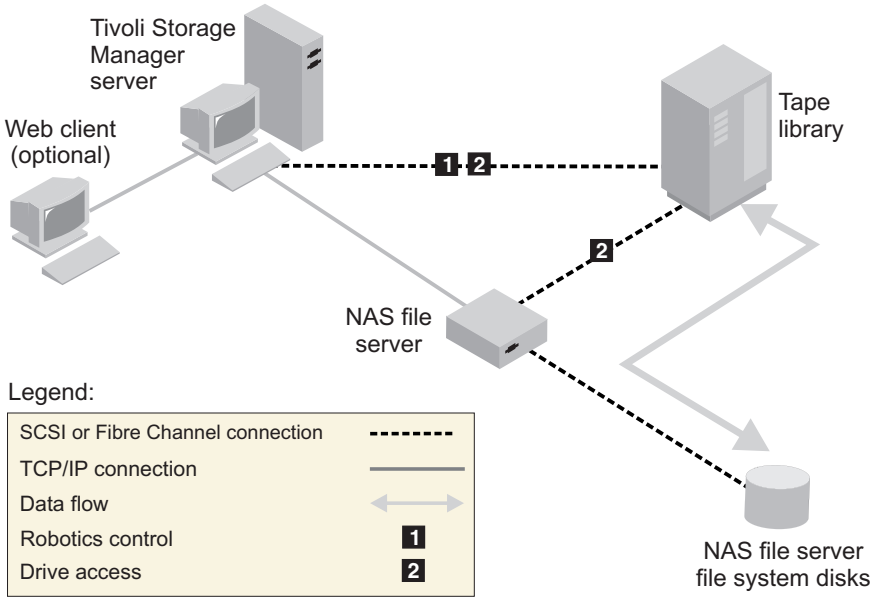


Figure 15. Configuration 1: SCSI library connected to Tivoli Storage Manager server

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.

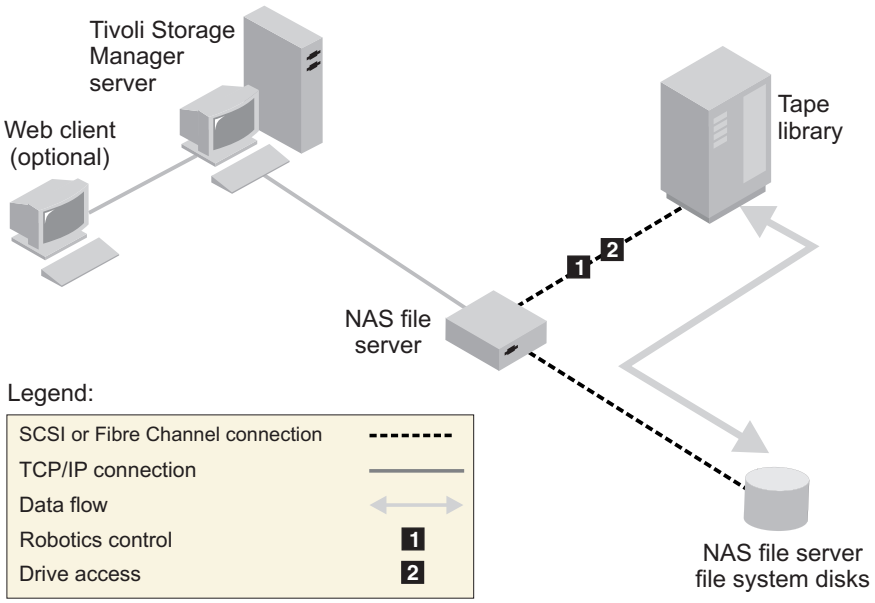


Figure 16. Configuration 2: SCSI library connected to the NAS file server

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.

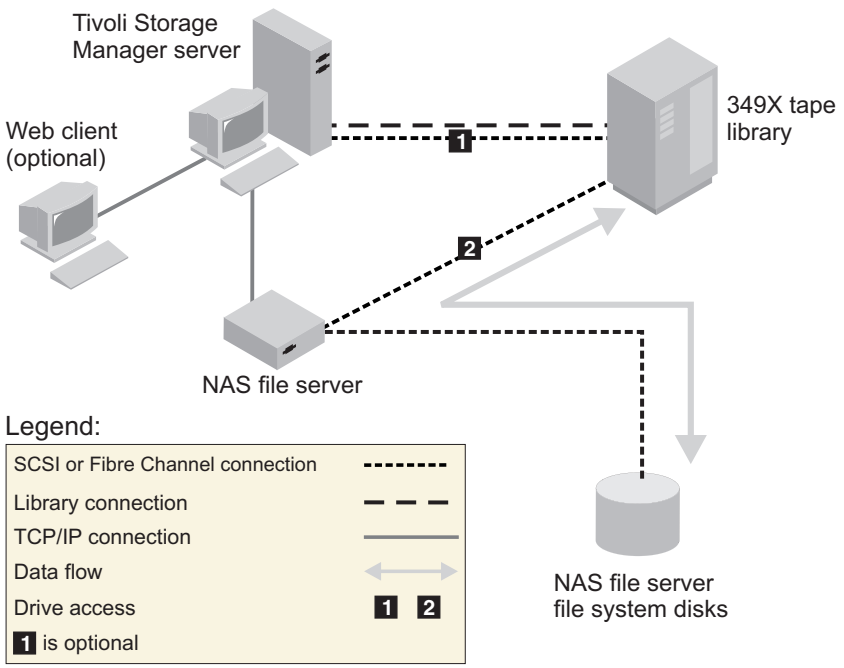


Figure 17. Configuration 3: 349x library connected to the Tivoli Storage Manager server

See Chapter 5, "Attaching devices for the server," on page 81 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.

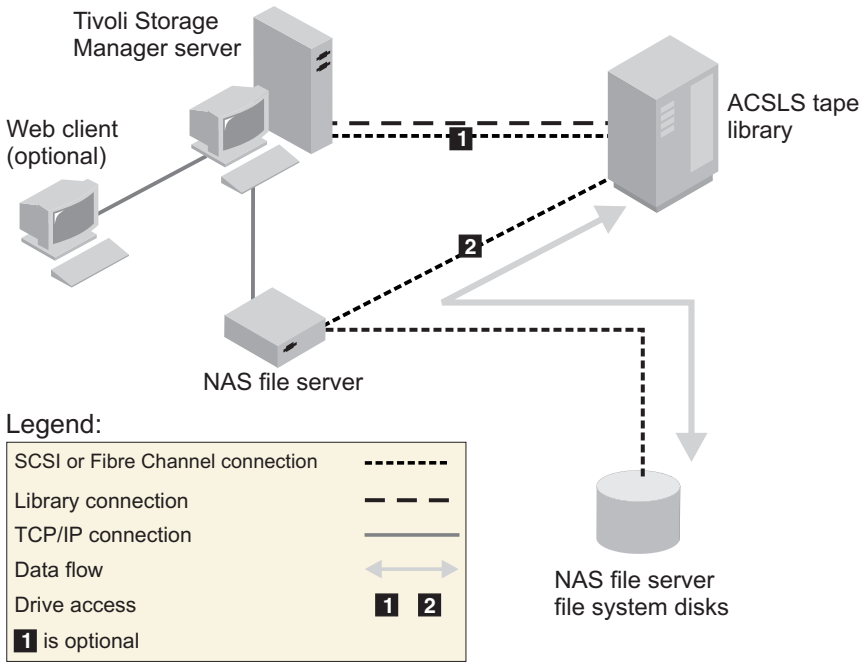


Figure 18. Configuration 4: ACSLS library connected to the Tivoli Storage Manager server

See Chapter 5, "Attaching devices for the server," on page 81 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.

Procedure

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.

Procedure

To define a data mover for a NAS node named NASNODE1, enter the following example command:

```
define datamover nasnode1 type=nas hladdress=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 paths for NDMP operations

For network data management protocol (NDMP) operations, you create paths to drives and to libraries.

Defining paths to drives for NDMP operations

The method that you choose for creating paths to drives depends on whether the drives are accessed by a network-attached storage (NAS) file server and the Tivoli Storage Manager server or only by a NAS file server.

Defining paths for drives attached only to a NAS file server and to the Tivoli Storage Manager server:

If a tape drive is to be accessed by a network-attached storage (NAS) file server and the Tivoli Storage Manager server, you must create two paths. One path exists between the tape drive and the NAS file server. The other path exists between the tape drive and the Tivoli Storage Manager server.

About this task

Complete the following steps:

Procedure

1. If the drive is not defined for the Tivoli Storage Manager server, create the drive definition. For example, to define a drive NASDRIVE1 for a library NASLIB, issue the following command:

   ```
   define drive naslib nasdrive1 element=autodetect
   ```

Remember: If the drive is attached to the Tivoli Storage Manager server, the element address is automatically detected .

2. Map the NAS drive name to the corresponding drive definition on the Tivoli Storage Manager server:
   - On the Tivoli Storage Manager server, issue the **QUERY DRIVE FORMAT=DETAILED** command to obtain the worldwide name (WWN) and serial number for the drive that is to be connected to the NAS file server.
   - On the NAS device, obtain the tape device name, serial number, and WWN for the drive.

   If the WWN or serial number matches, a drive on a NAS file server is the same as the drive on the Tivoli Storage Manager server .
3. Using the drive name, define a path to the drive from the NAS file server and a path to the drive from the Tivoli Storage Manager server.
   - For example, to define a path between a tape drive with a device name of rst01 and a NetApp file server, issue the following command:

     ```
     define path nasnode1 nasdrive1 srctype=datamover desttype=drive
       library=naslib device=rst01
     ```
   - To define a path between the tape drive and the Tivoli Storage Manager server, issue the following command:

     ```
     define path server1 nasdrive1 srctype=server desttype=drive
      library=naslib device=/dev/rmt/0mt
     ```

Related information:

Obtaining device names for devices attached to NAS file servers

Defining paths for drives attached only to NAS file servers:

If a tape drive is to be accessed only by a network-attached storage (NAS) file server and not by the Tivoli Storage Manager server, only a single path between the tape drive and the NAS file server is required.

About this task

Complete the following steps:

Procedure

1. Obtain the SCSI element addresses, worldwide name (WWN), and serial numbers for the drive to be connected to NAS file server.

   Restriction: If the SCSI drive is connected only to a NAS file server, the element address is not automatically detected, and you must supply it. If a library has more than one drive, you must specify an element address for each drive.
   To obtain a SCSI element address, go to one of the following Tivoli device-support websites:
   - AIX, HP-UX, Solaris, and Windows: http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html
   - Linux: http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_Linux.html

   Element number assignment and device WWN assignments are also available from tape-library device manufacturers.

2. Create drive definitions by specifying the element addresses identified in the preceding step. Specify the element address in the **ELEMENT** parameter of the **DEFINE DRIVE** command. For example, to define a drive NASDRIVE1 with the element address 82 for the library NASLIB, issue the following command:

   ```
   define drive naslib nasdrive1 element=82
   ```

   Attention: For a drive connected only to the NAS file server, do not specify ASNEEDED as the value for the **CLEANFREQUENCY** parameter of the **DEFINE DRIVE** command.
3. Obtain the device name, serial number, and WWN for the drive on the NAS device.
4. Using the information obtained in steps 1 and 3, map the NAS device name to the element address in the drive definition in the Tivoli Storage Manager server.
5. Define a path between the tape drive and the NAS file server. For example, to define a path between a NetApp file server and a tape drive with a device name of rst01, issue the following command:

   ```
   define path nasnode1 nasdrive1 srctype=datamover desttype=drive
    library=naslib device=rst0l
   ```

Related information:

Obtaining device names for devices attached to NAS file servers

Obtaining names for devices attached to NAS file servers:

For paths from a network-attached storage (NAS) data mover, the value of the **DEVICE** parameter in the **DEFINE PATH** command is the name by which the NAS file server knows a library or drive.

About this task

You can obtain these device names, also 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.

Procedure

- To obtain the device 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 device 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.
- 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
  ```
- For fibre-channel-attached drives and the Celerra data mover, complete the following steps:

1. Log on to the EMC Celerra control workstation using an administrative ID. Issue the following command:

   ```
   server_devconfig server_1 -l -s -n
   ```

   Tip: The `-l` option for this command lists only the device information that was saved in the database of the data mover. The command and option do not display changes to the device configuration that occurred after the last database refresh on the data mover. For details about how to obtain the most recent device configuration for your data mover, see the EMC Celerra documentation.
   The output for the **server_devconfig** command includes the device names for the devices attached to the data mover. The device names are listed in the *addr* column, for example:

   ```
   server_1:
   Scsi Device Table
   name        addr      type   info
   tape1       c64t0l0   tape   IBM ULT3580-TD2 53Y2
   ttape1      c96t0l0   tape   IBM ULT3580-TD2 53Y2
   ```

2. Map the Celerra device name to the device worldwide name (WWN):
   a. To list the WWN, log on to the EMC Celerra control workstation and issue the following command. Remember to enter a period (.) as the first character in this command.

      ```
      .server_config server_# -v "fcp bind show"
      ```

      The output for this command includes the WWN, for example:

      ```
      Chain 0064: WWN 500507630f418e29 HBA 2 N_PORT Bound
      Chain 0096: WWN 500507630f418e18 HBA 2 N_PORT Bound
      ```

      Note: The **.server_config** command is an undocumented EMC Celerra command. For more information about how to use it, contact EMC.
   b. Use the chain number to identify the tape device that was listed in the output of the server_devconfig command and that has the same WWN, for example:

Tape device name	Chain number	WWN
c64t0l0	0064	500507630f418e29
c96t0l0	0096	500507630f418e18

Celerra commands might behave differently on different EMC Celerra systems and operating system levels. For details, see the EMC Celerra documentation or contact EMC.

Defining paths to libraries for NDMP operations

Define a path to the SCSI library from either the Tivoli Storage Manager server or the network-attached storage (NAS) file server.

Procedure

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/rmt/1lb
   ```

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: The **DEFINE PATH** command is not needed for an automated cartridge system library software (ACSLS) library.

Labeling and checking tapes into the library

You must label the tapes and check them into the tape library.

About this task

These tasks are the same as for other libraries. For more information, see:

"Labeling removable media volumes" on page 130

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.

About this task

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 226 for information about the commands.

Procedure

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 18, "Automating server operations," on page 609 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.

About this task

Procedure

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 232 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.

About this task

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.

Procedure

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.

About this task

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.

Procedure

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.

Procedure

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 232. 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 "Backing up NDMP file server 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.

Backing up NDMP file server 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.

About this task

To back up a server on a NAS file system, complete the following steps:

Procedure

1. Select an existing storage pool or set up a storage pool for the NAS data by issuing the following command:

   ```
   define stgpool naspool disk
   ```

2. Define volumes to add to the storage pool. For example, define a volume named naspool_volAB:

   ```
   define volume naspool /usr/storage/naspool_volAB formatsize=100
   ```

3. 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 policyset 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.

4. 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.

5. Back up the NAS device to the Tivoli Storage Manager storage pool by issuing the following command:

   ```
   backup node nas1 /vol/vol0
   ```

6. 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 tables of contents” on page 205 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.

About this task

For example, to backup a snapshot created for a NetApp file system, perform the following:

Procedure

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 virtualfsmapping  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 virtualfsmapping  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 large NetApp file systems by using the NetApp SnapMirror 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 large NetAppfile systems to auxiliary storage. For most NetAppfile systems, use the standard NDMP full or differential backup method.

Using a parameter option on the **BACKUP NODE** and **RESTORE NODE** commands, you can back up and restore file systems by 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 Operations 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 file space.
- 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. NetAppprovides 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. NetAppprovides 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 about 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.

Replicating NAS nodes with NDMP backup data

You can replicate a network-attached storage (NAS) node that contains NDMP (network data management protocol) backup data.

Only NDMP backup data in NATIVE data format storage pools can be replicated. You cannot replicate NDMP images that are stored in storage pools that has the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

A differential NDMP backup image requires the associated full backup to be restored. Therefore, to replicate a differential NDMP backup image, both full and differential images must be in NATIVE data format storage pools. A full NDMP image in a NATIVE data format storage pool is always eligible for replication. If an NDMP image is replicated, its associated Table of Contents data is also replicated.

When an NDMP image is replicated to a replication target server, a data mover definition must be created on the target replication server. To create a data mover definition, issue the **DEFINE DATAMOVER** command to restore the data.

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.

About this task

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 247.

Concepts
"Storage pools" on page 236
"Storage pool volumes" on page 249
"Access modes for storage pool volumes" on page 255
"Storage pool hierarchies" on page 256
"Migrating files in a storage pool hierarchy" on page 267
"Caching in disk storage pools" on page 280
"Writing data simultaneously to primary, copy, and active-data pools" on page 327
"Keeping client files together using collocation" on page 353
"Reclaiming space in sequential-access storage pools" on page 363
"Estimating space needs for storage pools" on page 375

Tasks
"Defining storage pools" on page 241
"Preparing volumes for random-access storage pools" on page 251
"Preparing volumes for sequential-access storage pools" on page 252
"Defining storage pool volumes" on page 252
"Updating storage pool volumes" on page 254
"Setting up a storage pool hierarchy" on page 257
"Monitoring storage-pool and volume usage" on page 378
"Monitoring the use of storage pool volumes" on page 380

© Copyright IBM Corp. 1993, 2013

Tasks
“Moving data from one volume to another volume” on page 396
“Moving data belonging to a client node” on page 402
“Renaming storage pools” on page 405
“Defining copy storage pools and active-data pools” on page 405
“Deleting storage pools” on page 409
“Deleting storage pool volumes” on page 410

For details about devices, see:

Chapter 4, “Magnetic disk devices,” on page 69

Chapter 6, “Configuring storage devices,” on page 101

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.

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 onsite.

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 might 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 offsite and still have the server track the volumes. Moving copy storage pool volumes offsite provides a means of recovering from an onsite 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.
- You cannot back up data from a primary storage pool that is defined with a CENTERA device class.

For details about copy storage pools, see:

- "Restoring storage pools and storage pool volumes" on page 895
- "Backing up primary storage pools" on page 871
- "Scenario: Recovering a lost or damaged storage pool volume" on page 898
- "Ensuring the integrity of files" on page 885
- "Backing up the data in a storage hierarchy" on page 261
- "Setting up copy storage pools and active-data pools" on page 262
- "Backing up primary storage pools" on page 871

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 onsite or offsite storage volumes, or reducing bandwidth when copying or restoring files that are vaulted electronically in a remote location.

Data that is 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 must be replaced but cannot find it in the active-data pool. As a best practice and to protect your inactive data, 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 must 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 that is 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, offer similar benefits. Although tapes must 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 that are used for onsite and offsite storage. If you vault data electronically to a remote location, an active-data pool that is associated with a SERVER device class can save bandwidth by copying and restoring only active data.

Remember:

- The server does 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 that is defined with the NETAPPDUMP, the CELERRADUMP, or the NDMPDUMP data format.
- You cannot copy active data from a primary storage pool that is defined with a CENTERA device class.

For details about active-data pools, see:

- "Backing up the data in a storage hierarchy" on page 261
- "Setting up copy storage pools and active-data pools" on page 262
- "Copying active versions of client backup data to active-data pools" on page 239
- "Selection for active file versions" on page 239

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 ACTIVEDATA 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.

About this task

You can issue the **COPY ACTIVEDATA** command either manually or in an administrative schedule or maintenance script.

Regardless whether you use the **COPY ACTIVEDATA** 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.

Selection for active file versions

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 that is associated with a FILE device class
2. A random-access disk (DISK) storage pool
3. A primary or copy storage pool that is associated with a FILE device class
4. A primary, copy, or active-data pool that is associated with onsite or offsite removable tape media

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.

About this task

Figure 19 on page 240 shows one way to set up server storage. In this example, the storage that is 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 that are defined in management classes direct the server to store files from clients in the ARCHIVE, BACKUP, or HSM disk storage pools. An extra 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 256.

For more information about backing up primary storage pools, see "Backing up primary storage pools" on page 871.

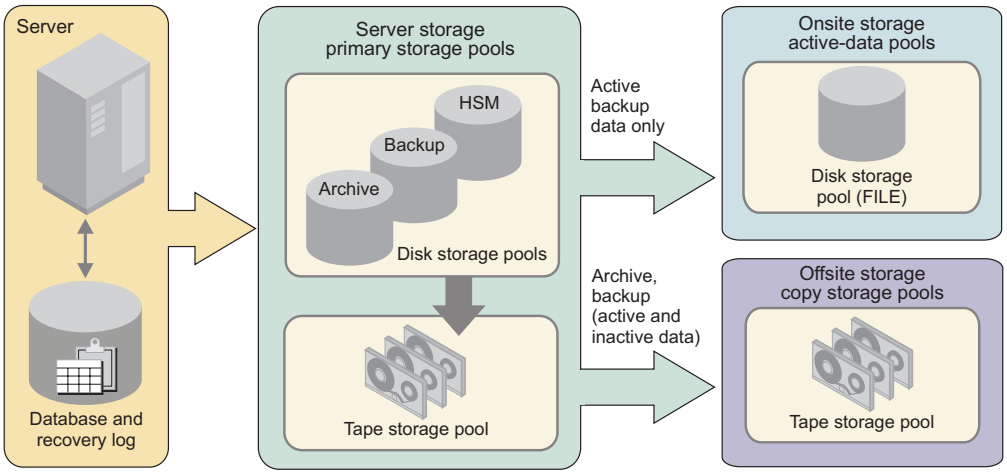


Figure 19. Example of server storage

To set up this server storage hierarchy, do the following:

Procedure

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 did not already do so.

   For more information, see

   "Configuring random access volumes on disk devices" on page 76
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 472 for details.

   Define another policy that specifies the active-data pool that you create as the destination for active data.
3. Assign nodes to the domains. Nodes whose active data you want to restore quickly must 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 101

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 247.
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 405.
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 primary storage pools" on page 871.

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.

About this task

Tip: When you define or update 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 19 on page 242. 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 19. 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
Maximum number of scratch volumes [2] (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 148	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] [2]	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 878. 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] [2]	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 256.	random, sequential

Table 19. Information for defining a storage pool (continued)

Information	Explanation	Type of Storage Pool
Migration thresholds [1] [2]	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 267.	random, sequential
Migration processes [1] [2]	Specifies the number of concurrent processes to use for migrating files from this storage pool. See "Migrating disk storage pools" on page 269 and "Specifying multiple concurrent migration processes" on page 278.	random, sequential
Migration delay [1] [2]	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 274 and "How the server migrates files from sequential-access storage pools" on page 276.	random, sequential
Continue migration process [1] [2]	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 274 and "How the server migrates files from sequential-access storage pools" on page 276.	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 280.	random
Collocation [2]	With collocation enabled, the server tries to keep all files belonging on a minimal number of sequential-access storage volumes. The files can belong to a single client node, a group of client nodes, a client file space, or a group of file spaces. See "Keeping client files together using collocation" on page 353.	sequential
Reclamation threshold [1] [2]	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 366.	sequential
Reclamation processes [1] [2]	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 367.	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 offsite volumes" on page 371.	sequential
Reclamation storage pool [1] [2]	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 368.	sequential
Reuse delay period [2]	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 874.	sequential
Overflow location [1] [2]	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 146	sequential

Table 19. Information for defining a storage pool (continued)

Information	Explanation	Type of Storage Pool
Data Format [2]	The format in which data will be stored. NATIVE is the default data format. NETAPPDUMP and NONBLOCK are examples of other data formats.	sequential
Copy Storage Pools [1 2]	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 and "Writing data simultaneously to primary, copy, and active-data pools" on page 327 for related information. **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.	random, sequential
Copy Continue [1 2]	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. This option has no effect on active-data pools.	sequential
Active-data pools [1 2]	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. **Attention:** The ACTIVEDATAPOOLS parameter is not intended to replace the COPY ACTIVEDATA command. If you use the simultaneous-write function, ensure that the copy of active backup data is complete by regularly issuing the COPY ACTIVEDATA command. If you do not issue the COPY ACTIVEDATA 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.	random, sequential
Shredding	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 533.	random

Table 19. Information for defining a storage pool (continued)

Information	Explanation	Type of Storage Pool

[1] This information is not available for sequential-access storage pools that use the following data formats:

- CELERRADUMP
- NDMPDUMP
- NETAPPDUMP

[2] This information is not available or is ignored for Centera sequential-access storage pools.

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.

About this task

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, complete the following steps:

Procedure

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
```

Results

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 9, "Using NDMP for operations with NAS file servers," on page 199.
- The Tivoli Storage Manager server does not support the following functions for Centera storage pools:
  - Data-movement operations:
    - Moving node data into or out of a Centera storage pool.
    - Migrating data into or out of a Centera storage pool.
    - Reclaiming a Centera storage pool.
  - Backup operations:
    - Backing up a Centera storage pool.
    - Using a Centera device class to back up a database.
    - Backing up a storage pool to a Centera storage pool.
    - Copying active data to an active-data pool.
  - Restore operations:
    - Restoring data from a copy storage pool or an active-data pool to a Centera storage pool.
    - Restoring volumes in a Centera storage pool.
  - Other:
    - Exporting data to a Centera device class or importing data from a Centera device class; however, files stored in Centera storage pools can be exported and files being imported can be stored on Centera.
    - Using a Centera device class for creating backup sets; however, files stored in Centera storage pools can be sent to backup sets.
    - Defining Centera volumes.
    - Using a Centera device class as the target of volume history, device configuration, trace logs, error logs, or query output files.

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.

About this task

In this example, the ENGBACK1 disk storage pool is defined as shown in "Example: Defining storage pools" on page 245. 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 20 gives tips on how to accomplish some tasks that are related to storage pools.

Table 20. 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 353
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 353
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 327
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 363
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 269

Table 20. Task tips for storage pools (continued)

For this Goal	Do This	For More Information
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 269
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 241 "Setting up copy storage pools and active-data pools" on page 262
Back up your storage pools.	Implement a copy storage pool.	"Setting up copy storage pools and active-data pools" on page 262
Copy active data from a primary storage pool.	Implement an active-data pool.	"Setting up copy storage pools and active-data pools" on page 262
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 241 "Changing policy" on page 472
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 367 "Specifying multiple concurrent migration processes" on page 278
Ensure that reclamation completes within the desired amount of time.	Limit the number of off-site volumes to be reclaimed.	"Reclamation of offsite volumes" on page 371 "Starting reclamation manually or in a schedule" on page 367
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 251 "Defining storage pool volumes" on page 252
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 251 "Defining storage pool volumes" on page 252

Storage pool volumes

Storage pool volumes are the physical media that are assigned to a storage pool.

The following list contains examples of a volume:

- Space that is allocated on a disk drive
- A tape cartridge

Storage pool volumes are either random access or sequential access, depending on the device type of the device class to which the storage 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 one of the following:

- Fixed-size files on a disk. You can create these files using the DEFINE VOLUME command, or you can use space triggers to automatically create and assign them to specified storage pools.
- Raw partitions on a disk, which must be defined before the server can access them.

For additional information, see:

"Preparing volumes for random-access storage pools" on page 251

"Requirements for disk systems" 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 252.

Types of sequential-access volumes

Each Tivoli Storage Manager sequential-access device type is associated with a particular type of storage pool volume.

The following list contains some examples of sequential-access volumes:

- Tape cartridge
- File

Table 21 on page 250 lists the types of volumes that are associated with each device type.

Table 21. 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
CENTERA	A logical collection of files that are stored on the Centera storage device	No
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 an operating system	No
GENERICTAPE	A tape that is compatible with the drives that are defined to the device class	Yes
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
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

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.

About this task

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

Procedure

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 /opt/tivoli/tsm/server/bin/stgvol.001 formatsize=21
```

Results

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.

Another option for preparing a volume is to create a raw partition on a formatted disk. For details, see the format command for the operating system. When creating the partition with the /etc/format utility, do not include cylinder 0 (zero) in the partition that is intended for use as a raw partition. On the Solaris 2.5.1 operating system, the partition must be less than 2 GB. After you create the partition, you define the volume to Tivoli Storage Manager.

Attention: The server cannot detect if data already exists on a raw partition. Existing data can be destroyed if you define a volume to that partition.

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.

About this task

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:

Procedure

1. Label the volume. Table 21 on page 250 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 130.

   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 133.
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 253 for information about scratch volumes.

Results

Defining storage pool volumes

You can use defined volumes to 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.

About this task

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 that is used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that are 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 by 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 are exceeded. One volume must be predefined.

Remember: You cannot define volumes for storage pools that are defined with a Centera device class.

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.

About this task

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 252.

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.

About this task

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 255.

Volume properties that you can update

Update volume properties by changing the values of those properties in the volume definition.

Table 22 lists volume properties that you can update.

Table 22. 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" on page 255 for descriptions of access modes. A random-access volume must be varied offline before you can change its access mode to *unavailable* or *destroyed*. To vary a volume offline, use the VARY command. See "Varying disk volumes online or offline" on page 78. 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.

Table 22. Information for updating a storage pool volume (continued)

Information	Explanation
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 78.

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

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 78. 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.

About this task

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 181.

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 353.

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.

About this task

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. It does not include copy storage pools or active-data pools. See "Backing up the data in a storage hierarchy" on page 261.
- A storage pool must use the NATIVE or NONBLOCK data formats to be part of a storage pool hierarchy. For example, a storage pool that uses 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 267.

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.

About this task

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:

Procedure

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.

About this task

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. You can update the storage pool by using the **UPDATE STGPOOL** command.

To define the new tape storage pool and update the hierarchy:

Procedure

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, in the server options file, affects the number of files allowed. TXNBYTELIMIT, 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 might 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 you transfer 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, and 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 675.

  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.

Server file aggregation is disabled for client nodes storing data that is associated with a management class that has a copy group whose destination is a Centera storage pool.

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.

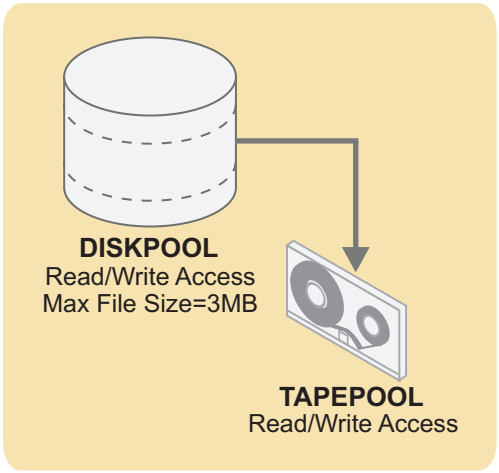


Figure 20. Storage hierarchy example

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.

Before you begin

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.

About this task

"Setting up copy storage pools and active-data pools" on page 262 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 primary storage pools" on page 871.)

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.

About this task

To set up a copy storage pool or an active-data pool:

Procedure

1. Define a copy storage pool or active-data pool. For details, see "Defining copy storage pools and active-data pools" on page 405.
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 492.

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 on page 262. For details about assigning nodes to a domain, see "Assigning client nodes to a policy domain" on page 506.
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 327.

   Ensure that client sessions have sufficient mount points. Each session requires one mount point for the primary storage pool and a mount point for each copy storage pool and each active-data pool. To allow a sufficient number of mounts points, use the **MAXNUMMP** parameter on the **REGISTER NODE** or **UPDATE NODE** commands.
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 610 and "Tivoli Storage Manager server scripts" on page 616.

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.

Before you begin

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.

About this task

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).
- Two mount points assigned for each client session.
- 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:

Procedure

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, ADPPOOL, where the server simultaneously will write data during a client backup operation:
   ```
   UPDATE STGPOOL BACKUPPOOL ACTIVEDATAPOOLS=ADPPOOL
   ```

   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 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.

Results

Every time NODE2 stores data into BACKUPPOOL, the server simultaneously writes the data to ADPPOOL. 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 ADPPOOL. 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, ADPPOOL, 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.

About this task

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).
- Two mount points for each client session.
- A device class of type 3592 named 3592CLASS.

To set up and enable an active-data pool, follow these steps:

Procedure

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)* To set up simultaneous write operations, complete the following steps:
   - 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.

- For each node, update the maximum number of mount points allowed to include the primary storage pool, the copy storage pool, and the active-data pool:

  ```
  UPDATE NODE NODE2 MAXNUMMP=3
  ```

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.

Results

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.

About this task

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:

Procedure

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).

Results

For more information about storage pool space, see "Estimating space needs for storage pools" on page 375

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 in the hierarchy.

About this task

The migration process can help 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 that is stored in a random-access disk storage pool to a slower, but less expensive sequential-access storage pool.

You can control the following options:

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. It is expressed as a percentage of the 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 that is stored in the pool as a percentage 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 that are specified for the pool. For tape storage pools, the server compares the threshold to the number of volumes that contain data as a percentage of the total number of volumes available to the pool, including scratch volumes.

You can also schedule migration activities to occur at optimal times in your environment. In addition, you can specify the following options:

- How long migration can run before it is automatically canceled
- Whether the server attempts reclamation before migration
- 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 stored in a storage pool, or how long since the file was accessed before it chooses files to migrate. There are optional parameters available for you to change the default behavior of migration. To ensure that files remain in a storage pool for a minimum number of days before the server migrates them to another pool, you can set a migration delay period for the storage pool. 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 that are associated with a FILE device class, all files on a volume must exceed migration delay period before the server can migrate 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. You can use multiple concurrent processes to make better use of your available tape drives, and FILE volumes. Since you can run 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 327.

Migration processing can differ for disk storage pools versus sequential-access storage pools. If you plan to modify the default migration settings for storage pools or want to understand how migration works, read the following topics:

- "Migrating disk storage pools" on page 269
- "Migrating sequential-access storage pools" on page 275
- "Starting migration manually or in a schedule" on page 278

Remember:

- Data cannot be migrated in to or out of storage pools that are defined with a CENTERA device class.

- If you receive an error message during the migration process, refer to *IBM Tivoli Storage Manager Messages*, which can provide useful information to diagnose and fix problems.
- If a migration process is started from a storage pool that does not have the next storage pool identified in the hierarchy, a reclamation process is triggered for the source storage pool. To prevent the reclamation process, define the next storage pool in the hierarchy. For details, see "Setting up a storage pool hierarchy" on page 257. As an alternative to prevent automatic migration from running, set the **HIGHMIG** parameter of the storage pool definition to 100.

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.

About this task

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 274.

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" and "Migration thresholds" on page 272. For information about using the cache, see "Minimizing access time to migrated files" on page 275 and "Caching in disk storage pools" on page 280.

How the server selects files to migrate

When the amount of data in a storage pool is equal to the high migration threshold, the server must migrate files to the next storage pool. The process for selecting files to migrate is based on the space that is used by a client node's files and the migration delay setting.

The server selects the files to migrate as follows:

1. The server checks for the client node that backed up or migrated the largest single file space or archived files that occupy the most space.
2. The next pool collocation setting determines how data is processed. If the collocation setting is GROUP, all the filespaces for all nodes in the collocation group are processed together. If the setting is NODE, all the filespaces for that node are processed together. If the collocation setting is NO or FILESPACE, each filespace and data type (backup, archive, or migrate) is processed separately.
3. For *all* files from *every* filespace belonging to the client node that was identified, the server examines the following statistics:
   - The number of days since the files were stored in the storage pool
   - The number of days since the files were 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.

The server checks the low migration threshold for the storage pool:

- After the server migrates the files for the node to collocate by GROUP or NODE, or
- After the server migrates the processing group to the next 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 continues with that group to collocate by group, if there are nodes that are left in the group. The server can also choose another group by using criteria that has already been described, and the migration process continues.

The server might not be able to reach the low migration threshold by migrating only files that were stored longer than the migration delay period. If so, the server checks the storage pool characteristic that determines whether migration stops, even if the pool is still above the low migration threshold. For more information, see "Keeping files in a storage pool" on page 274.

If multiple migration processes are running, the server might choose the files from more than one node for migration at the same time. The processes are controlled by the **MIGPROCESS** parameter of the **DEFINE STGPOOL** command.

For example, Table 23 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. When the setting is zero, any files can be migrated. Regardless of the amount of time they are stored in the pool or the last time of access, all files are migrated.

Table 23. Database information about 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	

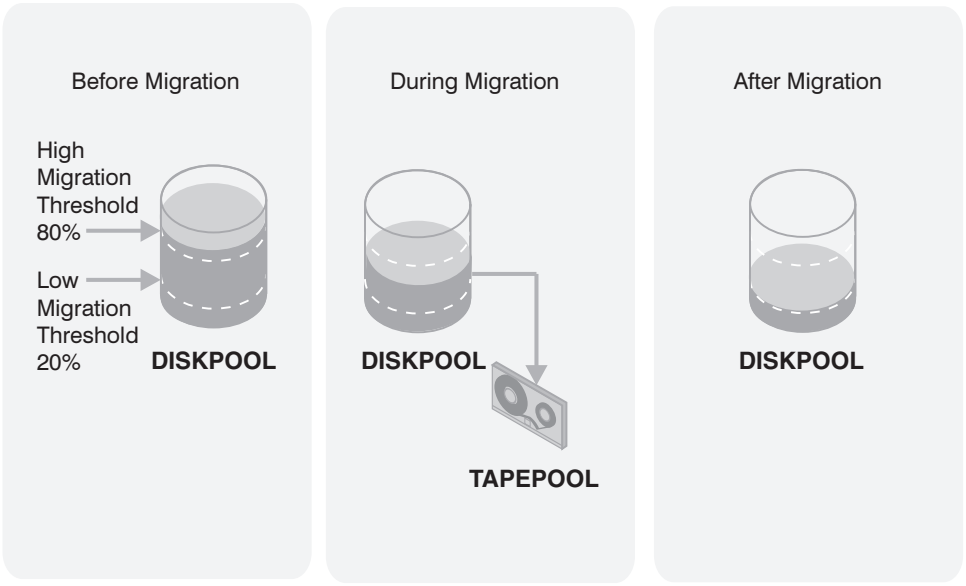


Figure 21. The migration process and migration thresholds

Figure 21 shows what happens when the following occur:

- The high migration threshold that is defined for the disk storage pool DISKPOOL is exceeded
- The TAPEPOOL COLLOCATE parameter is set at the default GROUP or NODE

When the amount of data that can be migrated in DISKPOOL reaches 80%, the server completes the following tasks:

1. Determines that the TOMC/C file space backup data is taking up the most space in the DISKPOOL storage pool. This space is more than any other single backed-up or space-managed or archived file space and more than any client node's archived files.
2. Locates all data that belongs to node TOMC stored in DISKPOOL. In this example, node TOMC backed up or archived files from file spaces TOMC/C and TOMC/D that are 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 that belong to node TOMC. The migration happens, even if the occupancy of the storage pool drops below the low migration threshold before the second file space is 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 280.
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 is not reached, the server determines which client node backed up or migrated the largest single file space or archived files that occupy the most space. The server begins migrating the files that belong 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 is reached or passed, migration ends.

If the TAPEPOOL COLLOCATE parameter is set to NO or FILESPACE, the server completes the following tasks:

1. Determines that the TOMC/C file space backup data is taking up the most space in the DISKPOOL storage pool. This space is more than any other single backed-up or space-managed file space and more than any client node's archived files.
2. Locates all backup data that belongs to node TOMC/C that is stored in the DISKPOOL storage pool.
3. Migrates all backup data from TOMC/C to the next available storage pool. In this example, the data migrates to the tape storage pool, TAPEPOOL.

   The server migrates all of the backup data from filespace C belonging to node TOMC. The migration happens, even if the occupancy of the storage pool drops below the low migration threshold of all the data that was migrated.

   If the cache option is enabled, files that are migrated remain on disk storage (files are cached) until space is needed for new files. For more information about using cache, see "Caching in disk storage pools" on page 280.
4. After all backup files that belong to TOMC/C are migrated to the next storage pool, the server checks the low migration threshold. If the low migration threshold is not reached, the server determines which filespace backed up, migrated, or archived files that occupy the most space. The server begins migrating files that belong to that filespace/type.

   In this example, the server migrates all archive files that belong to the client node named PEASE filespace /temp to the TAPEPOOL storage pool.
5. After all the archive files that belong to PEASE/temp are migrated to the next storage pool, the server checks the low migration threshold again. If the low migration threshold was reached or passed, 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 must consider the amount of free disk storage space needed for normal daily processing.

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 might need to empty the disk (a smaller low threshold).

  If your disk space is limited, try setting the threshold so that migration frees up enough space for the pool to manage the amount of client data that is 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 of your choosing.

  You might also want to identify clients that are daily, transferring large amounts of data. For these clients, you might want to set up a policy (a new copy group or a new policy domain) so that their data is stored directly to tape. Using a separate policy like this can optimize disk use for most 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 might 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 might 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. When you collocate by GROUP, all the file spaces for all nodes in the collocation group are processed together. When you collocate by NODE, all file spaces for that node are processed as a group. When collocation is NO or FILESPACE, each filespace and data type is processed separately, providing greater granularity.

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.

About this task

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

About this task

Important: For information about the disadvantages of using cache, see "Caching in disk storage pools" on page 280.

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.

About this task

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 396.

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 278.) 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 278.

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. The server checks the migration delay for the storage pool before any files are migrated.

For tape 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 that are specified for the pool.

Tip: When Tivoli Storage Manager calculates the capacity for a sequential-access disk storage pool, it considers the amount of disk space available in the file system. For this reason, ensure that you have enough disk space in the file system to hold all the defined and scratch volumes that are specified for the storage pool. For example, suppose that the capacity of all the scratch volumes that are 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 that is used in the migration threshold is 9 TB, not 10 TB. If the high migration threshold is set to 70%, migration begins 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 completes the following procedure:

1. The server first reclaims volumes that exceed 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 363.)
2. After reclamation processing, the server compares the space that is 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 that is 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.

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 must decide which of the following is of greater importance. Either ensuring that files stay in the storage pool for the amount of time that is specified 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 that is 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 must 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, consider the time that is required to mount tapes into drives and whether collocation is enabled.

When you define migration criteria for sequential-access storage pools, consider:

- The capacity of the volumes in the storage pool
- The time that is required to migrate data to the next storage pool
- The speed of the devices that the storage pool uses
- The time that is 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 241.
- Collocation is set the same in both storage pools. For example, if collocation is set to NODE in the first storage pool, then set collocation to NODE in the next storage pool. If collocation is set to FILESPACE in the first storage pool, then set collocation to FILESPACE in the next storage pool.

  When you enable collocation for a storage pool, the server tries to keep all files on a minimal number of volumes. The files can belong to a single client node, a group of client nodes, a client file space, or a group of file spaces. For information about collocation for sequential-access storage pools, see "Keeping client files together using collocation" on page 353.
- 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" on page 278.) 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 a high, high-migration threshold, such as 95%.

For information about setting a reclamation threshold for tape storage pools, see "Reclaiming space in sequential-access storage pools" on page 363.

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 402.

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.

About this task

You can specify the maximum number of minutes the migration runs before automatically canceling. 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 ALTPOOL 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*.

Results

Restriction: Data cannot be migrated into or out of storage pools that are defined with a CENTERA device class.

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.

About this task

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 178

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 327.

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.

About this task

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 533 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 237.

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 378.

Deduplicating data

Data deduplication is a method for eliminating redundant data in order to reduce the storage that is required to retain the data. Only one instance of the data is retained in a deduplicated storage pool. Other instances of the same data are replaced with a pointer to the retained instance.

About this task

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 overview

IBM Tivoli Storage Manager provides two options for performing data deduplication; server-side data deduplication and client-side data deduplication. Both methods use the same process to identify redundant data, however the time and location of the deduplication processing is different. In server-side data deduplication, processing takes place exclusively on the server after the data is backed up. In client-side data deduplication, the processing is distributed between the server and the backup-archive client during the backup process.

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.

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. Furthermore, duplicate data can be identified across objects regardless of whether the data deduplication is

performed on the client or the server.

Server-side data deduplication

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

Client-side data deduplication

In client-side data deduplication, the backup-archive client and the server identify and remove duplicate data to save storage space on the server.

Benefits

Client-side data deduplication provides the following advantages:

- It can reduce the amount of data that is sent over the local area network (LAN).
- Extra processing power and time that is required to remove duplicate data on the server is eliminated.
- Space savings occur immediately on the server because the client removed the duplicated data.
- Extra reclamation processing is not required to remove the redundant data from the server.
- It is no longer necessary to identify duplicates on the server.

Client-side data deduplication stores data directly in a deduplicated format. If storage pool backup is used to create secondary copies to a non-deduplicated storage pool, client extents are reassembled into contiguous files. (Extents are parts of a file that are created during the data-deduplication process.) This reassembly can cause storage pool backup processing to take longer when compared to processing data that was not previously deduplicated.

Requirements

When you configure client-side data deduplication, the following requirements must be met:

- The client and server must be at version 6.2.0 or later.
- Client-side deduplication cannot be used in combination with LAN-free backups.
- The primary storage pool must be a sequential-acces 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 Tivoli Storage Manager client GUI, or in the client option set on the Tivoli Storage Manager server.
- Client-side data deduplication must be enabled on the server by using the **DEDUPLICATION** parameter on the **REGISTER NODE** or **UPDATE NODE** server command.
- Files that are intended for deduplication must not be excluded.
- Files that are intended for deduplication must not be encrypted. Encrypted files and files from encrypted file systems cannot be deduplicated.

Configuration options for client-side deduplication

To take advantage of the client-side data deduplication feature, the following options are available:

- Exclude specific files on a client from data deduplication by using the `exclude.dedup` client option.
- Enable a data deduplication cache, which reduces network traffic between the client and the server. The cache on the client can be enabled through the client options file.

  Specify a size and location for a client cache.

  Restriction: For applications that use the Tivoli Storage Manager API, do not use the data deduplication cache because backup failures might occur when the cache is out of sync with the Tivoli Storage Manager server. If multiple, concurrent Tivoli Storage Manager client sessions are configured, you must configure a separate cache for each session.
- Enable both client-side data deduplication and compression to reduce the amount of data that is stored on the server. Each extent is compressed before it is sent to the server. However, you must balance the benefits of storage savings versus the processing power that is required to compress client data. In general, if you compress and deduplicate data on the client system, you typically use about twice as much processing power as data deduplication alone.

  The server can process compressed data that has been deduplicated. In addition, backup-archive clients earlier than V6.2 can restore deduplicated, compressed data.

Client-side data deduplication and storage pools

If client-side data deduplication is enabled and the primary destination storage pool is full, and another storage pool is 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 can complete processing, maintain sufficient free storage in your primary destination storage pool.

For details about client-side data deduplication, including options for controlling data deduplication, see the *Backup-Archive Clients Installation and User's Guide*.

LAN-free access to storage pools that contain client-side deduplicated data

Only V6.2 and later 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 later can complete operations over the LAN.

Table 24. 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.

Related concepts:

"Server-side data deduplication" on page 282

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
- Server-initiated sessions

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.

The following is an example of how simultaneous operations can create problems when using client-side data deduplication.

When client-side data deduplication is used to back up or archive files, it is possible for a server operation to mark a file extent on the server as expired, before the client operation completes a backup or an archive that involves the same extent. If this happens, the client is notified of the expired extent. The client makes three attempts to copy the extent to the server before it gives up and processes the other files in the backup operation.

A return code (RC=254) and message are written to the dsmerror.log file. The message is also displayed in the command-line client. The error message is:

```
ANS7899E The client referenced a duplicated extent that does not exist
   on the Tivoli Storage Manager server.
```

The workaround for this situation is to ensure that processes that can cause files to expire are not run at the same time that back up or archive operations with client-side data deduplication are performed.

Data deduplication of hierarchical storage management data

HSM data from UNIX and Linux clients is ignored by client-side data deduplication. Server-side data 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.

Moving or copying data from a deduplicated storage pool to a non-deduplicated storage pool

When you copy or move data from a deduplicated storage pool to a non-deduplicated storage pool, the data is reconstructed. However, after the data movement or copy operation, the amount of data that is reported as moved or copied is the amount of deduplicated data. For example, suppose that a storage pool contains 20 GB of deduplicated data that represents 50 GB of total file data. If the data is moved or copied, the server reports that 20 GB was moved or copied, even though 50 GB of data was sent.

When data deduplication is not appropriate

Using Tivoli Storage Manager data deduplication can provide several advantages. However, there are some situations where data deduplication is not appropriate. Those situations are:

- Your primary storage of backup data is on a Virtual Tape Library or physical tape. If regular migration to tape is required, the benefits of using data deduplication are lessened, since the purpose of data deduplication is to reduce disk storage as the primary location of backup data.
- You have no flexibility with the backup processing window. Tivoli Storage Manager data deduplication processing requires additional resources, which can extend backup windows or server processing times for daily backup activities.
- Your restore processing times must be fast. Restore performance from deduplicated storage pools is slower than from a comparable disk storage pool that does not use data deduplication. If fast restore performance from disk is a high priority, restore performance benchmarking must be done to determine whether the effects of data deduplication can be accommodated.

Related tasks:

“Keeping client files together using collocation” on page 353

“Detecting possible security attacks during client-side data deduplication” on page 298

Planning guidelines for data deduplication

Planning for data deduplication is important because there are many factors to consider. These factors include database and log size capacity, data deduplication location, storage pool setup, and additional hardware requirements.

As part of the planning process, ensure that you will benefit from using data deduplication. In the following situations, IBM Tivoli Storage Manager data deduplication can provide a cost-effective method for reducing the amount of disk storage that is required for backups:

- You have to reduce the disk space that is required for backup storage.
- You must perform remote backups over limited bandwidth connections.
- You are using Tivoli Storage Manager node replication for disaster recovery across geographically dispersed locations.
- You either have disk-to-disk backup configured (where the final destination of backup data is on a deduplicating disk storage pool), or data is stored in the FILE storage pool for a significant time (for example 30 days), or until expiration.
- For guidance on the scalability of data deduplication with Tivoli Storage Manager, see *Effective Planning and Use of IBM Tivoli Storage Manager V6 Deduplication* at http://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Tivoli Storage Manager/page/Effective Planning and Use of IBM Tivoli Storage Manager V6 Deduplication.

Related concepts:

"Example: Estimating active and archive log sizes for data deduplication operations" on page 653

Related tasks:

"Detecting possible security attacks during client-side data deduplication" on page 298

Implementing data deduplication

In order to implement data deduplication, you must make several decisions that are based on the outcome that you want to achieve.

About this task

- Determine database capacity requirements. When you use data deduplication, considerably more database space is required as a result of storing the metadata that is related to duplicate data.
- Determine database log size requirements. It is essential that you properly size the storage capacity for the database active log and archive log.
- 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. To help you make that determination, consider the following factors:
  - Server-side data deduplication is a two-step process in which duplicate data is identified and then storage space is reclaimed to remove the duplicate data. Client-side data deduplication stores the data directly in a deduplicated format.
  - Data deduplication and data compression can be combined on the backup-archive client to reduce data storage. This reduction is typically more than you can achieve by using server-side data deduplication alone.
  - If bandwidth is not restrictive, client-side data deduplication processing typically causes an increase in time for backup operations to complete.

Consider doubling the time that you allow for backups when you use client-side data deduplication in an environment that is not limited by the network. If you are creating a secondary copy by using storage pool backup, where the copy storage pool is not using data deduplication, it takes longer for data to be moved because of the extra processing that is required to reconstruct the deduplicated data.

  - Duplicate identification processing is handled by client systems when client-side deduplication is used. However, the Tivoli Storage Manager server still requires processing to handle the lookup requests from clients and to store data deduplication metadata that is produced by client systems.
  - Client-side data deduplication cannot be combined with LAN-free data movement that uses the Tivoli Storage Manager for Storage Area Networks feature. If you are implementing a Tivoli Storage Manager supported LAN-free to disk solution, consider server-side data deduplication.

- If you choose client-side data deduplication, decide what, if any, security precautions to take.
- 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 change the settings later, after the initial setup, to meet the requirements of your operations.

  The following table lists the options that you can use to control duplicate identification processes.

Table 25. Options for controlling duplicate-identification processes

If you create a storage pool for data deduplication...	If you update an existing storage pool...
You can specify 1 - 20 duplicate-identification processes to start automatically. The Tivoli Storage Manager server does not start any processes if you specify zero. 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. After the storage pool is created, you can increase and decrease the number of duplicate-identification processes manually. You can also start, stop, and restart duplicate-identification processes manually.	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. The Tivoli Storage Manager server does not start any duplicate-identification processes automatically by default.

- 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.

- Determine the best time to use data deduplication for the storage pool. The duplicate identification (IDENTIFY) processes can increase the workload on the processor and system memory. Schedule duplicate identification processes at the following times:
  - When the process does not conflict with other processes such as reclamation, migration, and storage pool backup
  - Before node replication (if node replication is being used) so that node replication can be used in combination with deduplication

Related tasks:

"Detecting possible security attacks during client-side data deduplication" on page 298

Checklist for data deduplication

Data deduplication requires additional processing resources on the server or client. Use the checklist to verify that hardware and your Tivoli Storage Manager configuration have characteristics that are key to good performance.

Question	Tasks, characteristics, options, or settings	More information
Are you using fast disk storage for the Tivoli Storage Manager database as measured in terms of input/output operations per second (IOPS)?	Use high-performance disk for the Tivoli Storage Manager database. At a minimum, use 10000-rpm drives for smaller databases that are 200 GB or less. For databases over 500 GB, use 15000-rpm drives or solid-state drives. Tivoli Storage Manager database should have a minimum capability of 3,000 IOPS. For each TB of data that is backed up daily (before data deduplication), include an additional 1,000 IOPS to this minimum. For example, a Tivoli Storage Manager server that is ingesting 3 TB of data per day would need 6,000 IOPS for the database disks: `3,000 IOPS minimum + 3,000 (3 TB x 1,000 IOPS) = 6,000 IOPS`	Checklist for server database disks
Do you have enough memory for the size of your database?	Use a minimum of 64 GB of system memory for Tivoli Storage Manager servers that are deduplicating data. If the retained capacity of backup data grows, the memory requirement might need to be as high as 128 GB. Monitor memory usage regularly to determine whether more memory is required.	

Question	Tasks, characteristics, options, or settings	More information
Have you properly sized your disk space for the database, logs, and storage pools?	For a rough estimate, plan for 150 GB of database storage for every 10 TB of data that is to be protected in deduplicated storage pools. *Protected data* is the amount of data before deduplication, including all versions of objects stored. Configure the server to have the maximum active log size of 128 GB by setting the **ACTIVELOGSIZE** server option to a value of 131072. Use a directory for the database archive logs with an initial free capacity of at least 500 GB. Specify the directory by using the **ARCHLOGDIRECTORY** server option. Define space for the archive failover log by using the **ARCHFAILOVERLOGDIRECTORY** server option.	
Are the Tivoli Storage Manager database and logs on separate disk volumes (LUNs)? Is the disk that is used for the database configured according to best practices for a transactional database?	The Tivoli Storage Manager database must not share disk volumes with Tivoli Storage Manager database logs or storage pools, or with any other application or file system.	See Server database and recovery log configuration and tuning
Are you using a minimum of 8 (2.2 GHz or equivalent) processor cores for each Tivoli Storage Manager server that you plan to use with data deduplication?	If you are planning to use client-side data deduplication, verify that client systems have adequate resources available during a backup operation to perform data deduplication processing. Use a processor that is at least the minimum equivalent of one 2.2 GHz processor core per backup process with client-side data deduplication.	

Question	Tasks, characteristics, options, or settings	More information
Have you estimated storage pool capacity to configure enough space for the size of your environment?	You can estimate storage pool capacity requirements for a deduplicated storage pool by using the following technique: 1. Estimate the base size of the source data. 2. Estimate the daily backup size by using an estimated change and growth rate. 3. Determine retention requirements. 4. Estimate the total amount of source data by factoring in the base size, daily backup size, and retention requirements. 5. Apply the deduplication ratio factor. 6. Round up the estimate to consider transient storage pool usage.	For an example of using this technique, see *Effective Planning and Use of IBM Tivoli Storage Manager V6 Deduplication* at http://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Tivoli Storage Manager/page/Effective Planning and Use of IBM Tivoli Storage Manager V6 Deduplication.
Have you distributed disk I/O over many disk devices and controllers?	Use arrays that consist of as many disks as possible, which is sometimes referred to as wide striping. Specify 8 or more file systems for the deduplicated storage pool device class so that I/O is distributed across as many LUNs and physical devices as possible.	See Checklist for storage pools on disk.
Do you have adequate resources to cover client-side data deduplication requirements?	If you are planning to use client-side data deduplication, verify that client systems have adequate resources available during a backup operation to perform data deduplication processing. Use a processor that is at least the minimum equivalent of one 2.2 GHz processor core per backup process with client-side data deduplication.	

Best practices for data deduplication

Implement data deduplication in IBM Tivoli Storage Manager by using the best practices to ensure that the benefits of data deduplication are optimized on your system.

When you use data deduplication, your system can achieve benefits such as these:

- Reduction in the storage capacity that is required for storage pools on the server that are associated with a FILE-type device class. This reduction applies for both server-side and client-side data deduplication.
- Reduction in the network traffic between the client and server. This reduction applies for client-side deduplication only.

Using data deduplication requires an increase in resources on the Tivoli Storage Manager server and clients. The upper limit on the size of objects that are stored in

deduplicated storage pools is one primary consideration. However, in general, a server that uses data deduplication must be allocated additional resources.

When you implement the suggested practices for data deduplication, you can help to avoid problems such as these on your system:

- Server outages that are caused by running out of active log space or archive log space
- Server outages or client backup failures that are caused by exceeding the IBM DB2 internal lock list limit
- Process failures and hangs that are caused during server data management

Properly size the server database, recovery log, and system memory:

When you use data deduplication, considerably more database space is required as a result of storing the metadata that is related to duplicate data. Data deduplication also tends to cause longer-running transactions and a related larger peak in recovery log usage.

In addition, more system memory is required for caching database pages that are used during duplicate data lookup for both server-side and client-side data deduplication.

Tips:

- Ensure that the Tivoli Storage Manager server has a minimum of 64 GB of system memory.
- Allocate a file system with two-to-three times more capacity for the server database than you would allocate for a server that does not use data deduplication. You can plan for 150 GB of database storage for every 10 TB of data that is protected in the deduplicated storage pools.
- Configure the server to have the maximum active log size of 128 GB by setting the `ACTIVELOGSIZE` server option to a value of 131072.
- Use a directory for the database archive logs with an initial free capacity of at least 500 GB. Specify the directory by using the `ARCHLOGDIRECTORY` server option.

For more information about managing resources such as the database and recovery log, see the *Installation Guide*. Search for database and recovery log capacity planning.

Avoid the overlap of server maintenance tasks with client backup windows:

When you schedule client backups for a period during which server maintenance tasks are not running, you create a *backup window*. This practice is important when you use data deduplication. Use this process regardless of whether data deduplication is used with Tivoli Storage Manager.

Perform server maintenance tasks in a sequence that avoids contention between the different types of processing. The server maintenance tasks are:

- Migration
- Storage pool backup
- Duplicate identification
- Database backup
- Node replication
- Reclamation

- Expiration

Migration and reclamation are the tasks most likely to interfere with the success of client backups.

Tips:
- Schedule client backups in a backup window that is isolated from data maintenance processes, such as migration and reclamation.
- Schedule each type of data maintenance task with controlled start times and durations so that they do not overlap with each other.
- If storage-pool backup is used to create a secondary copy, schedule storage-pool backup operations before you start data deduplication processing to avoid restoring objects that are sent to a non-deduplicated copy storage pool.
- If you are using node replication to keep a secondary copy of your data, schedule the **REPLICATE NODE** command to run after duplicate identification processes are completed.

For more information about tuning the schedule for daily server maintenance tasks, see the *Optimizing Performance* guide. Search for tuning the schedule for daily operations.

Modify DB2 lock list management:

The lock list storage of DB2 that is automatically managed can become insufficient. If you deduplicate data that includes large files or large numbers of files concurrently, the data deduplication can cause insufficient storage. When the lock list storage is insufficient, backup failures, data management process failures, or server outages can occur.

File sizes greater than 500 GB that are processed by data deduplication are most likely to cause storage to become insufficient. However, if many backups use client-side data deduplication, this problem can also occur with smaller-sized files.

The following activities can use more lock list storage:
- Client backups that use client-side data deduplication
- Data movement within a deduplicated storage pool that uses the reclamation and **MOVE DATA** commands
- Data movement out of a deduplicated storage pool that uses the migration and **MOVE DATA** commands

See Technote 1430874 at http://www.ibm.com/support/docview.wss?uid=swg21430874 for an explanation about how to estimate the peak volume of deduplication transactions that are processed. This document also contains information about the corresponding lock list requirements for managing the volume and information about how to change the DB2 limit, if necessary.

Tip: When you estimate the lock list storage requirements, follow the information described in the technote to manage storage for loads that are much larger than expected.

Limit the impact of large objects on data deduplication processing:

You can use controls to limit the potential effect of large objects on data deduplication processing on the Tivoli Storage Manager server.

You can use the following controls when you deduplicate large-object data:

- Server controls that limit the size of objects. These controls limit the size of objects that are processed by data deduplication.
- Controls on the data management processes of the server. These controls limit the number of processes that can operate concurrently on the server.
- Scheduling options that control how many clients run scheduled backups simultaneously. These scheduling options can be used to limit the number of clients that perform client-side data deduplication at the same time.
- Client controls whereby larger objects can be processed as a collection of smaller objects. These controls are primarily related to the Tivoli Storage Manager data protection products.

Server controls to limit the data deduplication of large objects:

Use the server controls that are available on Tivoli Storage Manager server to prevent large objects from being processed by data deduplication.

Use the following parameter and server options to limit the object size for data deduplication:

MAXSIZE

For storage pools, the **MAXSIZE** parameter can be used to prevent large objects from being stored in a deduplicated storage pool. Use the default `NOLIMIT` parameter value, or set the value to be greater than `CLIENTDEDUPTXNLIMIT` and `SERVERDEDUPTXNLIMIT` option values.

Use the **MAXSIZE** parameter with a deduplicated storage pool to prevent objects that are too large to be eligible for data deduplication from being stored in a deduplicated storage pool. The objects are then redirected to the next storage pool in the storage pool hierarchy.

SERVERDEDUPTXNLIMIT

The `SERVERDEDUPTXNLIMIT` server option limits the total size of objects that can be deduplicated in a single transaction by duplicate identification processes. This option limits the maximum file size that is processed by server-side data deduplication. The default value for this option is 300 GB, and the maximum value is 2048 GB. Because less simultaneous activity is typical with server-side data deduplication, consider having a limit larger than 300 GB on the object size for server-side data deduplication.

CLIENTDEDUPTXNLIMIT

The `CLIENTDEDUPTXNLIMIT` server option restricts the total size of all objects that can be deduplicated in a single client transaction. This option limits the maximum object size that is processed by client-side data deduplication. However, there are some methods to break up larger objects. The default value for this option is 300 GB, and the maximum value is 1024 GB.

Tips:

- Set the **MAXSIZE** parameter for deduplicated storage pools to a value slightly greater than `CLIENTDEDUPTXNLIMIT` and `SERVERDEDUPTXNLIMIT` option values.

- If you increase CLIENTDEDUPTXNLIMIT or SERVERDEDUPTXNLIMIT option values beyond the defaults, reevaluate your sizing or the server recovery log. Also, reevaluate your sizing for the IBM DB2 lock list.
- If you plan to run many simultaneous client backups that use client-side data deduplication, consider lowering the setting of the CLIENTDEDUPTXNLIMIT option to the minimum setting of 32 GB. Lower the setting if large objects will not be deduplicated.

Controls for data management processes:

Use the controls for the data management processes of the Tivoli Storage Manager server. These controls limit the number of large objects that are simultaneously processed by the server during data deduplication.

Use the following commands and parameters to limit the number of large objects that are simultaneously processed by the server:

- The storage pool parameters on the **DEFINE STGPOOL** command or the **UPDATE STGPOOL** command.
  - The **MIGPROCESS** parameter controls the number of migration processes for a specific storage pool.
  - The **RECLAIMPROCESS** parameter controls the number of simultaneous processes that are used for reclamation.
- The **IDENTIFYPROCESS** parameter on the **IDENTIFY DUPLICATES** command. The parameter controls the number of duplicate identification processes that can run at one time for a specific storage pool.

Tips:

- You can safely run duplicate identification processes for more than one deduplicated storage pool at the same time. However, specify the **IDENTIFYPROCESS** parameter with the **IDENTIFY DUPLICATES** command to limit the total number of all simultaneous duplicate identification processes. Limit the total number to a number less than or equal to the number of processors that are available in the system.
- Schedule duplicate identification processes to run when the additional load does not affect client operations or conflict with other server processes. For example, schedule the duplicate identification process to run outside the client backup window. The duplicate identification processes for the server intensively use the database and system resources. These processes place additional processing on the processor and memory of the system.
- Do not overlap different types of operations, such as expiration, reclamation, migration, and storage pool backup.
- Read the information about data deduplication and the server storage pool. The effect of data deduplication on system resources is also related to the size of the file for deduplication. As the size of the file increases, more processing time, processor resources, memory, and active log space are needed on the server.

  Review the document for information about data deduplication and the server storage pool.

Schedule controls for client backups:

For scheduled backups, you can limit the number of client backup sessions that perform client-side data deduplication at the same time.

You can use any of the following approaches to limit the number of client backup sessions:

- Clients can be clustered in groups by using different schedule definitions that run at different times during the backup window. Consider spreading clients that use client-side deduplication among these different groups.
- Increase the duration for scheduled startup windows and increase the randomization of schedule start times. This limits the number of backups that use client-side data deduplication that start at the same time.
- Separate client backup destinations by using the server policy definitions of the Tivoli Storage Manager server, so that different groups of clients use different storage pool destinations:
  - Clients for which data is never to be deduplicated cannot use a management class that has as its destination a storage pool with data deduplication enabled.
  - Clients that use client-side data deduplication can use storage pools where they are matched with other clients for which there is a higher likelihood of duplicate matches. For example, all clients that run Microsoft Windows operating systems can be set up to use a common storage pool. However, they do not necessarily benefit from sharing a storage pool with clients that perform backups of Oracle databases.

Client controls to limit the data deduplication of large objects:

A primary source of large objects that are processed by client-side data deduplication are backups by Tivoli Storage Manager data protection products. You can use controls to have the clients break these objects into multiple smaller objects that are within the object size limits for data deduplication.

Many of the data protection products process objects with sizes in the range of several hundred GBs to one TB. This range exceeds the maximum object size that is acceptable for data deduplication.

You can reduce large objects into multiple smaller objects by using the following methods:

- Use Tivoli Storage Manager client features that back up application data with the use of multiple streams. For example, a 1 TB database is not eligible for data deduplication as a whole. However, when backed up with four parallel streams, the resulting four 250 GB objects are eligible for deduplication. For Tivoli Storage Manager Data Protection for SQL, you can specify a number of stripes to change the backup into multiple streams.
- Use application controls that influence the maximum object size that is passed through to Tivoli Storage Manager. Tivoli Storage Manager Data Protection for Oracle has several RMAN configuration parameters that can cause larger databases to be broken into smaller objects. These configuration parameters include the use of multiple channels, or the `MAXPIECESIZE` option, or both.

Restriction: In some cases, large objects cannot be reduced in size, and therefore cannot be processed by Tivoli Storage Manager data deduplication:

- The backup-archive clients always send large files in a single transaction, which cannot be broken into smaller transactions.
- Image backups of a large file system are sent within a single transaction and cannot be broken into smaller components.

Performance considerations for data deduplication

Finding duplicate data is a processor-intensive process. When you use client-side data deduplication, the processor consumption occurs on the client system during the backup. With server-side data deduplication, the processor consumption occurs on the server during the duplicate identification process. Consider factors such as processor usage, network bandwidth, restore performance, and compression when you decide to use data deduplication.

Processor usage

The amount of processor resources that are used depends on how many client sessions or server processes are simultaneously active. Additionally, the amount of processor usage is increased because of other factors, such as the size of the files that are backed up. When I/O bandwidth is available and the files are large, for example 1 MB, finding duplicates can use an entire processor during a session or process. When files are smaller, other bottlenecks can occur. These bottlenecks can include reading files from the client disk or the updating of the Tivoli Storage Manager server database. In these bottleneck situations, data deduplication might not use all of the resources of the processor.

You can control processor resources by limiting or increasing the number of client sessions for a client or a server duplicate identification processes. To take advantage of your processor and to complete data deduplication faster, you can increase the number of identification processes or client sessions for the client. The increase can be up to the number of processors that are on the system. It can be more than that number if the processors support multiple hardware-assisted threads for the core, such as with simultaneous multithreading. Consider a minimum of at least 8 (2.2Ghz or equivalent) processor cores in any Tivoli Storage Manager server that is configured for data deduplication.

Client-side data deduplication can use a large amount of processor resources. Therefore, verify that the additional workload does not affect the primary workload of the client system.

Compressing the data, in addition to deduplicating it on the client, uses additional processor resources. However, it lowers the network bandwidth that is required if the data is compressible.

Network bandwidth

A primary reason to use client-side data deduplication is to reduce the bandwidth that is required to transfer data to a Tivoli Storage Manager server. Client compression can reduce this bandwidth further. The amount that the bandwidth is reduced by is directly related to how much of the data is duplicate that is already stored on the server. It is also directly related to how compressible this data is.

Network bandwidth for the queries for data from the Tivoli Storage Manager client to the server can be reduced by using the `enablededupcache` client option. The cache stores information about extents that have been previously sent to the server. If an extent is found that was previously sent, it is not necessary to query the

server again for that extent. Therefore, bandwidth and performance are not additionally reduced.

Restore performance

During a restore operation, performance for a deduplicated storage pool can be slower than a restore from a non-deduplicated pool. When data deduplication is used, the extents for a given file can be spread across multiple volumes on the server. This spreading of extents makes the reads from the volumes more random in nature, and also slower than during a sequential operation. In addition, more database operations are required.

Compression

Data deduplication is not performed on directories or file metadata. In comparison, compression can be performed on these types of data. Therefore, the reduction percentages do not typically add up to the total data-reduction percentage. When client-side data deduplication is used, the calculation of the compression-reduction percentage is performed differently. The calculation includes only the actual data reduction that results from use of the compression engine. Client-side data deduplication, when performed, occurs before compression.

For the quickest backups on an unconstrained network, choose server-side data deduplication. For the largest storage savings, choose client-side data deduplication combined with compression. Avoid performing client-compression in combination with server-side data deduplication.

For additional information about data deduplication and performance considerations, see the *Optimizing Performance* guide.

Detecting possible security attacks during client-side data deduplication

A rogue application that is part of 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.

About this task

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.

Procedure

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.

What to do next

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.
- Using storage-device encryption together with client-side data deduplication
- Enabling Secure Sockets Layer (SSL)
- Deduplicating data that uses only server-side data deduplication. Server-side data deduplication does not expose the server to security attacks from the client.

To show 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.

Tip: 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.

Estimating space savings from server-side data deduplication

Before you set 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.

Procedure

To estimate space savings, complete the following steps:

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.

Results

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.

Verifying your implementation

After you configured data deduplication, there are some actions that you can take to verify whether your solution is processing effectively.

About this task

Use the following questions to evaluate your data deduplication configuration.

Have you scheduled data deduplication processing based on your backup strategy?

If you are not creating a secondary copy of backup data or if you are using node replication for the second copy, client backup and duplicate identification can be overlapped. This can reduce the total elapsed time for these operations, but might increase the time that is required for client backup.

If you are using storage pool backup, do not overlap client backup and duplicate identification. The best practice sequence of operations is client backup, storage pool backup, and then duplicate identification.

For data that is not stored with client-side data deduplication, schedule storage-pool backup operations to complete before you start data deduplication processing. Set up your schedule this way to avoid reconstructing objects that are deduplicated to make a non-deduplicated copy to a different storage pool.

See Scheduling data deduplication and node replication processes for more information.

Are the processes for identifying duplicates able to handle all new data that is backed up each day?

If the process completes, or goes into an idle state before the next scheduled operation begins, then all new data is being processed.

Is reclamation able to run to a sufficiently low threshold?

If a low threshold cannot be reached, consider the following actions:

- Increase the number of processes that are used for reclamation.
- Upgrade to faster hardware.

Is deduplication cleanup processing able to clean out the dereferenced extents to free disk space before the start of the next backup cycle?

Run the **SHOW DEDUPDELETE** command. The output shows that all threads are idle when the workload is complete.

If cleanup processing cannot complete, consider the following actions:

- Increase the number of processes that are used for duplicate identification.
- Upgrade to faster hardware.
- Determine if you are ingesting more data that your Tivoli Storage Manager server can process with data deduplication and consider deploying an additional Tivoli Storage Manager server.

Related reference:

"Checklist for data deduplication" on page 289

Managing deduplication-enabled storage pools

You can create a storage pool for data deduplication or you can update an existing storage pool. If you are implementing server-side data deduplication, IBM Tivoli Storage Manager provides the option of running duplicate-identification processes automatically or manually.

Before you begin

Before you set 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 is already 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.

About this task

You can create a storage pool for data deduplication or update an existing storage pool for data deduplication. You can store client-side deduplicated data and server-side deduplicated data in the same storage pool.

Procedure

To set up a storage pool for data deduplication, complete the following steps:

- If you are defining a new storage pool:
  1. Use the **DEFINE STGPOOL** 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 that belongs to the excluded nodes from the storage pool to be converted to another storage pool.
     b. Direct data that belongs 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 with the **UPDATE STGPOOL** command. Specify the **DEDUPLICATE** and **NUMPROCESSES** parameters.

Results

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 50 duplicate-identification processes to start automatically. However, the number of duplicate-identification processes must not exceed the number of processor cores available on the Tivoli Storage Manager server. 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, along with 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.

Creating a copy of primary storage pools with data deduplication enabled

Tivoli Storage Manager provides two methods that you can use to create a copy of data in your deduplicated storage pools.

About this task

You can create a copy of the data by using **BACKUP STGPOOL** or **REPLICATE NODE** command. When you back up a primary storage pool, you create a copy of the entire storage pool. When you replicate data by using node replication, you copy data from one or more nodes from primary storage pools to a primary storage pool on another Tivoli Storage Manager server.

For deduplicated storage pools, you must determine whether the DEDUPREQUIRESBACKUP should be set to yes or no.

The value of the DEDUPREQUIRESBACKUP option specifies whether the following actions will occur:

- Volumes in primary sequential-access storage pools that are set up for data deduplication can be reclaimed
- Duplicate data can be discarded before the storage pools are backed up

The default value is Yes.

Important: If you do not want to create a copy of the storage pool data, you must change the DEDUPREQUIRESBACKUP option to No. Failure to set the option correctly can cause deduplication to not work correctly and cause your data to not be reclaimed.

Table 26 describes the different scenarios that you can use to create a copy of data in your deduplicated storage pools, and which value of DEDUPREQUIRESBACKUP to use.

Table 26. Setting the value for the DEDUPREQUIRESBACKUP option

Creating a copy of your primary storage pool data	DEDUPREQUIRESBACKUP value	Method
Back up your primary storage pool data to a non-deduplicated copy pool, such as a copy pool that uses tape.	Yes	**BACKUP STGPOOL**
Back up your primary storage pool data to a deduplicated copy pool.	No	**BACKUP STGPOOL**

Table 26. Setting the value for the DEDUPREQUIRESBACKUP option (continued)

Creating a copy of your primary storage pool data	DEDUPREQUIRESBACKUP value	Method
Use node replication to create a copy of your data on another Tivoli Storage Manager server.	No	**REPLICATE NODE**
No copy is created.	No	

Depending on the method that you chose to create a copy of the data in the primary storage pools, complete one of the following actions:

Procedure

- Use the storage pool backup command to back up data:
  1. Issue the **BACKUP STGPOOL** command. If you set the DEDUPREQUIRESBACKUP option to yes, you must back up data to a copy storage pool that is not set up for data deduplication.

     Tip: When you copy data to an active data pool, it does not provide the same level of protection that occurs when you create a storage pool backup or use node replication.
  2. Issue the **IDENTIFY DUPLICATES** command to identify duplicate data.

     Tip: If you backup storage pool data after duplicate data is identified, the copy process can take longer because the data must be reconstructed to find any duplicate data.
- Use the node replication command to back up data:
  1. Issue the **IDENTIFY DUPLICATES** command to identify duplicate data.
  2. Issue the **REPLICATE NODE** command to start node replication.

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 27. 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.

Table 27. Effects when moving or copying data (continued)

If the source storage pool is...	...and you move or copy data to a target storage pool that is...	The result is...
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.

When you copy or move data from a deduplicated storage pool to a non-deduplicated storage pool, the data is reconstructed. However, after the data movement or copy operation, the amount of data that is reported as moved or copied is the amount of deduplicated data. For example, suppose a storage pool contains 20 GB of deduplicated data that represents 50 GB of total file data. If the data is moved or copied, the server reports that 20 GB was moved or copied, even though 50 GB of data was sent.

Turning data deduplication on or 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.

About this task

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.

Procedure

To turn off data deduplication for a storage pool, use the **UPDATE STGPOOL** command and specify `DEDUPLICATE=NO`.

What to do next

If you turn data deduplication on for the same storage pool, duplicate-identification processes resume, skipping any files that are previously processed. You can change the number of duplicate-identification processes. When you are calculating the number of duplicate-identification processes to specify, consider the workload on the server and the amount of data that requires data deduplication. The number of duplicate-identification processes must not exceed the number of processor cores available on the IBM Tivoli Storage Manager server.

To enable data deduplication for a storage pool, specify `DEDUPLICATE=YES` on the **UPDATE STGPOOL** command.

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 by using a combination of settings on the client and the server.

About this task

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 28. 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.

Procedure

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" on page 282

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.

About this task

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.

Remember: The number of duplicate-identification processes should not exceed the number of processor cores available on the Tivoli Storage Manager server.

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 29 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 29. 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.

Table 29. Controlling duplicate-identification processes manually (continued)

Using the IDENTIFY DUPLICATES command, you specify...	**...and a duration of...**	**The result is...**
0 duplicate-identification processes	None specified	All duplicate-identification processes finish the files that they are working on, if any, and stop. This change lasts indefinitely, or until you reissue the **IDENTIFY DUPLICATES** command, update the storage pool definition, or cancel a process.
	60 minutes	All duplicate-identification processes finish the files that they are working on, if any, and stop. At the end of 60 minutes, the server starts three processes.
None specified	Not available	The number of duplicate-identification processes resets to the number of processes specified in the storage pool definition. This change lasts indefinitely, or until you reissue the **IDENTIFY DUPLICATES** command, update the storage pool definition, or cancel a process.

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.

Procedure

To specify the number and duration of duplicate-identification processes for a storage pool, issue the **IDENTIFY DUPLICATES** command.

Example

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.

Enabling deduplication of NetApp file-server data:

To deduplicate data that is stored by a network-attached storage (NAS) file server, use the ENABLENASDEDUP server option. This option applies only to NetApp file servers. The default is NO. Deduplication for the storage pool must be enabled. If you are using NDMP backup of NetApp file servers in your environment, use the YES option.

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.

Procedure

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.

Results

For details about client-side data deduplication options, see the *Backup-Archive Clients Installation and User's Guide*.

Related concepts:

"Client-side data deduplication" on page 282

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.

About this task

The name of the domain that you use to manage deduplicated data is dedupdomain1. The primary storage pool that is 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.

Procedure

To enable client-side data deduplication for a single node, complete the following steps:

- 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.

What to do next

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.

Procedure

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 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.

About this task

The data belonging client MATT is bound to a management class with a copy group that specifies a deduplication-enabled destination storage pool.

Procedure

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.

About this task

High levels of database activity can produce the following symptoms:

- Reduced throughput for client backup and archive operations
- Resource contention that results from concurrent server operations
- Excessive recovery log activity

The extent to which these symptoms occur depends on the number and size of the objects that are being processed, the intensity, and the type of concurrent operations that are taking place on the IBM Tivoli Storage Manager server, and the Tivoli Storage Manager server configuration.

Procedure

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.

Results

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.

What to do next

For details about these options, see the *Administrator's Reference*.

Tiered data deduplication

With tiered data deduplication, you can use larger average extent sizes as objects reach a greater size.

Data deduplication uses an average extent size of 256 KB. When deduplicating large objects, for example, over 200 GB, the number of extents for an object can grow large. Assuming extents are 256 KB, there are 819,200 extents for a 200 GB object. When you need to restore this object, all 819,200 database records must be read before the object is accessible.

Tiered data deduplication can manage larger objects because a larger average extent size is used when deduplicating the data. For example, after an object reaches 200 GB, the Tivoli Storage Manager server uses 1 MB as the average extent size, instead of 256 KB. 819,200 extents become 204,800 extents.

Using the server options, DEDUPTIER2FILESIZE and DEDUPTIER3FILESIZE, you can configure three different tier sizes for data deduplication:

Tier 1 All objects greater than 0 and less than the value set for the DEDUPTIER2FILESIZE option. Tier 1 uses an average extent size of 256 KB.

Tier 2 All objects greater than or equal to the value of DEDUPTIER2FILESIZE and less than the value for DEDUPTIER3FILESIZE. Tier 2 uses an average extent size of 1 MB.

Tier 3 All objects greater than or equal to the value for DEDUPTIER3FILESIZE. Tier 3 uses an average extent size of 2 MB.

Note: By default, objects under 100 GB in size are processed at Tier 1. Objects in the range of 100 GB to under 400 GB are processed in Tier 2. All objects 400 GB and larger are processed in Tier 3.

Setting options for tiered deduplication:

Depending on your environment, you can set different options for using tiered data deduplication. However, if possible, avoid changing the default tier settings. Small changes might be tolerated, but frequent changes to these settings can prevent matches between previously stored backups and future backups.

Example

If you want to use two tiers for data deduplication instead of three, you can set the DEDUPTIER2FILESIZE and DEDUPTIER3FILESIZE accordingly.

Use Tier 1 and Tier 2 only
To have two tiers with an average extent size of 256 KB and 1 MB, specify these values:

```
DEDUPTIER2FILESIZE 100
DEDUPTIER3FILESIZE 9999
```

Use Tier 1 and Tier 3 only
To have two tiers with an average extent size of 256 KB and 2 MB, specify these values:

```
DEDUPTIER2FILESIZE 100
DEDUPTIER3FILESIZE 100
```

If you do not want to use tiered data deduplication and instead preserve your existing environment, set the value for both of the tiered data deduplication options to 9999. For example:

```
DEDUPTIER2FILESIZE 9999
DEDUPTIER3FILESIZE 9999
```

If both options are set to 9999, then all files that are 10 TB or less are processed with the default extent size of 256 KB.

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.

About this task

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.

About this task

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

Procedure

To query a storage pool for statistics about data deduplication, issue the **QUERY STGPOOL** command.

What to do next

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.

If you run a query before reclamation of the storage pool, the *Duplicate Data Not Stored* value in the command output is inaccurate and does not reflect the most recent data reduction.

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.

Procedure

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.

About this task

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.

Procedure

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.

Displaying statistics about client-side data deduplication

Use client statistics to compare the amount of data that is transferred during a backup with the amount of data that would be transferred if no deduplication took place. Client statistics compare the amount of transferred data that was unique with the amount of data that was inspected by the client.

About this task

You can use the activity log to view client statistics about data deduplication. The activity log can show historical information about one or more nodes. You can also view data reduction information for data deduplication by using the Tivoli Storage Manager API.

Procedure

To view client statistics for data deduplication, see the activity log, or use the Tivoli Storage Manager API.

Example

The following client statistics are taken from the activity log:

```
tsm> incremental c:\test\* -sub=yes
Incremental backup of volume 'c:\test\*'
Normal File-->        43,387,224 \\naxos\c$\test\newfile [Sent]
Successful incremental backup of '\\naxos\c$\test\*'


Total number of objects inspected:        7
Total number of objects backed up:        1
Total number of objects updated:          0
Total number of objects rebound:          0
Total number of objects deleted:          0
Total number of objects expired:          0
Total number of objects failed:           0
Total objects deduplicated:               1
Total number of bytes inspected:     143.29 MB <= sum of all bytes inspected
Total number of bytes processed:      21.07 MB <= bytes after client-side
   deduplication
Total number of bytes transferred:    21.07 MB <= bytes after compression
Data transfer time:                    1.76 sec
Network data transfer rate:       12,217.36 KB/sec
Aggregate data transfer rate:      4,223.10 KB/sec
Objects compressed by:                    0%
Deduplication reduction:                49.10%  <== 1 - (21.07 / 41.37)
Total data reduction ratio:             85.30%  <== 1 - (21.07 / 143.29)
Elapsed processing time:              00:00:05
ANS1900I Return code is 0.
ANS1901I Highest return code was 0.
```

The `\\naxos\c$\test` directory uses approximately 143.29 MB of space. All files are already stored on the Tivoli Storage Manager server except the `c:\test\newfile` file, which is 41.37 MB (43,387,224 bytes). After client-side data deduplication, it is determined that only approximately 21 MB will be sent to the server.

The following client statistics are produced using the Tivoli Storage Manager API:

```
typedef struct tsmEndSendObjExOut_t
{
  dsUint16_t     stVersion;      /* structure version */
  dsStruct64_t    totalBytesSent;   /* total bytes read from app */
  dsmBool_t     objCompressed;    /* was object compressed */
```

```
  dsStruct64_t    totalCompressSize;  /* total size after compress */
  dsStruct64_t    totalLFBytesSent;  /* total bytes sent LAN free */
  dsUint8_t     encryptionType;   /* type of encryption used  */
  dsmBool_t     objDeduplicated;   /* was object processed for dist. data dedup */
  dsStruct64_t    totalDedupSize;   /* total size after de-dup */
} tsmEndSendObjExOut_t;
```

The API returns the totalDedupSize member of the tsmEndSendObjExOut_t structure. The calling application can calculate the client-side data deduplication reduction. It is calculated by comparing the total number of bytes that are sent to the server with the total deduplication size structure members. If data deduplication took place but no duplicate extents were found, the following calculation is used:

```
totalBytesSent == totalDedupSize
```

To determine whether client-side data deduplication processing took place, examine the Boolean value of the objDeduplicated member of the tsmEndSendObjExOut_t structure.

Querying the server activity log for client-side data deduplication statistics

You can use the Tivoli Storage Manager client reports for data deduplication statistics after each backup or archive operation. To find out the cumulative data reduction for a node or the reduction across multiple nodes, you can query the server activity log.

About this task

After each backup or archive operation, the Tivoli Storage Manager client reports the data deduplication statistics in the server activity log. For details about the activity log, see the Tivoli Storage Manager Information Center, and search for *activity log*.

Procedure

To query the data deduplication statistics for the client, issue the **QUERY ACTLOG** command.

Example

See the following example for sample information provided by the **QUERY ACTLOG** command:

```
tsm: HALLEY>q actlog nodename='modo'

Date/Time        Message
-------------------   ------------------------------------------------------------
03/15/10  09:56:56   ANE4952I (Session: 406, Node: MODO)
 Total number of objects inspected:    1 (SESSION: 406)
03/15/10  09:56:56   ANE4954I (Session: 406, Node: MODO)
 Total number of objects backed up:    1 (SESSION: 406)
03/15/10  09:56:56   ANE4958I (Session: 406, Node: MODO)
 Total number of objects updated:     0 (SESSION: 406)
03/15/10  09:56:56   ANE4960I (Session: 406, Node: MODO)
 Total number of objects rebound:     0 (SESSION: 406)
03/15/10  09:56:56   ANE4957I (Session: 406, Node: MODO)
 Total number of objects deleted:     0 (SESSION: 406)
03/15/10  09:56:56   ANE4970I (Session: 406, Node: MODO)
 Total number of objects expired:     0 (SESSION: 406)
03/15/10  09:56:56   ANE4959I (Session: 406, Node: MODO)
```

```
 Total number of objects failed:      0 (SESSION: 406)
03/15/10  09:56:56   ANE4982I (Session: 406, Node: MODO)
 Total objects deduplicated:        1(SESSION: 406)
03/15/10  09:56:56   ANE4977I (Session: 406, Node: MODO)
 Total number of bytes inspected:   7.05 MB(SESSION: 406)
03/15/10  09:56:56   ANE4975I (Session: 406, Node: MODO)
 Total number of bytes processed:    33 B(SESSION: 406)
03/15/10  09:56:56   ANE4961I (Session: 406, Node: MODO)
 Total number of bytes transferred:   33 B (SESSION: 406)
03/15/10  09:56:56   ANE4963I (Session: 406, Node: MODO)
 Data transfer time:          0.00 sec (SESSION: 406)
03/15/10  09:56:56   ANE4966I (Session: 406, Node: MODO)
 Network data transfer rate:     77.09 KB/sec (SESSION: 406)
03/15/10  09:56:56   ANE4967I (Session: 406, Node: MODO)
 Aggregate data transfer rate:    0.01 KB/sec (SESSION: 406)
03/15/10  09:56:56   ANE4968I (Session: 406, Node: MODO)
 Objects compressed by:          0% (SESSION: 406)
03/15/10  09:56:56   ANE4981I (Session: 406, Node: MODO)
 Deduplication reduction:       100.00%(SESSION: 406)
03/15/10  09:56:56   ANE4976I (Session: 406, Node: MODO)
 Total data reduction ratio:     100.00%(SESSION: 406)
03/15/10  09:56:56   ANE4964I (Session: 406, Node: MODO)
 Elapsed processing time:      00:00:02 (SESSION: 406)
```

The following example shows how to use the activity log to gather the data reduction information across all nodes that belong to the DEDUP domain:

```
dsmadmc -id=admin -password=admin -displaymode=list -scrollprompt=no "select
DISTINCT A1.MESSAGE, A2.MESSAGE from ACTLOG A1, ACTLOG A2 where A1.NODENAME
in (select NODE_NAME from nodes where domain_name='DEDUP') and
A1.SESSID=A2.SESSID and A1.MSGNO=4977 and A2.MSGNO=4961 and EXISTS
(select A3.SESSID from ACTLOG A3 where A3.SESSID=A1.SESSID and A3.MSGNO=4982)"
| grep 'MESSAGE:' | sed -r 's/MESSAGE:.*:\s+([0-9]+(\.[0-9]+)?)\s+
(B|KB|MB|GB|TB).*(SESSION: .*)/\1 \3/' | sed -r 's/\.//' | awk -f awk.txt
```

The `awk.txt` file contains the following statements:

```
BEGIN { bytesInspected = 0; bytesTransferred = 0; }

{ if ($2=="B") valueInKB = 0;
 if ($2=="KB") valueInKB = $1;
 if ($2=="MB") valueInKB = $1 * 1024;
 if ($2=="GB") valueInKB = $1 * 1024 * 1024;
 if ($2=="TB") valueInKB = $1 * 1024 * 1024 *1024;

 if (NR % 2 == 1) bytesInspected += valueInKB;
 if (NR % 2 == 0) bytesTransferred += valueInKB;
}

END {printf("Number of bytes inspected:
%d KB\n Number of bytes transferred: %d
KB\nData reduction ratio:     %d \%\n", bytesInspected,
bytesTransferred, (1-bytesTransferred/bytesInspected)*100)}
```

The **QUERY ACTLOG** command gives a summary, as shown in the following example:

```
Number of bytes inspected:  930808832 KB
Number of bytes transferred: 640679936 KB
Data reduction ratio:     31 %
```

Querying about data deduplication in file spaces

You can view the amount of physical space that is occupied by a file space after removing the deduplication savings that are gained by removing duplicated data from the file space.

Procedure

To query where client file spaces are stored and how much space they occupy, issue the **QUERY OCCUPANCY** command.

Example

In the following example, 10 MB of data is placed in the FS1 file space, and 2 MB is marked for expiration and is removed during the next expiration process. Therefore, Physical Space Occupied reports 10 MB and Logical Space Occupied reports 8 MB. The Physical Space Occupied value for storage pools that use data deduplication is not shown.

```
tsm: SERVER1>q occupancy dedup*

Node Name  Type Filespace   FSID Storage    Number of  Physical   Logical
               Name             Pool Name      Files     Space     Space
                                                       Occupied  Occupied
                                                           (MB)      (MB)
---------- ---- ---------- ----- ---------- --------- --------- ---------
DEDUP_N    Bkup \\abcde\c$    17 NODEDUPOOL        63     10.00      8.00
DEDUP_Y    Bkup \\abcde\d$    18 YADEDUPOOL        63         -      8.00
```

The occupancy table shows how much physical space is occupied by a file space after the removal of the deduplication savings. These savings are gained by removing duplicated data from the file space. You can use **select * from occupancy** to get LOGICAL_MB and REPORTING_MB values.

LOGICAL_MB is the amount of space that is used by this file space. REPORTING_MB is the amount of space that is occupied when the data is not placed in a deduplication-enabled storage pool.

For example, the \\brian\c$ file space manages 30 MB of space, but 10 MB is eliminated by either client-side or server-side data deduplication. The resulting occupancy tables shows:

```
tsm: SERVER1>select * from occupancy where NODE_NAME='BRIAN'

     NODE_NAME: BRIAN
         TYPE: Bkup
FILESPACE_NAME: \\brain\c$
  STGPOOL_NAME: MYFILEPOOL
     NUM_FILES: 63
   PHYSICAL_MB: 0.00
    LOGICAL_MB: 10.00
  REPORTING_MB: 30.00
  FILESPACE_ID: 17
```

Tip: The LOGICAL_MB value takes into account only the amount of data that is removed or not stored because the data is identified as a duplicate of data that is stored elsewhere.

For example, the \\brian\c$ file space backs up 20 MB of unique data, which means that LOGICAL_MB is 20 MB. The \\jim\c$ file space must not back up any data because it is an exact duplicate of the \\brian\c$ file space, which means that

LOGICAL_MB is 0 MB. In this case, the \\brian\c$ file space does not report any duplication efficiencies, even though the \\jim\c$ file space is able to identify duplicate data between the file spaces.

Scenarios for data deduplication

You can benefit from client-side data deduplication when it is used in combination with other IBM Tivoli Storage Manager functions and products. You can reduce storage requirements when you use client-side data deduplication with image backups, system state backups, and file-level backups. When you use caching with data deduplication, you can reduce network traffic.

For example, IBM Tivoli Storage Manager for Mail and IBM Tivoli Storage Manager for Databases can use client-side data deduplication through the Tivoli Storage Manager API to create backup sets and export node data.

The progressive incremental backup technology in Tivoli Storage Manager provides an effective method of efficiently reducing the amount of data that is processed in each backup. This technology can also be effectively combined with data deduplication. When used in combination, data is initially reduced by the incremental processing which is able to skip unchanged objects without applying deduplication processing against them. For those objects which do require a backup, data deduplication is applied.

Another Tivoli Storage Manager backup model is the use of incremental-forever backups. With this technology, each subsequent backup achieves significant storage savings by not having to send unchanged objects. These unchanged objects then do not need to go through deduplication processing, which is an efficient method of reducing data.

Reducing the storage requirement of image backups

With client-side data deduplication, you can further reduce the storage requirement of image backups. Tivoli Storage Manager can perform a snapshot-based image backup during which the volume is available to other system applications.

Image backup can be full or incremental. In a typical scenario, full image backups are scheduled less frequently than incremental image backups. For example, a full image backup is scheduled weekly and incremental backups are scheduled daily, except for the day of the selective image backup. The frequency of full image backups is often driven by the available storage space. For example, each image backup of a 50 GB volume might need 50 GB of space in a storage pool.

One approach to reduce the required storage space is to use compression. To further reduce the storage footprint of image backups, you can use client-side data deduplication. With data deduplication, you can perform more frequent full image backups without significant use of storage space. The reduced need for storage space is because only unique data is sent to the server during an image backup operation.

Reducing the storage requirement of system state backups

With client-side data deduplication, you can reduce the amount of duplicate data that is stored on the server during system state backups. *System state* consists of all system state and system services components that can be started. Tivoli Storage Manager uses Microsoft Volume Shadow Copy Service (VSS) to back up all system state components as a single object, which provides a consistent point-in-time snapshot of the system state.

You can use VSS on Windows Server 2003, Windows Server 2008, and Windows Vista operating systems. For details about backing up the Windows system state, see *Tivoli Storage Manager: Client Installation and User Guide*.

System state can contain thousands of objects and take a large amount of storage space on the server. It is likely that system state objects do not change much between backups. This results in a large amount of duplicate data being stored on the server. In addition, similar systems are likely to have a similar system state. Therefore, when you perform system state backups on these systems, there is an increase in duplicate data.

System state backup, when used in combination with client-side data deduplication, reduces the amount of duplicate data that is stored on the server. The reduction amount can vary, depending on the makeup of your data.

In the following example, a backup of the system state was performed on two similar systems that run Windows Server 2008. There was no data backed up to the storage pool. On the first system, the system-state data was deduplicated by 45%, as shown in Figure 22. A backup of the system state yielded a deduplication reduction of 98% on the second system, as shown in Figure 23 on page 322.

This example shows a sample deduplication reduction of 45% for the system state data:

```
Total number of objects inspected:  63,800
Total number of objects backed up:  63,800
Total number of objects updated:     0
Total number of objects rebound:     0
Total number of objects deleted:     0
Total number of objects expired:     0
Total number of objects failed:      0
Total objects deduplicated:     43,302
Total number of bytes inspected:   9.77 GB
Total number of bytes processed:   5.32 GB
Total number of bytes transferred:  5.32 GB
Data transfer time:          70.30 sec
Network data transfer rate:    79,431.06 KB/sec
Aggregate data transfer rate:    482.50 KB/sec
Objects compressed by:          0%
Deduplication reduction:       45.75%
Total data reduction ratio:      45.51%
Elapsed processing time:      03:12:54
```

Figure 22. Final statistics of the system state backup on system 1

This example shows a sample deduplication reduction of 98% for the system state data:

```
Total number of objects inspected:  55,007
Total number of objects backed up:  55,007
Total number of objects updated:     0
Total number of objects rebound:     0
Total number of objects deleted:     0
Total number of objects expired:     0
Total number of objects failed:      0
Total objects deduplicated:     34,606
Total number of bytes inspected:   6.19 GB
Total number of bytes processed:   152.25 MB
Total number of bytes transferred:  152.25 MB
Data transfer time:          3.68 sec
Network data transfer rate:    42,354.40 KB/sec
Aggregate data transfer rate:     63.47 KB/sec
Objects compressed by:          0%
Deduplication reduction:       98.04%
Total data reduction ratio:      97.60%
Elapsed processing time:      00:40:56
```

Figure 23. Final statistics of the system state backup on system 2

Reducing the storage requirement of Tivoli Storage Manager for Virtual Environments backups

A Tivoli Storage Manager for Virtual Environments backup can take a large amount of storage space on the server. Using client-side data deduplication, you can reduce the amount of duplicate data that is stored during a backup operation.

Tivoli Storage Manager for Virtual Environments backups consist of all virtual machines in the environment. Often, large portions of individual backups are common with other backups. Therefore, when you perform backup operations, there is an increase in duplicate data.

When you use client-side data deduplication in combination with backups for Tivoli Storage Manager for Virtual Environments, you can reduce the amount of duplicate data that is stored on the server. The reduction amount varies, depending on the makeup of your data.

Reducing the storage requirement of Tivoli Storage Manager for Mail and Tivoli Storage Manager for Databases

You can use client-side data deduplication with Tivoli Storage Manager for Mail and Tivoli Storage Manager for Databases through the Tivoli Storage Manager API. Using client-side deduplication with these products reduces space requirements.

Before you use data deduplication, ensure that your system meets all prerequisites.

You can turn on client-side data deduplication by adding `DEDUPLICATION YES` to the `dsm.sys` file.

Related concepts:

"Client-side data deduplication" on page 282

Verifying data deduplication for Tivoli Storage Manager for Mail and Tivoli Storage Manager for Databases:

In Tivoli Storage Manager V6.1 or earlier, data protection clients do not provide data deduplication reduction statistics in the graphical user interface. In this situation, you can verify that data deduplication occurs.

Procedure

To verify data deduplication in V6.1 or earlier, complete the following steps:

1. Add the following statements to the `dsm.sys` file:

   ```
   TRACEFILE api.txt
   TRACEFLAGS dedup api
   ```

2. Locate a trace entry for the tsmEndSendObjEx API call. The entry looks like the following example:

   ```
   03/05/2010 13:41:16.286 : dsmsend.cpp     (2052): tsmEndSendObjEx: Total bytes
   send 0 2105534, encryptType is NO encryptAlg is NONE compress is 0, dedup
   is 1, totalCompress is 0 0 totalLFBytesSent 0 0 totalDedupSize 0 15001
   ```

   In the example, the **dedup** parameter of the **TRACEFLAGS** statement indicates whether the objects were processed for client-side data deduplication. The **totalDedupSize** parameter of the **TRACEFLAGS** statement reports the number of bytes that were sent to the server after deduplication.

   With image and system-state backups, you can have more frequent full backups. Similarly, with client-side data deduplication you can have more frequent full backups of the application data, for example, with backups of databases and storage groups.

Reducing the storage requirement of file-level backups

The Tivoli Storage Manager backup-archive client provides several data reduction methods. These methods include compression, incremental backup processing, subfile backup, and data deduplication. With client-side data deduplication, you can reduce the amount of data that is sent to the server during file-level backup.

When only the metadata of the file is changed, for example, with access control lists or extended attributes, typically the whole file is backed up again. With client-side data deduplication, although the file is backed up again, only the metadata is sent to the server.

Reducing network traffic by using data deduplication cache

You can avoid queries to the Tivoli Storage Manager server during data deduplication when you cache hash sums that are already sent to the server. It is beneficial with client-side data deduplication to query the server, for example, when extents are identified and hash sums calculated. Also, data deduplication cache can be used to your advantage in some situations when multiple processes run concurrently. When you use data deduplication cache, consider factors such as insufficient file space and synchronization with the server that might impact on data deduplication.

Client-side data deduplication identifies extents in the data stream and calculates the associated hash sums. Data deduplication determines whether a data extent with the same hash sum is already stored on the server. If it is already stored, the backup-archive client only needs to notify the server about the hash sum, and can avoid sending the corresponding data extent. This process reduces the amount of data that is exchanged between the Tivoli Storage Manager backup-archive client and the server.

You can avoid having to query the server when the hash sums that are already sent to the server are cached. When additional data extents are identified with identical hash sums, the server can be notified of the hash sum, saving additional network traffic. The time spent on the query and response is also reduced. In most cases, it is beneficial to use client deduplication cache for these hash sums.

Multiple processes and API applications

When multiple processes, such as concurrent backups or Tivoli Storage Manager API applications, transfer content concurrently, they cannot use the cache. In such a situation, only a single process between the open and close processes can access the client data deduplication cache. Multi-session backups from the same process can use the deduplication cache. In general, processes that do not get the lock on the cache can perform client-side data deduplication. However, the processes cannot query or store hash sums that are found in the cache.

Insufficient file space

The Tivoli Storage Manager client cannot use a cache for data deduplication if there is not enough file space for a hash sum cache. Client-side data deduplication can take place, but it has no memory of hash sums that are already sent by the client or already found on the server. Data deduplication, generally, must query the server to find out if hash sums are duplicates. Hash sum lists are maintained in memory for the life of a transaction. If a hash sum is encountered multiple times within the same transaction, the hash sum is detectable without a cache.

Synchronization of the cache for client-side data deduplication

The cache for client-side data deduplication can become unsynchronized with the deduplicated disk storage pool of the server. Object expiration, file space deletion, or overflow to an associated tape storage pool can cause the cache to be unsynchronized. When the client cache contains entries that are no longer in the deduplicated storage pool of the Tivoli Storage Manager server, the client cache resets. The client cache cannot delete specific entries when objects are deleted from the storage pool of the server.

In environments where it is likely that the client deduplication cache would become unsynchronized with the deduplicated storage pool of the server, use server-side data deduplication. The cache can become unsynchronized during a backup or an API application function. The Tivoli Storage Manager client can back up data again, but the Tivoli Storage Manager API cannot. Wherever it is difficult to perform a backup again, do not use the cache of client-side data deduplication.

Restoring deduplicated data from backup sets

You can restore deduplicated objects to the same storage pool or to another storage pool that uses or does not use data deduplication.

When a backup set is created for a node by using the **GENERATE BACKUPSET** command, all associated node data is placed onto the backup media. It is also placed on the backup media when node data is exported for a node by the **EXPORT NODE** command. This placement ensures that the associated objects can be restored without any server dependencies, apart from the backup media.

When you restore deduplicated objects, consider the following situations:

- If the object is restored to the same storage pool, for example, by the **RECLAIM STGPOOL** command, any duplicate data blocks are replaced with references to deduplicated extents. These duplicate data blocks were identified by the IDENTIFY processes.
- If a deduplicated object is moved to another storage pool that uses deduplication, duplicate extents are processed as the object is moved. The object is placed in the target storage pool in a deduplicated state, and no additional processing is needed on that object. The deduplicated object is moved, for example, by the **MOVE VOLUME** command or by storage pool migration.
- If a deduplicated object is moved to a storage pool that does not use deduplication, that object is created in the target storage pool. All associated data is also in the target storage pool.

Data deduplication and data compatibility

With client-side data deduplication, data is compressed extent by extent, which means data compatibility is possible between the client and the server and between different versions of Tivoli Storage Manager. Also, data extents that are created by different operations are compatible, which can mean that a greater proportion can be deduplicated.

Compression

With client-side data deduplication, compression is done extent by extent. When you use client-side data deduplication, data is compatible between compressed extents and extents that are not compressed. This compatibility means that extents can be shared between the client and server, regardless of how the data was backed up. Compression can save storage space, however, it can increase processor usage on the client system.

Consider the following factors when you use data compression in an environment that uses multiple clients:

- Extents that are compressed by a backup-archive client that uses Tivoli Storage Manager V6.1 or earlier are not compatible with compressed extents from a V6.2 client. Extents are also not compatible with uncompressed extents because each version uses a different compression algorithm.
- With a deduplication storage pool that contains data from clients that are V6.2 and earlier, there is a mixture of compressed and non-compressed extents. For example, assume that a restore operation is run from a client that is earlier than V6.2. Compressed extents from a client at a later version of Tivoli Storage Manager are uncompressed by the server during the restore operation.
- When backup sets are generated for clients that are at a version earlier than V6.2, V6.2 compressed extents that are also part of the data to be backed up are uncompressed.

Even though most data is compatible when using compression, ensure that all clients are at V6.2 and later. This method minimizes the need for data compression when you restore data or create a backup set.

Compatibility with data from earlier client versions

Data that is stored by earlier client versions and processed for deduplication extents by the server is compatible with new extents. For example, an extent that is identified by the server from an earlier client version matches the query from client-side data deduplication to the server. The extent is not sent to the server,

only a reference to it is sent. Therefore, in the same storage pool, extents from different levels of the client are equivalent. Regardless of where extents are processed, whether on the server or on the client, they are compatible. Migration of client and server versions is possible without the data being sent again to the server.

File image and Tivoli Storage Manager FastBack® mount backups

Data extents that are created by different operations are compatible. For example, data extents are compatible that are created by file-level, image, or IBM Tivoli Storage Manager FastBack mount backups. This can mean that a greater proportion of the extents can be deduplicated.

Image and Tivoli Storage Manager mount backups

Assume that you integrate the Tivoli Storage Manager FastBack mount with Tivoli Storage Manager to back up volumes to a Tivoli Storage Manager server. The Tivoli Storage Manager client backs up the Tivoli Storage Manager FastBack repository to a remote server. You previously performed an image or a file-level backup of this data with Tivoli Storage Manager client. Then it is likely that the Tivoli Storage Manager FastBack mount backup can use many data extents that are already stored on the server.

For example, you perform an image backup of a volume that uses the Tivoli Storage Manager client. Then you back up the same volume with Tivoli Storage Manager FastBack. You can expect a greater amount of data deduplication when you back up the Tivoli Storage Manager FastBack mount.

Image and file-level backups

Data extents that are created by a file-level backup can be used by the Tivoli Storage Manager client during an image backup. For example, you perform a full incremental backup of the `C` drive on your computer. Then you run an image backup of the same drive. You can expect a greater amount of data deduplication during the image backup. You can also expect a greater amount of data deduplication during a file-level backup or an archive operation that immediately follows an image backup.

Data deduplication and disaster recovery management

Use the following scenarios to help you consider factors that can affect disaster recovery management when you use deduplicated primary or copy storage pools.

Data deduplication is only permitted for storage pools that are associated with a devtype=FILE device class. The following scenarios show how you can implement the data deduplication of storage pools to ensure that you can restore data if a failure occurs.

Primary storage pool is deduplicated and a single copy storage pool is not deduplicated

The amount of time required to back up the primary storage pool to a non-deduplicated copy storage pool can increase. While data is copied to the copy storage pool, the deduplicated data that represents a file must be read. The file must be recreated and stored in the copy storage pool.

With this scenario, to recover data at a disaster recovery site, you can use disaster recovery management and copy storage pool volumes that are off-site.

Primary storage pool is deduplicated and multiple copy storage pools are not deduplicated

The amount of time required to back up the primary storage pool can increase. This is because of the reads that are necessary to get the various data that represent an object written to one of these copy storage pools.

With this scenario, to recover data at a disaster recovery site, you can use disaster recovery management and copy storage pool volumes that are off-site. There is also a copy of the storage pool data on-site, in addition to the off-site support.

Primary storage pool is deduplicated and a single copy storage pool is deduplicated

The main aim of this scenario is space savings at the primary server location. However, you cannot use disaster recovery management. Also, you cannot use the off-site rotation of copy storage pool volumes because the deduplicated copy storage pool can be achieved only by using devtype=FILE volumes.

With this scenario, you can have a risk in that there is no off-site copy of the data. The off-site copy could be used if the primary server location, hardware, or data center is damaged or destroyed. This scenario can result in the loss of data or the inability to recover that data.

Primary storage pool is deduplicated, a single copy storage pool is not deduplicated, and a single copy storage pool is deduplicated

The deduplicated copy storage pool provides space savings for the primary server location and the duplication of the data in the primary storage pool. You use server-side data deduplication at the level of a single storage pool. Therefore, a deduplicated primary storage pool and a deduplicated copy storage pool reference different data. They also maintain different database references to track and manage the data that represent a given file.

Alternatively, you can use the non-deduplicated copy storage pool, which is probably tape, along with disaster recovery management as an off-site data protection site. You can use the copy storage pool volumes that are off-site, together with an appropriate database backup, to restore the server. You can also use the volumes to restore or retrieve data.

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.

About this task

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 ACTIVEDATA** 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 ACTIVEDATA** 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 simultaneous-write operations

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
  - MOVE DATA processes
  - MOVE NODEDATA processes
  - Storage pool backup
  - Copy of active data to an active-data storage pool

The only server-data movement process that can write data simultaneously is migration.

- Simultaneous-write operations take precedence over LAN-free data movement. The operations go over the LAN, and the simultaneous-write configuration is accepted.
- 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 that are used for simultaneous-write operations can have different device classes if the device classes have compatible data formats. Performance is limited by the speed of the slowest device.
- Data deduplication has the following effects on simultaneous-write operations:

  Simultaneous write during client store operations

  If client-side deduplication is enabled, it is disabled before simultaneous-write operations occur. Non-deduplicated data is simultaneously written to target storage pools.

  Simultaneous write during server migration of data

  If a primary storage pool is set up for server-side deduplication, all copy storage pools and active-data pools are removed from simultaneous-write operations. For example, suppose that your simultaneous-write configuration comprises a primary storage pool, a copy storage pool, and an active-data pool. The primary storage pool is set up for server-side deduplication. Data is written only to the primary storage pool.

  If a primary storage pool is not set up for server-side deduplication, data is written simultaneously only to copy storage pools or active data pools that are not set up for server-side deduplication. For example, suppose that your simultaneous-write configuration comprises a primary storage pool, a copy storage pool, and an active-data pool. The primary storage pool and the active-data pool are not set up for server-side deduplication. However, the copy storage pool is set up for server-side deduplication. When simultaneous-write operations occur, data is written only to the primary storage pool and the active-data pool.
- You cannot use the simultaneous-write function with Centera storage devices.
- 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.

About this task

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:

Procedure

- 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.

Example

The following table provides examples of **AUTOCOPY** settings for some common scenarios in which the simultaneous-write function is used.

Table 30. **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.

About this task

The parameters that are used to specify copy storage pools and active-data pools are on the **DEFINE STGPOOL** and **UPDATE STGPOOL** commands.

Procedure

- To specify copy storage pools, use the **COPYSTGPOOLS** parameter.
- To specify active-data pools, use the **ACTIVEDATAPOOLS** parameter.

Results

Ensure that client sessions have sufficient mount points. Each session requires one mount point for the primary storage pool and a mount point for each copy storage pool and each active-data pool. To allow a sufficient number of mounts points, use the **MAXNUMMP** parameter on the **REGISTER NODE** or **UPDATE NODE** commands.

Example

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 334

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.

About this task

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:

Procedure

- 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**.

What to do next

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"

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 332

"Specifying how the server reacts to a write failure during simultaneous-write operations" on page 333

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 13, "Implementing policies for client data," on page 469

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 copypool1 mytapedevice pooltype=copy
define stgpool copypool2 mytapedevice pooltype=copy
define stgpool activedatapool mydiskdevice pooltype=activedata
update stgpool diskpool 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.

The storage pool hierarchy and the copy storage pools and active-data pool associated with DISKPOOL are displayed in Figure 24 on page 336.

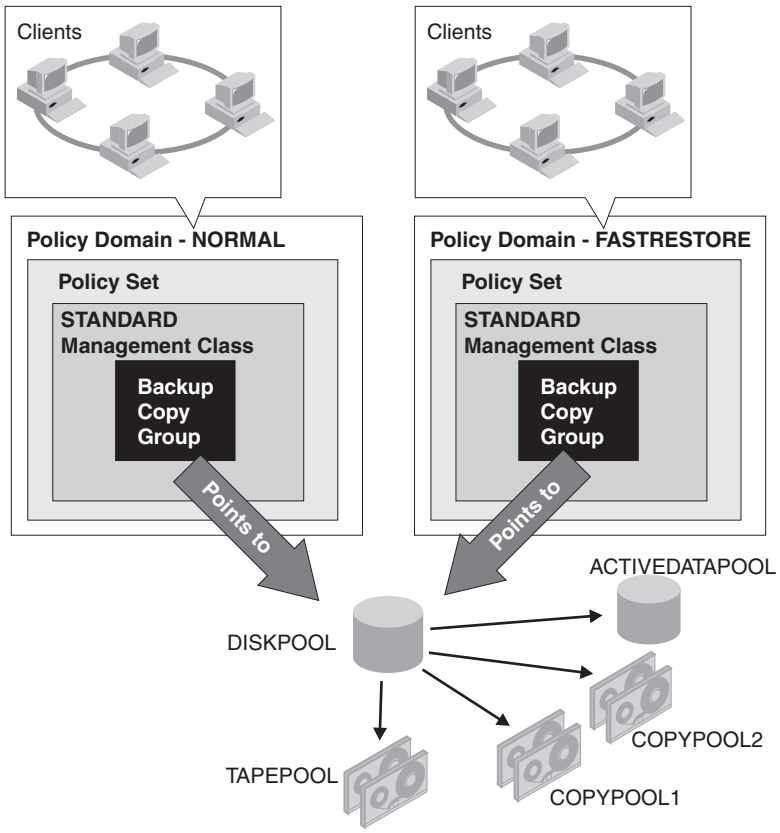


Figure 24. Example of storage pool hierarchy with copy storage pools defined for DISKPOOL

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 25 on page 337.

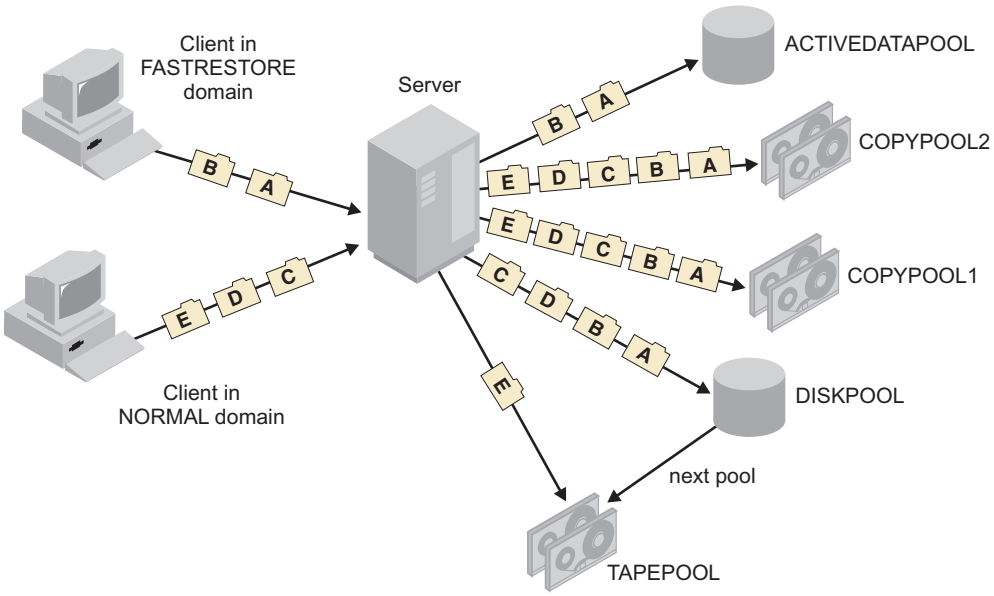


Figure 25. Inheriting a list of copy storage pools

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 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 (ACTIVEDATAPOOL). 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 26 on page 338 displays this configuration.

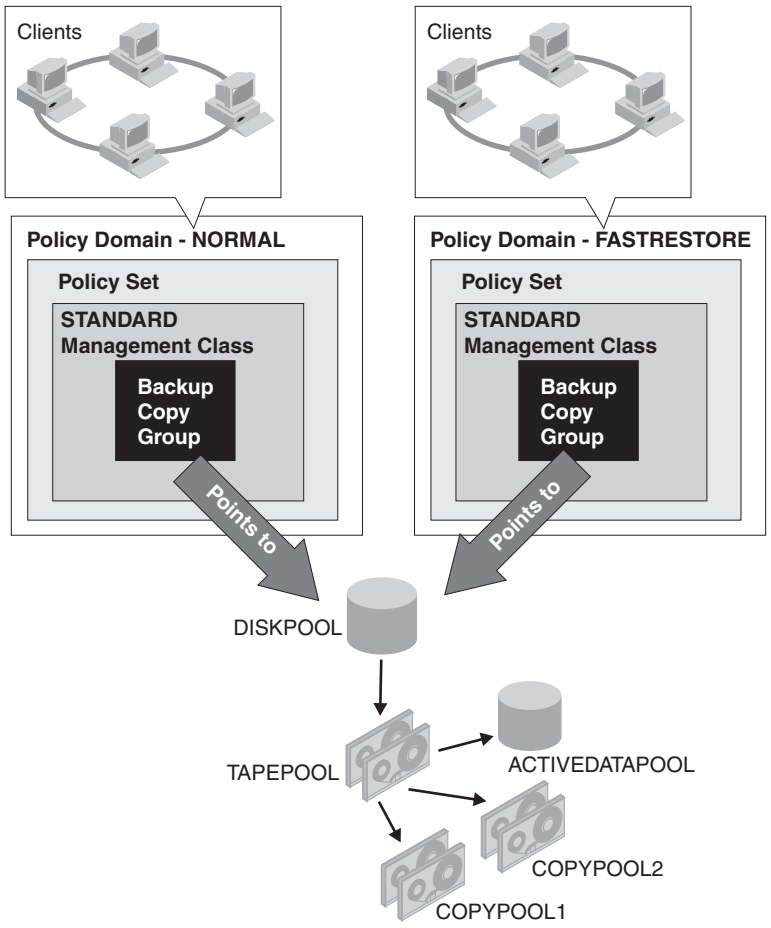


Figure 26. Example of storage pool hierarchy with copy storage pools defined for TAPEPOOL

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 27 on page 339.

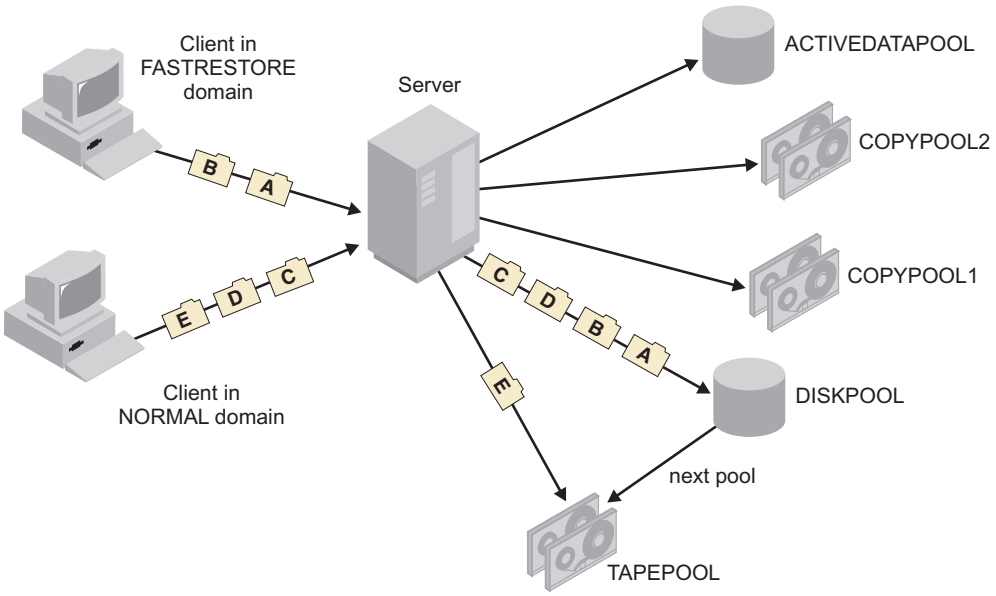


Figure 27. Inheriting an empty copy storage pool list

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 28 on page 340.

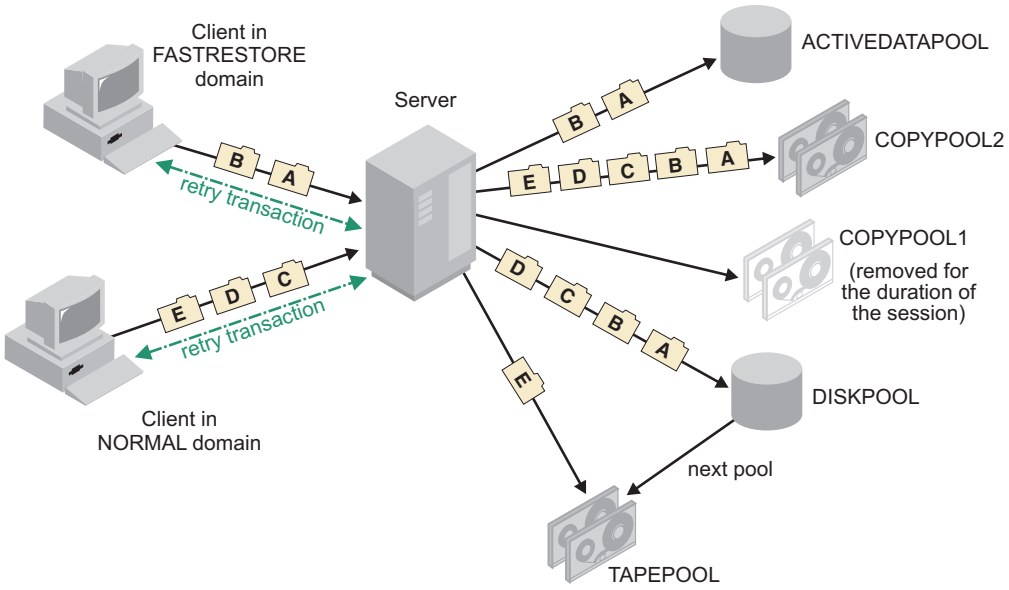


Figure 28. Inheriting a list of copy storage pools

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 29.

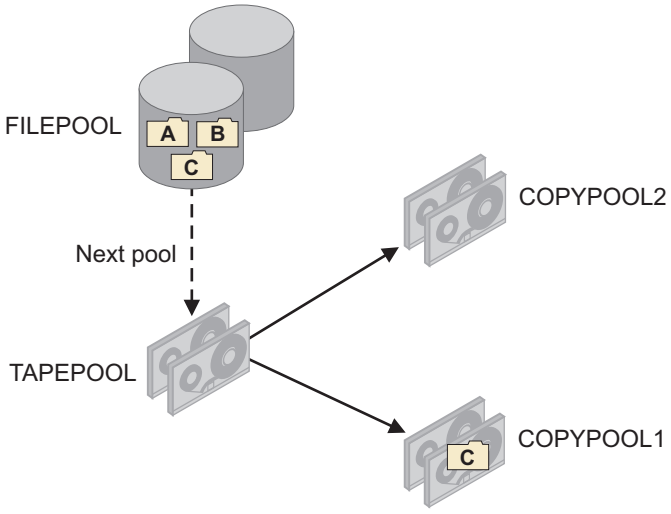


Figure 29. A storage pool hierarchy with files eligible to be migrated

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:

- 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 30.

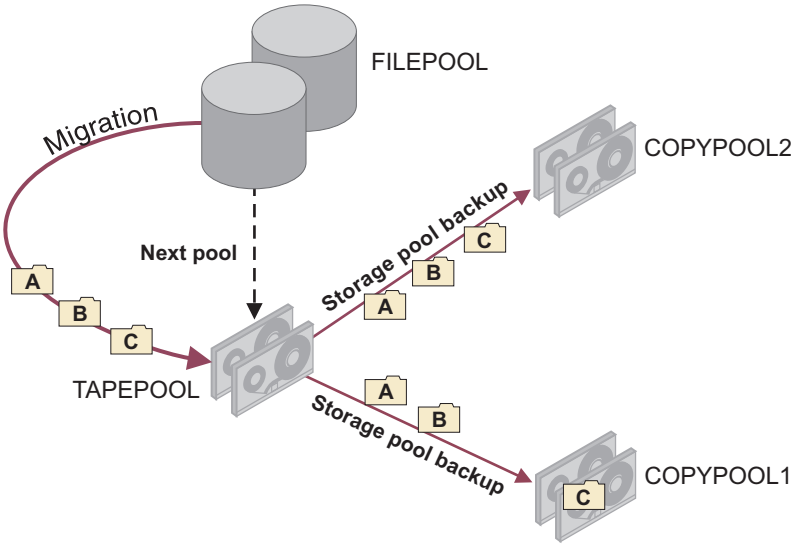


Figure 30. Simultaneous-write operation during migration to two copy storage pools

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.

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 31 on page 343.

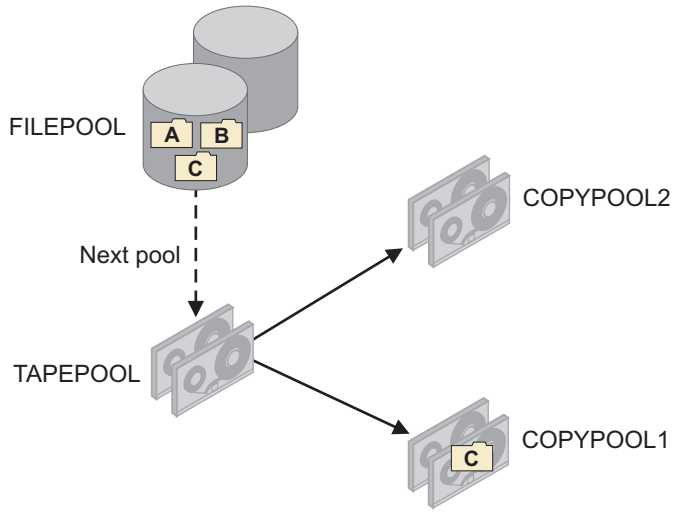


Figure 31. A storage pool hierarchy with files eligible to be migrated

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 32.

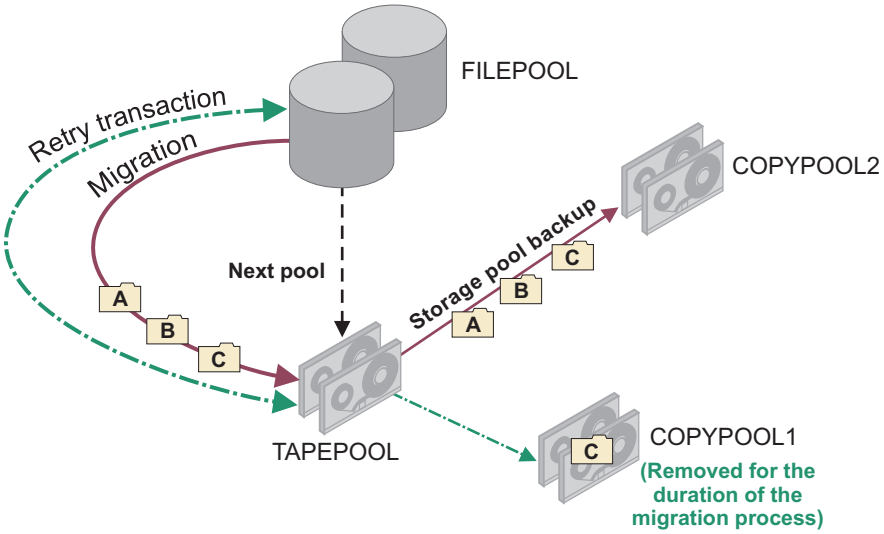


Figure 32. An error occurs during simultaneous-write operation during migration

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.

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 33.

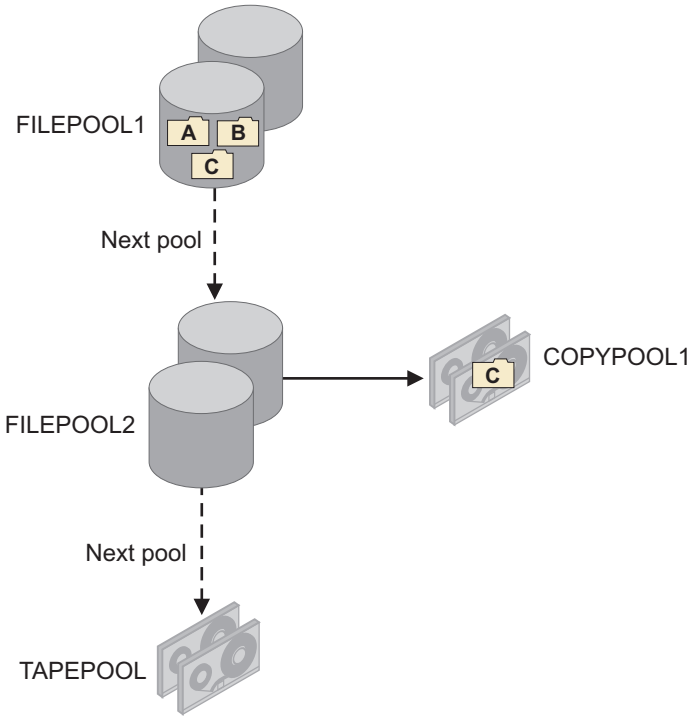


Figure 33. Three-tiered storage pool hierarchy with one copy storage pool

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 34.

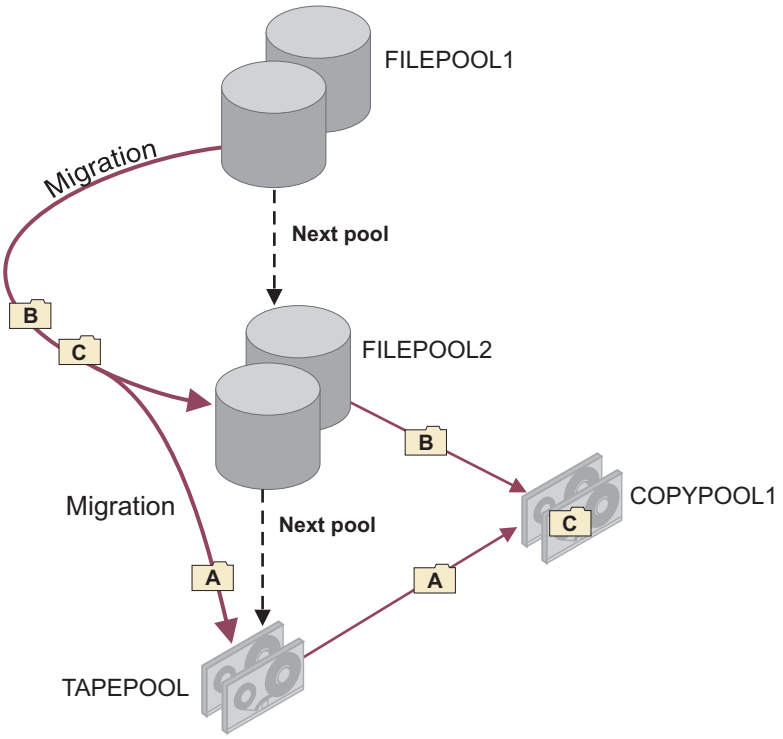


Figure 34. Inheriting a list of copy storage pools

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 35.

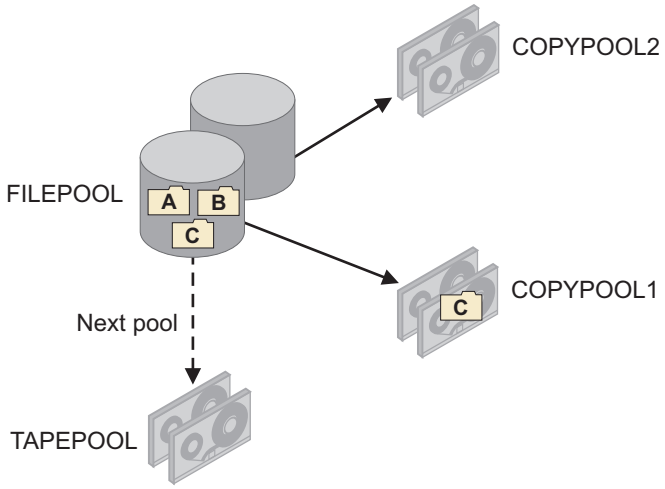


Figure 35. Storage pool hierarchy with files eligible to be migrated

When files A, B and C are migrated, they are written to TAPEPOOL. See Figure 36.

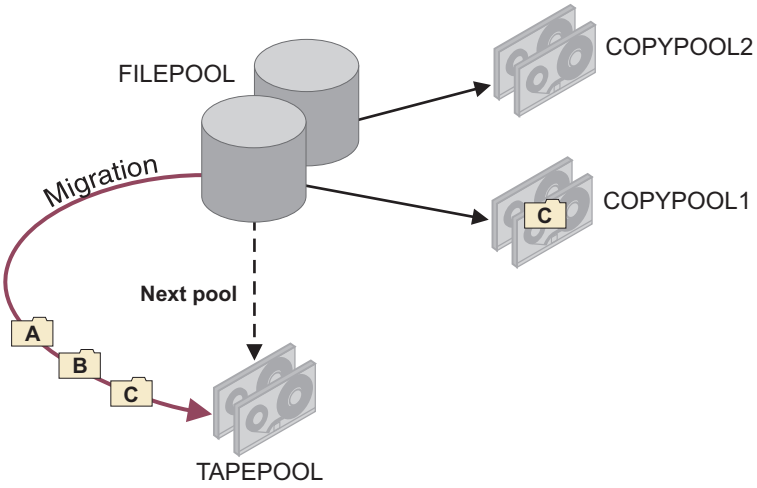


Figure 36. Migration after a simultaneous-write operation

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 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 37.

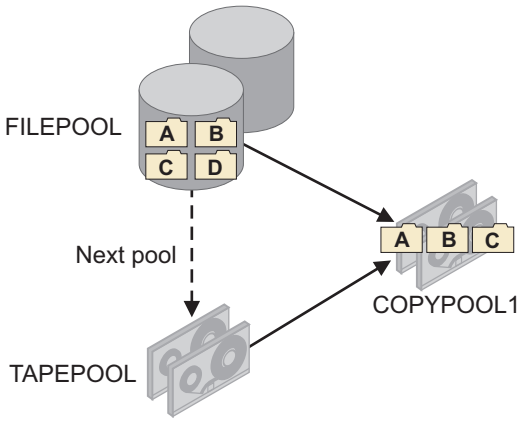


Figure 37. Storage pool hierarchy with files eligible to be migrated

When files A, B, C, and D are migrated, the follow 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 38 on page 348.

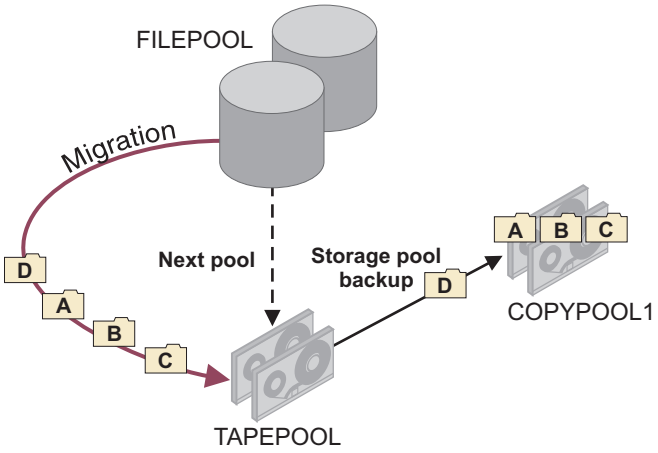


Figure 38. A simultaneous-write operation during both migration and client backup operations

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.

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.

About this task

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 24 on page 336. 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.

About this task

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 ACTIVEDATA** commands to store data in copy storage pools and active-data pools.

About this task

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 ACTIVEDATA** 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.

About this task

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 39 on page 352. 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.

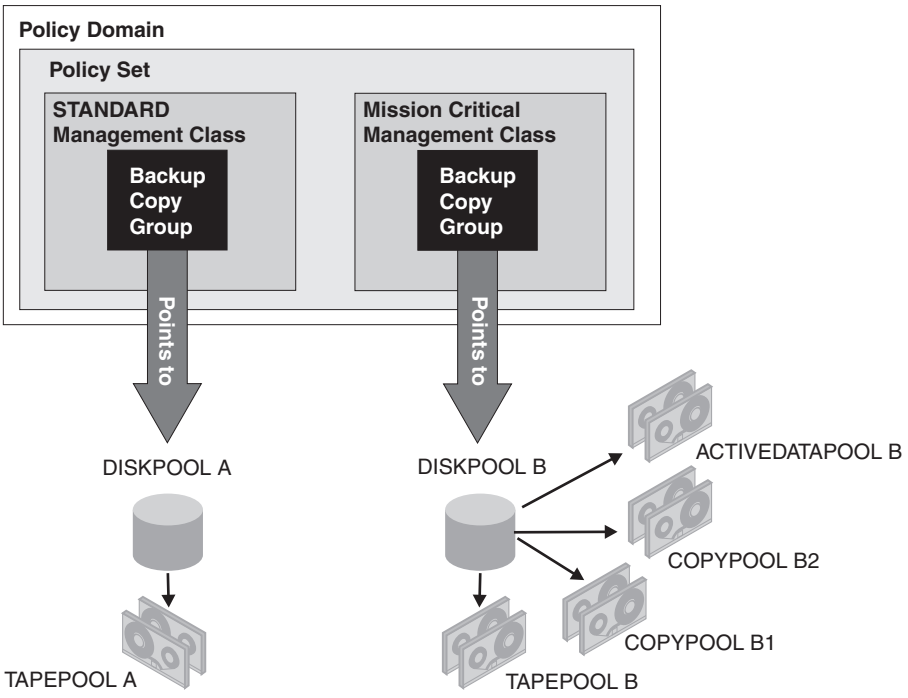


Figure 39. Separate storage pool hierarchies for different types of data

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.

About this task

This example also shows how to use the **COPY ACTIVEDATA** 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.

Procedure

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 onnsitepool mytapedevice
   define stgpool drcopypoool 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 24 on page 336.

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

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.

About this task

With collocation enabled, the server tries to keep files on a minimal number of sequential-access storage volumes. The files can belong to a single client node, a group of client nodes, a client file space, or a group of file spaces. You can set collocation for each sequential-access storage pool when you define or update the pool.

Figure 40 on page 354 shows an example of collocation by client node with three clients, each having a separate volume containing that client's data.

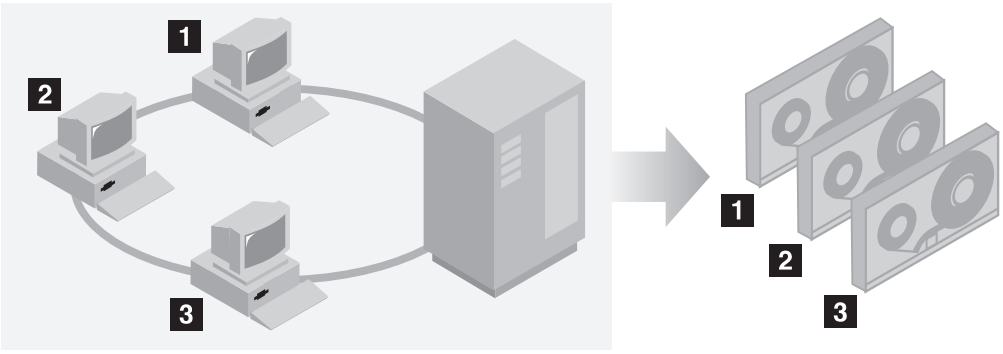


Figure 40. Example of collocation enabled

Figure 41 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.

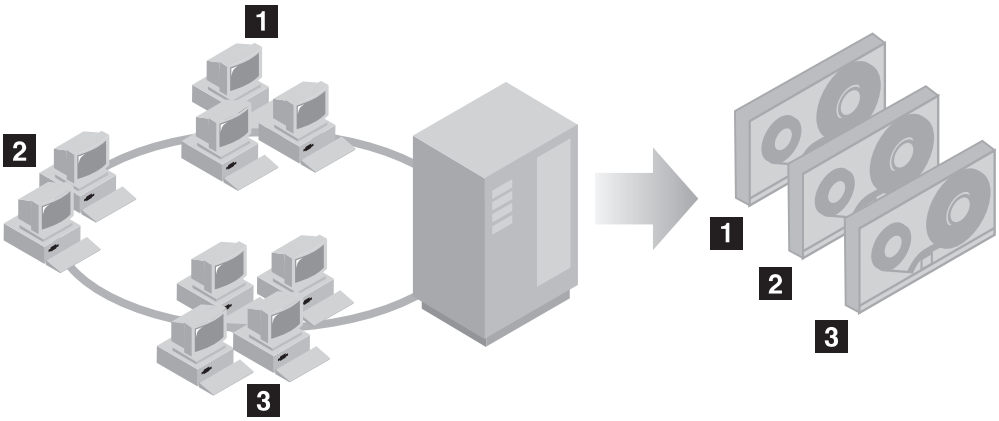


Figure 41. Example of collocation enabled by node collocation group

Figure 42 shows an example of collocation by file space group. Six groups have been defined. Each group contains data from file spaces that belong to a single node. The data for each group is stored on a separate volume.

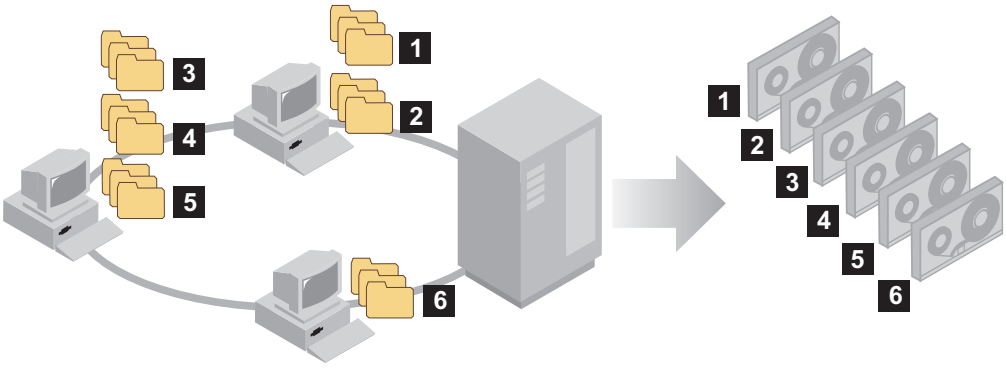


Figure 42. Example of collocation enabled by file space collocation group

When collocation is disabled, the server tries 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 43 on page 355 shows an example of collocation disabled, with

three clients sharing space on single volume.

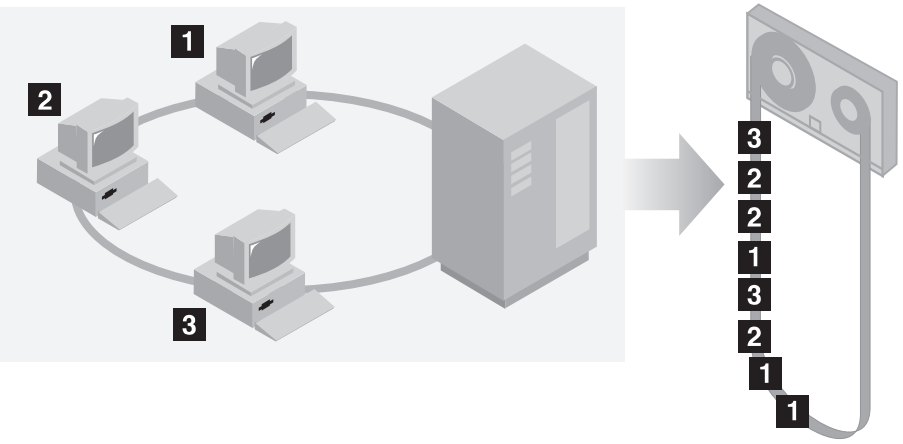


Figure 43. Example of collocation disabled

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 run.

Table 31 summarizes the effects of collocation on operations.

Table 31. Effect of collocation on operations

Operation	Collocation Enabled	Collocation Disabled
Backing up, archiving, or migrating client files	More media mounts to collocate files.	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 on fewer volumes.	Multiple mounts of media might be required for a single user because files might 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 try to recover a file that is on the same volume, the second user is 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 you use another tape volume.

Table 31. Effect of collocation on operations (continued)

Operation	Collocation Enabled	Collocation Disabled
Media mount operations	More mount operations when user files are backed up, archived, or migrated from client nodes directly to sequential-access volumes. More mount operations during reclamation and storage pool migration. More volumes to manage because volumes are not fully used.	More mount operations are required during restore, retrieve, and recall of client files.
Generating backup sets	Less time is spent searching database entries and fewer mount operations.	More time is spent searching database entries and fewer mount operations.

When collocation is enabled for a group, single client node or file space, all the data that belongs to the group, the node, or the file space is moved or copied by one server process. For example, if data is collocated by group, all data for all nodes that belong to the same collocation group is migrated by the same process.

When collocating data, the Tivoli Storage Manager server tries 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 that you are collocating by node and you specify that a node can use two mount points on the server. Suppose also that the data that is backed up from the node can 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. When you used collocation, the following server operations use one server process:

- Moving data from random-access and sequential-access volumes
- Moving node data from sequential-access volumes
- Backing up a random-access or sequential-access storage pool
- Restoring a sequential-access storage pool
- Reclamation of a sequential-access storage pool or off-site volumes
- Migration from a random-access storage pool.

When migrating data from a random disk storage pool to a sequential storage pool, and collocation is by node or file space, nodes or file spaces are automatically 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 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 that belong to that collocation group. This process takes place, regardless of how much data is stored in the file spaces of nodes and regardless of whether the low migration threshold was reached.

However, when migrating collocated data from a sequential storage pool to another sequential storage pool, the server orders the volumes according to the date when the volume was last accessed. The volume with the earliest access date is migrated first, and the volume with the latest access date is migrated last.

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. Also, collocating data by groups of file spaces reduces the unused tape to a finer degree.

The data that belongs to all the nodes in the same collocation group are migrated by the same process. Therefore, collocation by group can reduce the number of times a volume that is going 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 32 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 32. 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 that contains 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 must continue to store data on a second volume, it uses the following selection order to acquire more 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 tries to provide the best use of individual volumes and minimizes file mixing from different clients or file spaces on volumes. This configuration is depicted in Figure 44 on page 358, 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.

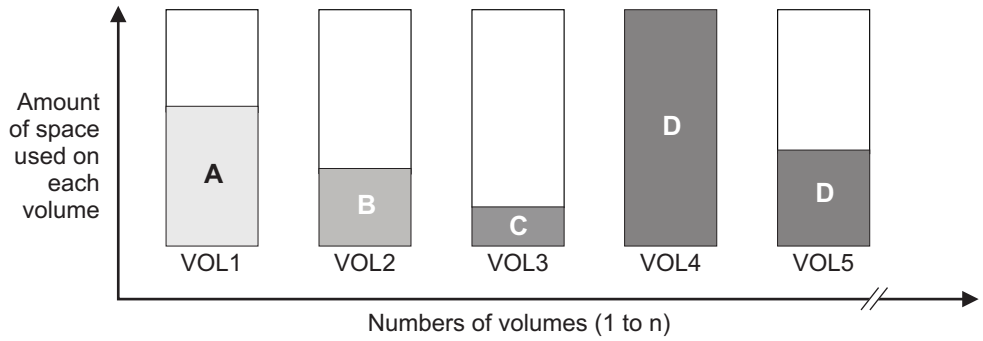


Figure 44. Using all available sequential access storage volumes with collocation enabled at the node or file space level

Collocation can be by filespace group or node group. When collocation is by node group (node collocation group), the server tries to collocate data from nodes that belong to the same collocation group. A filespace collocation group uses the same methods as a node collocation group, but can use up more space because of the granularity of filespace sizes. As shown in Figure 45, data for the following groups of nodes was 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 that belongs 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 that are 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.

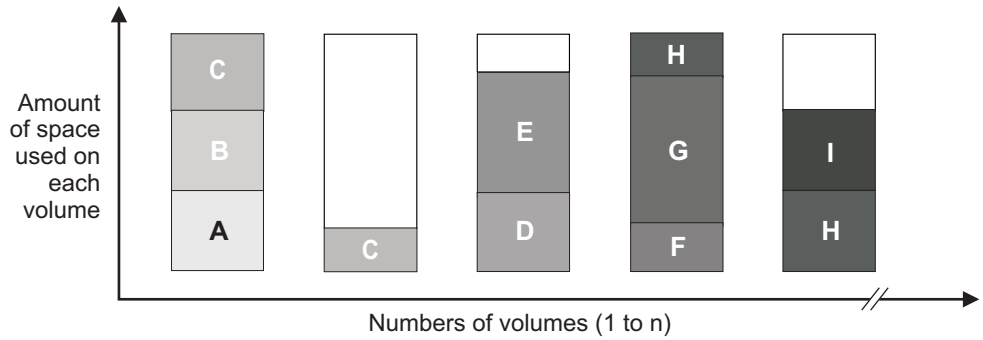


Figure 45. Using all available sequential access storage volumes with collocation enabled at the group level

Remember: Normally, the Tivoli Storage Manager server always writes data to the current filling volume for the operation that is running. Occasionally, however, you might notice more than one filling volume in a collocated storage pool. Having more than one filling volume in a collocated storage pool can occur if different

server processes or client sessions try to store data into the collocated pool at the same time. In this situation, Tivoli Storage Manager allocates a volume for each process or session that needs 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 46 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.

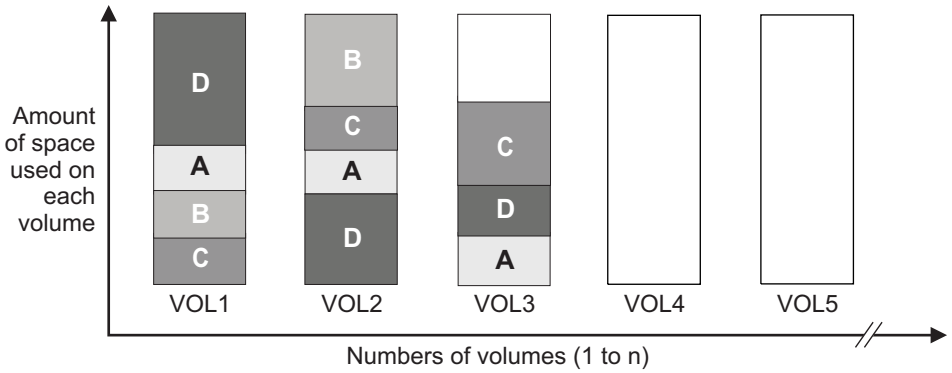


Figure 46. Using all available space on sequential volumes with collocation disabled

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 offsite reclamation activity.

Using collocation on copy storage pools and active-data pools requires special consideration.

Primary storage pools complete a different recovery role than the recovery role completed 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 offsite 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 helps 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, partially filled volumes might be unacceptable for copy storage pools and active-data pools whose storage pool volumes are taken offsite immediately. If you use collocation for copy storage pools or active-data pools, you must decide among the following:

- Taking more partially filled volumes offsite, which increases the reclamation activity when the reclamation threshold is lowered or reached. The 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 onsite until they fill and risk not having an offsite copy of the data on these volumes.
- Whether to collocate by group 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 your options carefully before you use collocation for copy storage pools and active-data pools, and whether to use simultaneous write. When not using simultaneous write, if you use collocation for your primary storage pools, you might 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 with each of them having large amounts of incremental backup data each day. For collocation with simultaneous write, you must ensure that the collocate settings are identical for the primary storage pools, active-data pools, and copy storage pools.

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.

About this task

Table 33 lists the four collocation options that you can specify on the **DEFINE STGPOOL** and **UPDATE STGPOOL** commands. The table also shows the effects of collocation on data that belongs to nodes that are and are not members of collocation groups.

Table 33. Collocation options and the 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 data for the node is not collocated.	The data for the node is not collocated.
Group	The server stores the data for the node 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 data for the node on as few volumes as possible.	The server stores the data for the node 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.

Table 34. Collocation group options and effects on file space data

Collocation option	If a file space is not defined as a member of a collocation group	If a file space is defined as a member of a collocation group
No	The data for the file space is not collocated.	The data for the file space is not collocated.
Group	The server stores the data for the filespace on as few volumes in the storage pool as possible.	The server stores the data for the file space and other file spaces that belong to the same collocation group on as few volumes as possible.
Node	The server stores the data for the node on as few volumes as possible.	The server stores the data for the node 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 file spaces 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, complete the following steps:

Procedure

1. Familiarize yourself with the potential advantages and disadvantages of collocation, in general. For a summary of effects of collocation on operations, see Table 31 on page 355.

2. If the decision is to collocate, determine how the data is to be organized, whether by client node, group of client nodes, or file space. If the decision is to collocate by group, you must decide how to group nodes:
   - If the goal is to save space, you might want 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 wanted result:
   a. Define collocation groups with the **DEFINE COLLOCGROUP** command.
   b. Add client nodes to the collocation groups with the **DEFINE COLLOCMEMBER** 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, see 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 with 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 with the **UPDATE COLLOCGROUP** command and delete entire collocation groups by issuing the **DELETE COLLOCGROUP** command.

Results

Tip: If you use collocation, but want to reduce the number of media mounts and use space on sequential volumes more efficiently, you can:

- 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 that belong 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 particular 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 details about the COLLOCATEBYFILESPEC option, see the *Backup-Archive Clients Installation and User's Guide*.

When creating collocation groups, keep in mind that the ultimate destination of the data that belongs to nodes in a collocation group depends on the policy domain to which nodes belong. For example, suppose that you create a collocation group that consists 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 which belongs 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.

Related reference:

"Managing client option files" on page 458

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.

About this task

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 that are 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.
- Storage pools that are defined with a CENTERA device class cannot be reclaimed.

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 47 on page 365 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 401 for details.

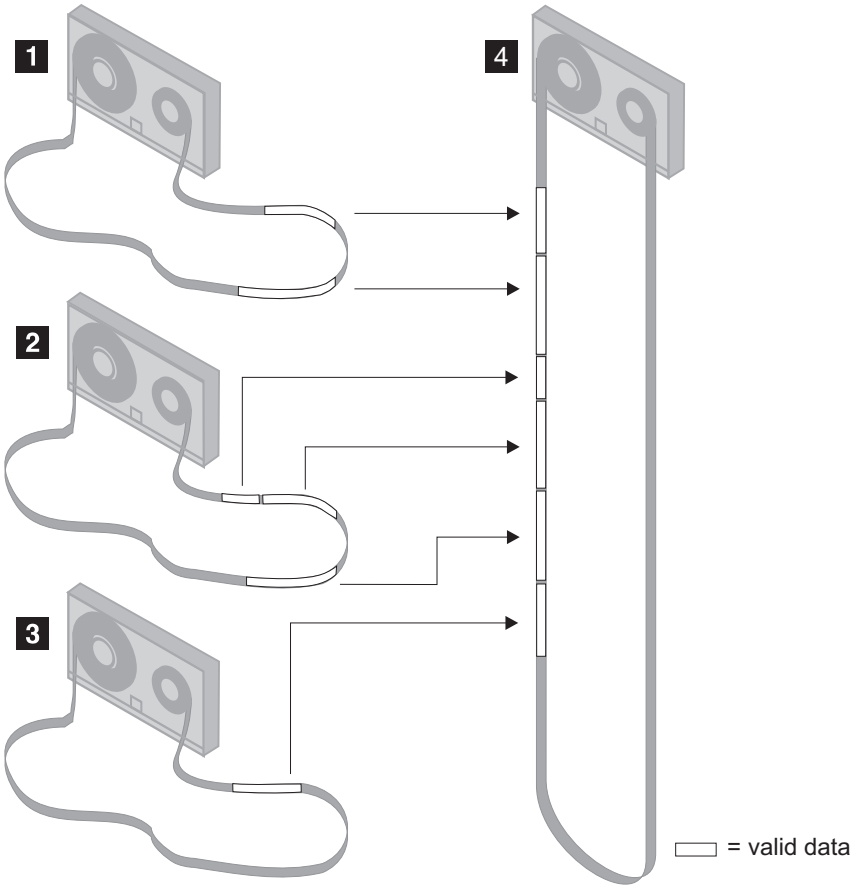


Figure 47. Tape reclamation

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 370 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 370.

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.

About this task

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 takes before it is automatically canceled.

About this task

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 ALTPOOL, 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 offsite 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*.

Restriction: Storage pools that are defined with a CENTERA device class cannot be reclaimed.

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.

About this task

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 178

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 374.

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.

About this task

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:

- 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.

About this task

To help reduce overall process time:

Procedure

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.

Reclamation of write-once, read-many (WORM) media

Reclamation of WORM volumes does not mean that you can reuse this write-once media. However, reclamation does allow you to make more library space available.

Reclamation of WORM volumes consolidates data from partially filled volumes to other WORM volumes. You can then eject the empty, used WORM volumes and add new volumes.

To allow reclamation, you can set the reclamation value to something lower when you define or update the storage pool.

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.

About this task

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:

Procedure

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.

Results

For information about using the SERVER device type, see "Using virtual volumes to store data on another server" on page 722.

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.

About this task

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 offsite volumes."

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 372.

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 367.

Reclamation of primary storage pool volumes does not affect copy storage pool files or files in active-data pools.

Reclamation of offsite volumes

Volumes with the access value of offsite 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 offsite volume is reclaimed, the files on the volume are rewritten to a read/write volume. Effectively, these files are moved back to the onsite location. The files might be obtained from the offsite 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 offsite volume.

The server reclaims an offsite 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 onsite volume of a copy storage 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 offsite volume was reclaimed.

   During offsite reclamation, active-data pools cannot be used to obtain data.

   For a single storage pool, the server reclaims all offsite 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 998.

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.

About this task

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:

Procedure

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.

Results

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.

About this task

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 722 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.

About this task

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 874.

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 367.

See also "Reducing the time to reclaim tape volumes with high capacity" on page 369.

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.

About this task

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 392.

Estimating space requirments 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.

About this task

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 272 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.

About this task

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:

```
Backup space = WkstSize * Utilization * VersionExpansion * 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.

About this task

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.

About this task

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 363 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 353 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 140

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 that are defined to the system.

About this task

To obtain a standard report, issue the following command:

```
query stgpool
```

Figure 48 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      Device       Estimated     Pct    Pct  High   Low  Next
Pool Name    Class Name    Capacity    Util   Migr   Mig   Mig  Storage
                                                    Pct   Pct  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 48. 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 sequential-access storage pools, estimated capacity is the total estimated space of all the sequential-access volumes in the storage pool, regardless of their access mode. At least one volume must be used in a sequential-access storage pool (either a scratch volume or a private volume) to calculate estimated capacity.

For tape and FILE, the estimated capacity for the storage pool includes the following factors:

- The capacity of all the scratch volumes that the storage pool already acquired or can acquire. The number of scratch volumes is defined by the **MAXSCRATCH** parameter on the **DEFINE STGPOOL** or **UPDATE STGPOOL** command.
- The capacity of all the private volumes that are defined to the storage pool when using the **DEFINE VOLUME** command.

The calculations for estimated capacity depend on the available space of the storage for the device that is assigned to the storage pool. For FILE storage pools, the capacity for the storage pool is reduced if the available storage falls below the total estimated space of all the FILE volumes in the storage pool. The value that is displayed for capacity is reduced by the size of a FILE volume incrementally as the available space continues to decline.

Tape 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 tape 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. This characteristic does not apply to FILE volumes in sequential-access storage pools.

Pct Util

Specifies, as a percentage, the space that is 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 that is 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 48 on page 378 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.

About this task

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.

About this task

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.

About this task

To request general information about all volumes defined to the server, enter:

```
query volume
```

Figure 49 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      Device      Estimated    Pct   Volume
                         Pool Name    Class Name   Capacity   Util   Status
-----------------------  -----------  ----------  ---------  -----  --------
/dev/raixvol1            AIXPOOL1     DISK         240.0 M    26.3  On-Line
/dev/raixvol2            AIXPOOL2     DISK         240.0 M    36.9  On-Line
/dev/rdosvol1            DOSPOOL1     DISK         240.0 M    72.2  On-Line
/dev/rdosvol2            DOSPOOL2     DISK         240.0 M    74.1  On-Line
/dev/ros2vol1            OS2POOL1     DISK         240.0 M    55.7  On-Line
/dev/ros2vol2            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 49. 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 50 on page 382 shows the output of this detailed query. Table 35 on page 382 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 50. Detailed information for a storage pool volume

Table 35. Using the detailed report for a volume

Task	Fields and Description
Ensure the volume is available.	*Volume Status* *Access*
	Check the *Volume Status* to see if a disk volume has been varied offline, or if a sequential-access volume is currently being filled with data. Check the *Access* to determine whether files can be read from or written to this volume.
Monitor the use of storage space.	*Estimated Capacity* *Pct Util*
	The *Estimated Capacity* 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 (*Pct Util*). In this example, 26.3% of the estimated capacity is currently in use.
Monitor the error status of the volume.	*Number of Write Errors* *Number of Read Errors*
	The server reports when the volume is in an error state and automatically updates the access mode of the volume to read-only. The *Number of Write Errors* and *Number of Read Errors* 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 875 for information about auditing a volume.

Table 35. 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.	*Scratch Volume?* *Write Pass Number* *Number of Times Mounted* *Approx. Date Last Written* *Approx. Date Last Read*
	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). In this example, WREN01 is a volume defined to the server by an administrator, not a scratch volume (*Scratch Volume?* is *No*). The *Write Pass Number* 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. 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. 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 396. Use the *Number of Times Mounted*, the *Approx. Date Last Written*, and the *Approx. Date Last Read* 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 875 for information about auditing a volume. The number given in the field, *Number of Times Mounted*, 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.
Determine the location of a volume in a sequential-access storage pool.	*Location*
	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).
Determine if a volume in a sequential-access storage pool is waiting for the reuse delay period to expire.	*Date Became Pending*
	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.

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 258 and "Obtaining information about the use of storage space" on page 392.

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.

About this task

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.

About this task

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 51 on page 386 displays a standard report which shows the first seven files from file space /usr on TOMC stored in WREN01.

```
Node Name               Type Filespace  Client's Name for File
                             Name
----------------------- ---- ---------- --------------------------------------
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 51. 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.

About this task

To display detailed information about the files stored on volume VOL1, enter:

```
query content vol1 format=detailed
```

Figure 52 on page 387 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\DSMCOMMN\ 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\DSMCOMMN\ 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\DSMCOMMN\ 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\DSMCOMMN\ AB0CTSYM.ENT
          Aggregated?: 4/4
          Stored Size: 202,927
       Segment Number: 1/1
         Cached Copy?: No
```

Figure 52. 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.

About this task

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.

About this task

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 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 that 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.

About this task

Figure 48 on page 378 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 53 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      Device       Estimated    Pct    Pct  High   Low  Next
Pool Name    Class Name    Capacity   Util   Migr   Mig   Mig  Storage
                                                    Pct   Pct  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 53. Information on backup storage pools

You can query the server to monitor the migration process by entering:

```
query process
```

A message similar to Figure 54 is displayed:

```
 Process Process Description      Status
  Number
-------- ------------------------ --------------------------------------------
       2 Migration                Disk Storage Pool BACKUPPOOL, Moved Files:
                                   1086, Moved Bytes: 25555579, Unreadable
                                   Files: 0, Unreadable Bytes: 0
```

Figure 54. 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.

About this task

First determine the identification number of the migration process by entering:

```
query process
```

A message similar to Figure 55 on page 390 is displayed:

```
 Process Process Description      Status
  Number
-------- ------------------------ --------------------------------------------
       1 Migration                ANR1113W Migration suspended for storage pool
                                   BACKUPPOOL - insufficient space in
                                   subordinate storage pool.
```

Figure 55. 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.

About this task

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 388, 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 388, 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.

About this task

In the example in "Example: Monitoring data migration between storage pools" on page 388, 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.

About this task

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 56 on page 392 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 56. 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.

About this task

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.

About this task

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 445.

Figure 57 on page 394 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.

```
                                                          Physical   Logical
Node Name        Type  Filespace    Storage      Number of      Space     Space
                       Name         Pool Name        Files   Occupied  Occupied
                                                             (MB)      (MB)
---------------  ----  -----------  -----------  ---------  ----------  --------
MIKE             Bkup  /home        ENGBACK1           513        3.52      3.01
```

Figure 57. 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   Physical
                                               Name           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.

About this task

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 58 on page 395 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/peas-e/dir	ARCHTAPE	492	18.40	18.40
PEASE	Bkup	/home/peas-e/dir	BACKTAPE	33	7.60	7.38
PEASE	Bkup	/home/peas-e/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 58. 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.

About this task

For example, to request a report about backup versions stored in the disk storage pool named BACKUPPOOL, enter:

```
query occupancy stgpool=backuppool type=backup
```

Figure 59 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 59. 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 59:

Procedure

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.

Results

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 375 and "Estimating space for archived files in random-access storage pools" on page 377.

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.

Procedure

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.

Example

Figure 60 displays the output for this command.

```
Device       Directory                    Estimated   Estimated
Class                                      Capacity   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 60. 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 onsite volumes or off-site volumes.

About this task

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

Task	Required Privilege Class
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.
- You cannot move data into or out of a storage pool that is defined with a CENTERA device class.
- 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.
- If you run the **MOVE DATA** command on an offsite volume that contains collocated data, it might be necessary to issue the **MOVE DATA** command multiple times to move all of the data out of the volume. Each **MOVE DATA** command moves the data for a single collocation group.

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 410 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 533 for more information about shredding.

Data movement from offsite volumes in copy storage pools or active-data pools

You can move data from offsite volumes without bringing the volumes onsite.

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 offsite volumes are copied from the original files in the primary-storage pools. In this way, valid files on offsite 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 that is marked as offsite, the server performs the following actions:

1. Determines which files still remain on the volume from which you are moving data
2. Obtains the remaining files from a primary-storage pool or from another copy-storage pool

3. Copies the files to one or more volumes in the destination copy-storage pool or active-data pool

During the data movement process, active-data pools cannot be used to obtain data.

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 you begin

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 254 for information about updating the access mode of a storage pool volume.
- Ensure that 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 380 for information about requesting information about a storage volume.
  2. Querying the specified destination storage pool to ensure that there is sufficient capacity to store the files that are being moved. See "Monitoring space available in a storage pool" on page 378 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 252 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 195
- If you are moving files from a tape volume to a tape storage pool, ensure that the two tape drives that are required are available.

Procedure

To move data, issue the **MOVE DATA** command.
For example, to move the files that are 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 might 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 DISCARDDATA=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.
- If you run the **MOVE DATA** command on an offsite volume that contains collocated data, it might be necessary to issue the **MOVE DATA** command multiple times to move all of the data out of the volume. Each **MOVE DATA** command moves the data for a single collocation group.

Requesting information about the data-movement process

You can query the server for statistics about the data-movement process.

About this task

To request information, issue the **QUERY PROCESS** command.

Figure 61 shows an example of the report that you receive about the data movement process.

```
 Process Process Description  Status
  Number
-------- -------------------- -------------------------------------------------
      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 61. 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.

About this task

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.

About this task

Procedure

To request information, use the QUERY VOLUME command.

Example

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      Device         Estimated      Pct     Volume
                    Pool Name    Class Name      Capacity     Util     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      Device         Estimated      Pct     Volume
                    Pool Name    Class Name      Capacity     Util     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      Device         Estimated      Pct     Volume
                    Pool Name    Class Name      Capacity     Util     Status
----------------    ----------   ----------     ---------    -----    --------
/dev/vol3           BACKUPPOOL   DISK             15.0 M       0.0    On-Line
```

Moving data belonging to a client node

You can move data 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 in either a primary storage pool, copy storage pool, or active-data pool.

About this task

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 that you are moving. That is, the server cannot use a destination volume that contains data that needs to be moved.
- When you move 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.
- You cannot move node data into or out of a storage pool that is defined with a CENTERA device class.

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.

About this task

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.

About this task

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 ENGPOOL 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$ unifilespace=\\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.

About this task

To request information on the data movement process, enter:

```
query process
```

Figure 62 shows an example of the report that you receive about the data movement process.

```
 Process    Process Description   Status
  Number
--------    --------------------  -------------------------------------------------
       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 62. 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.

About this task

The most common reasons are:

- Files are 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 884.

- Files in the source storage pool reside on volumes whose access mode is offsite, destroyed, or unavailable. To complete the move operation, bring the volumes onsite, 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

Note: If you run the **MOVE DATA** command on an offsite volume that contains collocated data, it might be necessary to issue the **MOVE DATA** command multiple times to move all of the data out of the volume. Each **MOVE DATA** command moves the data for a single collocation group.

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.

About this task

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 695.

Defining copy storage pools and active-data pools

Use a copy storage pool or an active-data pool to back up one or more primary storage pools. When you definecopy storage pools or active-data pools, you can take advantage of various Tivoli Storage Manager functions by specifying certain properties.

About this task

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 36.

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.
2. You cannot define copy storage pools or active-data pools for a Centera device class.

Table 36. 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.
Access mode	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: **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. However, system processes can move files within the volumes in the storage pool. **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.
Maximum number of scratch volumes	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. 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 148
Collocation	When collocation is enabled, the server tries to keep all files on a minimal number of sequential-access storage volumes. The files can belong to a single client node, a group of client nodes, a client file space, or a group of file spaces. See "Collocation of copy storage pools and active-data pools" on page 360.

Table 36. Information for defining copy storage pools and active-data pools (continued)

Information	Explanation
Reclamation threshold	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. 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 onsite. 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 370 and "Controlling reclamation of virtual volumes" on page 370 for more details.
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 "Delayed reuse of reclaimed volumes" on page 374.
Off-site reclaim limit	Specifies the number of off-site volumes to be reclaimed during reclamation for a storage pool. See "Reclamation of offsite volumes" on page 371.
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 370.

For more information, see "Backing up primary storage pools" on page 871.

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.

About this task

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 primary storage pools" on page 871.

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 37 compares the characteristics of primary, copy-storage, and active-data pools.

Table 37. 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
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	Within the same storage pool only. • 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.	Within the same storage pool only. • 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

Table 37. Comparing primary-storage pools, copy-storage pools, and active-data pools (continued)

Characteristic	Primary-storage pool	Copy-storage pool	Active-data pool
Reclamation	Yes (sequential-access storage pools only)	Yes Virtual volumes (volumes with device type SERVER) and off-site volumes are handled differently. For details, see "Controlling reclamation of virtual volumes" on page 370 and "Reclamation of offsite volumes" on page 371.	Yes Virtual volumes (volumes with device type SERVER) and off-site volumes are handled differently. For details, see "Controlling reclamation of virtual volumes" on page 370 and "Reclamation of offsite volumes" on page 371.
File deletion	Files are deleted: • 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	Files are deleted: • 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	Files are deleted: • 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.

About this task

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 411 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 378.

  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 definition

  See "Defining storage pools" on page 241 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.

About this task

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 are 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

You cannot delete a Centera volume if the data in the volume was stored using a server with retention protection enabled and if the data has not expired.

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

Task	Required Privilege Class
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.

About this task

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 533 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.

About this task

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 note deleted until the data on it is shredded. See “Securing sensitive client data” on page 533 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 396 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.

© Copyright IBM Corp. 1993, 2013

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 417

"Overview of clients and servers as nodes"

Related tasks:

"Installing client node software" on page 416

"Registering nodes with the server" on page 416

Related reference:

"Connecting nodes with the server" on page 421

"Comparing network-attached nodes to local nodes" on page 423

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.

© Copyright IBM Corp. 1993, 2013

Installing client node software

Administrators can install backup-archive clients, application clients, or Tivoli Storage Manager for Space Management clients using three different methods.

About this task

The following are the 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

You can also install using the silent installation technique.

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.

About this task

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: You can connect to a Web backup-archive client directly from a supported Web browser or from a hyperlink in the Web administrative Enterprise Console. To do so, specify the node's URL and port number during the registration process or update the node later with this information.

Related concepts:

"Overview of remote access to web backup-archive clients" on page 439

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.

The administrator must register client nodes with the server when registration is set to closed. Closed registration is the default.

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 issue the **REGISTER NODE** command to register the node and specify the initial password.

About this task

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

The server prompts the user, on their first attempt to connect to the server, for a node name, password, and contact information. That information is added to a node during open registration.

About this task

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 in server storage. Nodes are created with the default authentication method that is defined on the server. Nodes are registered with the default authentication method if it is defined on the server with the **SET DEFAULTAUTHENTICATION** command. The default is LOCAL.

Procedure

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, see 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.

About this task

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.

Remember: Use either client compression or drive compression, but not both.

Related concepts:

"Data compression" on page 197

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.

About this task

Client option sets are considered advanced implementation.

Procedure

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 458

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.

About this task

The **REGISTER NODE** and **UPDATE NODE** commands have a default parameter of TYPE=CLIENT.

Procedure

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 9, "Using NDMP for operations with NAS file servers," on page 199

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.

About this task

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.

Procedure

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 722

Registering an API to the server

Workstation users can request IBM Tivoli Storage Manager services by using an application that uses the Tivoli Storage Manager application programming interface (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 IBM Tivoli Storage Manager application programming interface (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 422

Required client options

Each node requires a client options file. Each client options file must contain the network address of the IBM Tivoli Storage Manager server and other communication options that allow the node to communicate with the server.

Figure 63 on page 422 shows the contents of a client options file that is configured to connect to the server using TCP/IP. The communication options specified in the client options file satisfy the minimum requirements for the node to connect to the server.

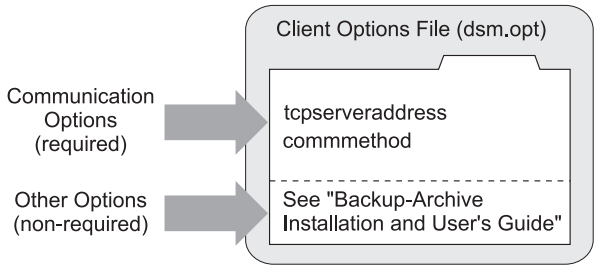


Figure 63. Client options file

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 64.

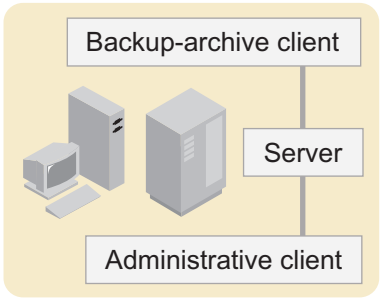


Figure 64. Stand-alone environment

Figure 65 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.

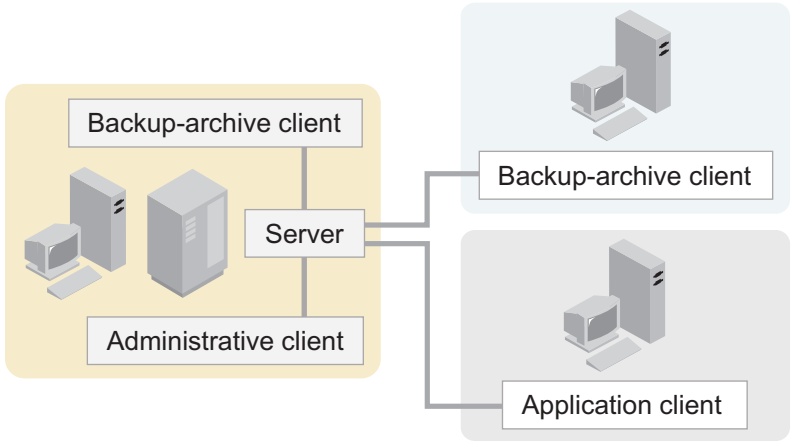


Figure 65. Network environment

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 the *Administrator's Reference*.

Enabling open registration

The default registration mode at installation is closed.

Procedure

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.

About this task

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.

Procedure

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 13, "Implementing policies for client data," on page 469

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 417

"Overview of clients and servers as nodes" on page 415

Related tasks:

"Installing client node software" on page 416

"Registering nodes with the server" on page 416

Related reference:

"Connecting nodes with the server" on page 421

"Comparing network-attached nodes to local nodes" on page 423

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 445

© Copyright IBM Corp. 1993, 2013

"Comparing network-attached nodes to local nodes" on page 423

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.

Procedure

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.

Results

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 (ACSLS)
- Simple network management protocol (SNMP)
- Centera device support
- Shared memory protocol
- Windows Microsoft Management Console functions
- Tivoli Enterprise Console® support

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

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, specify the **SESSIONINITIATION** parameter on the server. You must also synchronize the information in the client option file.

About this task

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.

Restriction: Server-initiated sessions cannot utilize SSL (Secure Sockets Layer) or LDAP authenticated passwords.

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.
3. Server-initiated sessions cannot utilize SSL (Secure Sockets Layer) or LDAP authenticated passwords.

Table 38. 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.

Procedure

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
```

Command-line backup-archive client automatic deployment feature: overview

The backup-archive client automatic deployment feature helps you schedule updates to one or more backup-archive clients with deployment packages. The deployment packages can be backup-archive client updates or full releases.

The backup-archive client deployment feature can simultaneously update multiple backup-archive clients, but it does not install new backup-archive clients. The deployment is configured and scheduled from the administrative command line. For more information about the client deployment feature, see Technote 1653433 at: http://www.ibm.com/support/docview.wss?uid=swg21653433.

The following list shows the deployment restrictions:

- Tivoli Storage Manager backup-archive clients at levels earlier than 5.5 that are on other than Windows platforms 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.
- Do not schedule automatic client deployments to workstations that have an IBM Tivoli Storage Manager for Enterprise Resource Planning application installed on them.

  At deployment time, a dependency check is performed to ensure that the API library does not conflict with the client package that is currently installed. IBM Tivoli Storage Manager for Enterprise Resource Planning applications do not use the same installation technology that the client installation program uses. The API library that is installed might not be compatible with the API library that was installed by the IBM Tivoli Storage Manager for Enterprise Resource Planning application. The newly deployed API library can cause the IBM Tivoli Storage Manager for Enterprise Resource Planning applications to fail.

To use the feature, the backup-archive client must meet these requirements:

- The `PASSWORDACCESS` option must be set to `generate`.
- The client acceptor (CAD) or backup-archive client schedule must be running at the time of the deployment. The backup-archive client is deployed from the server as a scheduled task.
- The backup-archive client must meet the following disk space requirements:

Table 39. Disk space required to deploy a backup-archive client package

Operating system	Total required disk space
AIX	1500 MB
Solaris	1200 MB
HP-UX	900 MB
Mac OS X	200 MB
Linux x86/x86 64	950 MB
Windows	2 GB

Important: You must use the **SET SERVERHLADDRESS** command for all automatic client deployments.

You can find the deployment packages in the maintenance directory on the FTP site: ftp://public.dhe.ibm.com/storage/tivoli-storage-management/maintenance/client.

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 maxscratch=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. The default for the Administration Center is five years.

  Important: The retention value must be set to a value that includes the amount of time that the package was on the FTP site. For example, if a deployment package is on the FTP site for 30 days, the retention value for the copy group must be greater than 30 days. If not, the package expires when the next **EXPIRE INVENTORY** command is issued.
- *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

The backup-archive client deployment packages must be imported to the servers after you acquire them from the FTP site.

Before you begin

Ensure that you configure the server for backup-archive client automatic deployments before you import the packages.

Procedure

Complete the following steps to import deployment packages to the server:

1. Download the backup-archive client deployment packages. If you are working outside of the Administration Center, you can download the packages from the FTP site: ftp://public.dhe.ibm.com/storage/tivoli-storage-management/maintenance/client. Download the client deployment packages from the FTP site to the Tivoli Storage Manager server or to an internal FTP site.
2. Issue the **IMPORT** command. For example:

   ```
   import node  IBM_CLIENT_DEPLOY_UNX devclass=upgradedev
   filedata=archive mergefilespace=yes replace=all
   volumenames=volname1.exp
   ```

   For Macintosh, issue the following command:

   ```
   import node  IBM_CLIENT_DEPLOY_MAC devclass=upgradedev
   filedata=archive mergefilespace=yes replace=all
   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.
3. 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_UNX'
   ```

   For Macintosh, issue the following command:

   ```
   select * from ARCHIVES where node_name='IBM_CLIEN_DEPLOY_MAC'
   ```

   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 429

Defining a schedule for an automatic deployment

You can schedule a backup-archive client deployment after you import the packages to the server.

About this task

Complete the following steps to create a schedule for an automatic client deployment:

Procedure

1. Issue the **DEFINE SCHEDULE** command. For example:

   Note: The **-fromnode** option can have the following values:

   IBM_CLIENT_DEPLOY_WIN

   IBM_CLIENT_DEPLOY_UNX

   IBM_CLIENT_DEPLOY_MAC

   ```
   define schedule domain_name schedule_name action=deploy
   objects='deployment_package_location destination_for_package duration=1
   durunits=hours perunits=onetime startdate=today starttime=now
   options='-fromnode=IBM_CLIENT_DEPLOY_UNX
   ```

```
-postnschedulecmd='relative_pathdeployclient.sh>
/opt/tivoli/tsm/client/IBM_ANR_UNX/v63011/upminx#schedule=schedule_name
#domain=domain_name#nodeinfo2=TBD'-subdir=yes -replace=all'
```

where

deployment_package_location is the path to the deployment package

destination_for_package is the path to where you want to store the deployment package

IBM_CLIENT_DEPLOY_UNX is the predefined name (for a UNIX deployment) for the **-fromnode** option

nodeinfo2=TBD must be entered exactly as shown.

If your current backup-archive client is AIX, Linux, Solaris, or HP-UX and is at version 6.1 or later, use `nodeinfo2=TBD` in the **POSTNSCHEDULECMD** command. Macintosh Backup-Archive Clients at version 5.5 and later also must use `nodeinfo2=TBD`.

Important: Not all backup-archive clients can process post-schedule commands that contain spaces. Use the number sign (#) in place of blank spaces in the **-postnschedulecmd** command. If your command fails and a log file is not available, reissue the command and use the number sign instead of spaces.

The target location for backup-archive client packages on Macintosh operating systems is in relation to the location of the working client scheduler directory. This location is also the deployment manager's working directory. Here is an example post-schedule command for Macintosh:

```
-postnschedulecmd='../relative_path/deployclient.sh>
./IBM_ANR_MAC/V6303/upmin#schedule=schedule_name#domain=standard#nodeinfo2=TBD'
```

For operating systems that are not Windows, the locations are determined by the default installation directory of the backup-archive client. You can specify a different directory from the Administration Center. For example, on operating systems such as AIX, the default, working directory is `/usr/tivoli/tsm/client/IBM_ANR_UNX/VXXXX`. On operating systems such as UNIX and Linux, you can replace the "`/usr`" file path with one of the following names:

- `/tmp`
- `/opt`
- `/home`

Important: If you use **AUTODEPLOY=NOREBOOT** in your command, the client operating system does not restart after the deployment completes. Restarting can affect any critical applications that are running on the client operating system. Applications that are not Tivoli Storage Manager applications that are 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 **DEFINE ASSOCIATION** command to associate nodes with the deployment schedule.

Verifying the server for backup-archive client automatic deployments

You can verify that the server is configured to automatically update the backup-archive clients.

About this task

You can verify that the server is configured to manage deployment packages for the backup-archive client.

Procedure

Complete the following steps to verify that the server is configured to manage deployment packages:

1. Verify that the IBM_CLIENT_DEPLOY_IMPORT device class is defined by issuing `QUERY DEVCLASS IBM_CLIENT_DEPLOY_IMPORT F=D`.
2. Verify that a storage pool is defined to store the deployment packages on the server by issuing `QUERY STGPOOL stgpool_name F=D`.
3. Verify that an archive copy group is defined by issuing `QUERY COPYGROUP IBM_CLIENT_DEPLOY IBM_CLIENT_DEPLOY IBM_CLIENT_DEPLOY TYPE=ARCHIVE`.
4. Verify that the default management class is set to the active policy set by issuing `QUERY POLICYSET IBM_CLIENT_DEPLOY F=D`.
5. Verify that the schedule is created for your client nodes by issuing `QUERY EVENT domain_name sched_name F=D`.
6. Verify that the `targetlevel` is correctly updated to match the level of the deployment package by issuing `QUERY NODE node_name F=D`.

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.

About this task

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.

Procedure

To rename CAROLH to ENGNODE, issue the following example command:

```
rename node carolh engnode
```

ENGNODE retains the contact information and access to back up and archive data that belonged to CAROLH. All files backed up or archived by CAROLH now belong to ENGNODE.

Results

If you rename a node that authenticates with an LDAP directory server, names for same-named nodes on other servers that share namespace are not renamed. You must issue a **RENAME** command for each node. If you want to keep the nodes in sync, change their name to match the new name. If you do not, the node on the

other server can no longer authenticate with the LDAP directory server if you specify **SYNCLDAPDELETE=YES**.

Rename all nodes *starship* that authenticate to LDAP directory servers

If you have a node that shares namespace on an LDAP directory server with other nodes, you can rename each node. The renaming must, however, be done on each server. For example, you can issue the following command on each server:

```
rename node starship moonship syncldapdelete=yes
```

The node *starship*, that authenticates to an LDAP directory server, changes their name to *moonship*. With **SYNCLDAPDELETE=YES**, the entry on the LDAP directory server changes to *moonship* and removes node *starship* from the LDAP server. Therefore, other servers cannot authenticate node *starship* with the LDAP server. You can register node *starship* with the LDAP server, or rename node *starship* to *moonship*.

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.

About this task

You can restore a locked node's access to the server with the **UNLOCK NODE** command.

Procedure

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 465

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.

Before you begin

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.

Procedure

Remove client node DEBBYG by completing the following example commands:

1. 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 457

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.

About this task

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.

Procedure

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 myhiddp3as
   ```

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 that are assigned to specific policy domains.

Procedure

Issue the following command to view information about client nodes that are assigned to the STANDARD and ENGPOLDOM policy domains:

```
query node * domain=standard,engpoldom
```

The data from that command might display similar to the following output:

```
Node Name    Platform     Policy Domain   Days Since   Days Since   Locked?
                          Name                 Last     Password
                                           Access          Set
----------   --------     --------------  ----------   ----------   -------
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      SOLARIS     ENGPOLDOM               <1            1    No
```

Displaying information about a specific client node

You can view information about specific client nodes.

Procedure

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.

Restriction: Administrators who authenticate their password with an LDAP directory server cannot access the web client.

A web backup-archive client can be accessed from a web browser or opened from the Operations 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.

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 that they must 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 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 Tivoli Storage Manager backup-archive client computer that is running the web client. The browser that 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://www.oracle.com/.

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 exists with the same name as the node that is 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 443

Granting client authority

You might have to grant client authority to a user at some time.

Procedure

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.

About this task

When the node is created, the authentication method and Secure Sockets Layer (SSL) settings are inherited by the administrator.

Procedure

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.

Procedure

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.

Procedure

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.

About this task

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.

Procedure

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.

Results

For information about what functions are not supported on the Web backup-archive client, refer to *Backup-Archive Clients Installation and User's Guide*.

Collecting processor value unit information in a VMware host environment

If you are installing Tivoli Storage Manager on a system that uses VMware virtualization software, ensure that PVU information is collected in the virtual environment. To collect PVU information in a virtual environment, you must have VMware ESX 4 software installed on your system, and you must set up and run hardware scans.

Procedure

Complete the following steps:

1. Obtain the Tivoli Storage Manager backup-archive client installation package for the Linux x86 operating system.

   For information about obtaining installation packages, see the Passport Advantage® website at http://www.ibm.com/software/lotus/passportadvantage/pacustomers.html.
2. Install the Tivoli Storage Manager backup-archive client on a computer in your system.
3. From the `../CIT/VMware/esx-4` directory, download the `dispatcher`, `retrieve.sh`, `wenvmw.sh`, and `cpuid` files.
4. Copy the downloaded files to the ESX 4 host operating system.

   Tip: You can copy the files to any location on the host operating system, but ensure that all files are copied to the same directory.
5. Ensure that guest virtual machines are running. This step is necessary to ensure that the guest virtual machines are detected during the hardware scan.
6. To collect PVU information, issue the following command:

   ```
   retrieve -v
   ```

What to do next

If you restart the host machine or change the configuration, run the **retrieve** command again to ensure that current information is retrieved.

Tip: When the IBM Tivoli Storage Manager for Virtual Environments license file is installed on a VMware vStorage backup server, the platform string that is stored on the Tivoli Storage Manager server is set to TDP VMware for any node name that is used on the server. The reason is that the server is licensed for Tivoli Storage Manager for Virtual Environments. The TDP VMware platform string can be used for PVU calculations. If a node is used to back up the server with standard backup-archive client functions, such as file-level and image backup, interpret the TDP VMware platform string as a backup-archive client for PVU calculations.

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
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 `VIRTUALMOINTPOINT` 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 `VIRTUALMOUNTPOINT` option, refer to the appropriate *Backup-Archive Clients Installation and User's Guide*.

Defining object names for clients

A fully qualified name for a client object can refer to a full file name and path that is combined, or to the full directory structure for the object.

About this task

For client nodes that are running on Windows, it is possible to create objects with long fully qualified names. The IBM Tivoli Storage Manager clients for Windows are able to support fully qualified names of up to 8704 bytes in length for backup and restore functions. These long names are often generated with an automatic naming function or are assigned by an application.

Important: The Unicode representation of a character can occupy several bytes, so the maximum number of characters that a fully qualified name might contain can vary. See *Setting options in the client options file* in the *Backup-Archive Clients Installation and User's Guide* for Windows for a full explanation of file path names and limits.

Long object names can be difficult to display and use through normal operating system facilities, such as a command prompt window or Windows Explorer. To manage them, Tivoli Storage Manager assigns an identifying token to the name and abbreviates the length. The token ID is then used to display the full object name. For example, an error message might display as follows, where [TSMOBJ:9.1.2084] is the assigned token ID:

```
ANR9999D file.c(1999) Error handling file [TSMOBJ:9.1.2084] because of
lack of server resources.
```

The token ID can then be used to display the fully qualified object name by specifying it in the **DISPLAY OBJNAME** command.

Procedure

Issue the **DISPLAY OBJNAME** command with the token ID [TSMOBJ:9.1.2084]:

```
display objname 9.1.2084
ANR0160I Displaying segment 1 of 2 for client object  9.1.2084 with text:
\L08000-100\SUBDIRECTORY00000000000000000000000000000000000000000000000000000000
00000000000000000000000000002\SUBDIRECTORY00000000000000000000000000000000000000
00000000000000000000000000000000000000000000000003\SUBDIRECTORY000000000000000000
00000000000000000000000000000000000000000000000000000000000004\SUBDIRECTORY
```

```
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000005\SUBDIRECTORY000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000006\SUBDIRECTORY000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000007\SUBDIRECTORY000000000000000000000000
00000000000000000000000000000000000000000000000000000000000008\SUBDIRECTORY00000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
9\SUBDIRECTORY000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000010\SUBDIRECTORY000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000011\SUB.
ANR0160I Displaying segment 2 of 2 for client object  9.1.2084 with text:
DIRECTORY000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000012\SUBDIRECTORY000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000013\SUBDIRECTORY000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000014\SUBDIRECTORY000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000015\SUBDIREC
TORY000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000016\SUBDIRECTORY000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000017\SUBDIRECTORY000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000018\SUBDIRECTORY00000000000000000000
00000000000000000000000000000000000000000000000000000000000000000019\FILE001.
```

The fully qualified object name is displayed. If you are displaying long object names that are included in backup sets, a token ID might not be included if the entries for the path no longer exist in the Tivoli Storage Manager server database. You cannot 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 about fully qualified object names and issuing 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 456

Related tasks:

"Migrating clients to Unicode-enabled file spaces" on page 449

Related reference:

"Reasons for migrating clients to Unicode-enabled file spaces"

"Querying Unicode-enabled file spaces" on page 456

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.

About this task

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.

Procedure

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

Results

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 452

"Examining issues when migrating to Unicode" on page 453

"Example of a migration process" on page 454

Related reference:

"Defining options for automatically renaming file spaces" on page 450

"Defining the rules for automatically renaming file spaces" on page 451

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 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).

  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. 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. Therefore, a client can have mixed file spaces.

  Attention: If you force the renaming of the file space for all clients at the same time, client operations can contend for network and storage resources. Also, 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 or whether the user is prompted for renaming during a 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, the file spaces might be renamed. If the client runs a 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 one time 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 to create a new Unicode-enabled file space. The file space

is created the next time that 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. A client session can then run unattended. The prompt is displayed during the next interactive session on the client.

The following table summarizes what occurs with different parameter and option settings.

Table 40. 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
Client	No	Not renamed	Yes
Client	Prompt	Command-line or GUI: The user receives a one-time-only prompt about renaming	Depends on the response from the user (yes or no)
Client	Prompt	Client Scheduler: Not renamed (prompt is displayed during the next command-line or GUI session)	No

Related reference:

"Defining the rules for automatically renaming file spaces"

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
\\maria\c$	\\maria\c$_OLD

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
\\maria\c$	\\maria\c$_OLD	\\maria\c$_OLD1 \\maria\c$_OLD2

If the new name for the file space exceeds the limit of 64 characters, the file space name is truncated 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.

About this task

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.

Procedure

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 on page 452 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 642

Examining issues when migrating to Unicode:

When you migrate to Unicode, there are several issues that you must consider.

About this task

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.

About this task

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:

Procedure

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 568

Related reference:

"Managing the renamed file spaces"

"Defining the rules for automatically renaming file spaces" on page 451

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
```

The result is similar to the following output:

```
Node Name  Filespace   FSID  Platform  Filespace    Is     Capacity   Pct
           Name                        Type      Filespace     (MB)  Util
                                                 Unicode?
---------- ----------- ----  --------  --------- --------- -------- -----
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"

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 542

Displaying information about file spaces

You can display file space information by identifying the client node name and file space name.

About this task

You can display file space information for the following reasons:

- To identify file spaces that are 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 that is 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.

Procedure

To view information about file spaces that are defined for client node JOE, issue the following command:

```
query filespace 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   FSID  Platform  Filespace    Is     Capacity   Pct
           Name                        Type      Filespace     (MB)  Util
                                                 Unicode?
---------- ----------- ----  --------  --------- --------- -------- -----
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 might 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 Operations 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 402

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.

About this task

Administrators might 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.

Procedure

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, the user 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 that belong 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.

Results

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 that belongs 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 that are deleted from the file space are those that meet the retention criterion. The file space is not deleted if one or more files within the file space cannot be deleted.

Note: Data stored using the System Storage Archive Manager product cannot be deleted using the **DELETE FILESPACE** command if the retention period for the data has not expired. If this data is stored in a Centera storage pool, then it is additionally protected from deletion by the retention protection feature of the Centera storage device.

Related concepts:

"Accepting default closed registration or enabling open registration" on page 417

Related tasks:

"Deleting client nodes" on page 434

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" on page 459

"Managing client option sets" on page 461

Related reference:

"Connecting nodes with the server" on page 421

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.

About this task

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.

Procedure

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 421

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.

Procedure

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.

Procedure

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 client options that you can specify, refer to *Administrative client options* in 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 last in 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 first sequence number. The client can issue the **QUERY INCLEXCL** command to show the include-exclude statements in the order that 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 482

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.

Procedure

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.

Procedure

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.

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 clopset 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

Related concepts:

"Managing client restartable restore sessions" on page 465

Displaying information about IBM Tivoli Storage Manager sessions

Each client session is assigned a unique session number.

Procedure

To display information about client sessions, issue the following command:

```
query session
```

Figure 66 shows a sample client session report.
You can determine the state of the server by examining the session state and wait

```
  Sess Comm.  Sess     Wait   Bytes   Bytes Sess  Platform Client Name
Number Method State    Time    Sent   Recvd Type
------ ------ ------ ------ ------- ------- ----- -------- -----------
   471 Tcp/Ip IdleW   36 S      592     186 Node  WinNT    JOEUSER
   472 Tcp/Ip RecvW    0 S      730     638 Node  WinNT    STATION1
```

Figure 66. 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 363

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.

About this task

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.

Procedure

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
```

Results

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 462

Related reference:

"Server session states" on page 462

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.

About this task

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

Procedure

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 434

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 stopped, 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.

Procedure

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.

About this task

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.

Procedure

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.

About this task

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.

Procedure

Issue the following command to cancel a session:

```
cancel session -2
```

Session -2 cannot be restarted after you issue this command.

Chapter 13. 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 might 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.

Related concepts:

"More on management classes" on page 480

"Protection and expiration of archive data" on page 508

"Distributing policy using enterprise configuration" on page 522

Related tasks:

"Client operations controlled by policy" on page 474

"Getting users started" on page 472

"Changing policy" on page 472

"Assigning client nodes to a policy domain" on page 506

"Policy configuration scenarios" on page 516

"Configuring policy for direct-to-tape backups" on page 516

Related reference:

"Basic policy planning" on page 470

"Reviewing the standard policy" on page 471

"File expiration and expiration processing" on page 473

"The parts of a policy" on page 476

"How Tivoli Storage Manager selects files for policy operations" on page 486

"Creating your own policies" on page 490

© Copyright IBM Corp. 1993, 2013

"Defining and updating a policy domain" on page 492
"Defining and updating a management class" on page 495
"Validating and activating a policy set" on page 504
"Running expiration processing to delete expired files" on page 506
"Configuring policy for Tivoli Storage Manager application clients" on page 517
"Policy for logical volume backups" on page 517
"Configuring policy for NDMP operations" on page 518
"Configuring policy for LAN-free data movement" on page 519
"Policy for Tivoli Storage Manager servers as clients" on page 521
"Setting policy to enable point-in-time restore for clients" on page 522
"Querying policy" on page 523
"Deleting policy" on page 525

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 473
"Running expiration processing to delete expired files" on page 506
"Reviewing the standard policy" on page 471

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 41. 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** **RETONLY**
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
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 476

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.

About this task

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 459

Chapter 15, "Scheduling operations for client nodes," on page 553

Related reference:

"The include-exclude list" on page 482

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.

About this task

Other situations may also require policy changes. See "Policy configuration scenarios" on page 516 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.

Procedure

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 502

"Policy configuration scenarios" on page 516

Related reference:

"Validating a policy set" on page 504

"Activating a policy set" on page 505

"Defining and updating a management class" on page 495

"Defining and updating a backup copy group" on page 496

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 465

"Deletion hold" on page 509
"Expiration processing of base files and subfiles" on page 548
Related tasks:
"Reclaiming space in sequential-access storage pools" on page 363
Related reference:
"Running expiration processing to delete expired files" on page 506

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 475
"Client migration and recall" on page 475

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 517

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.

Restriction: If you back up or archive data with a Tivoli Storage Manager V6.3 client, you cannot restore or retrieve that data with a V6.2 or earlier client.

Related reference:

"Setting policy to enable point-in-time restore for clients" on page 522

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

With aTivoli Storage Manager HSM product, a user can migrate files from workstation storage to server storage and recall those files as needed.

The HSM client frees space for new data and makes more efficient use of your storage resources.

For details about using Tivoli Storage Manager for Space Management, see *Space Management for UNIX and Linux User's Guide* and *Tivoli Storage Manager HSM for Windows 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:

"Selection for active file versions" on page 239

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 67 on page 477 shows the parts of a policy and the relationships among the parts.

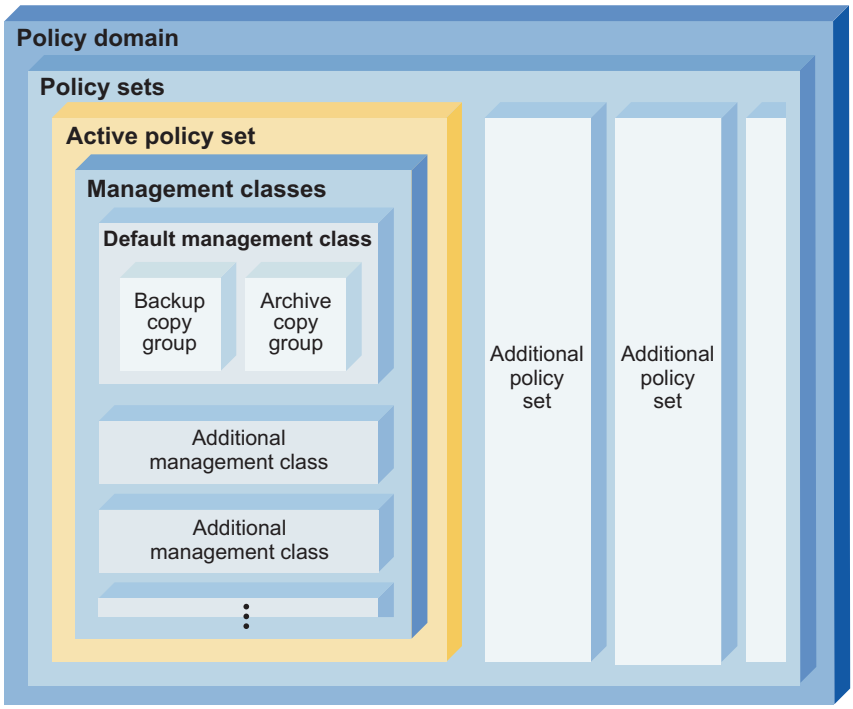


Figure 67. IBM Tivoli Storage Manager Policy

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 480

Related reference:

"Example: sample policy objects" on page 491

"Running expiration processing to delete expired files" on page 506

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.

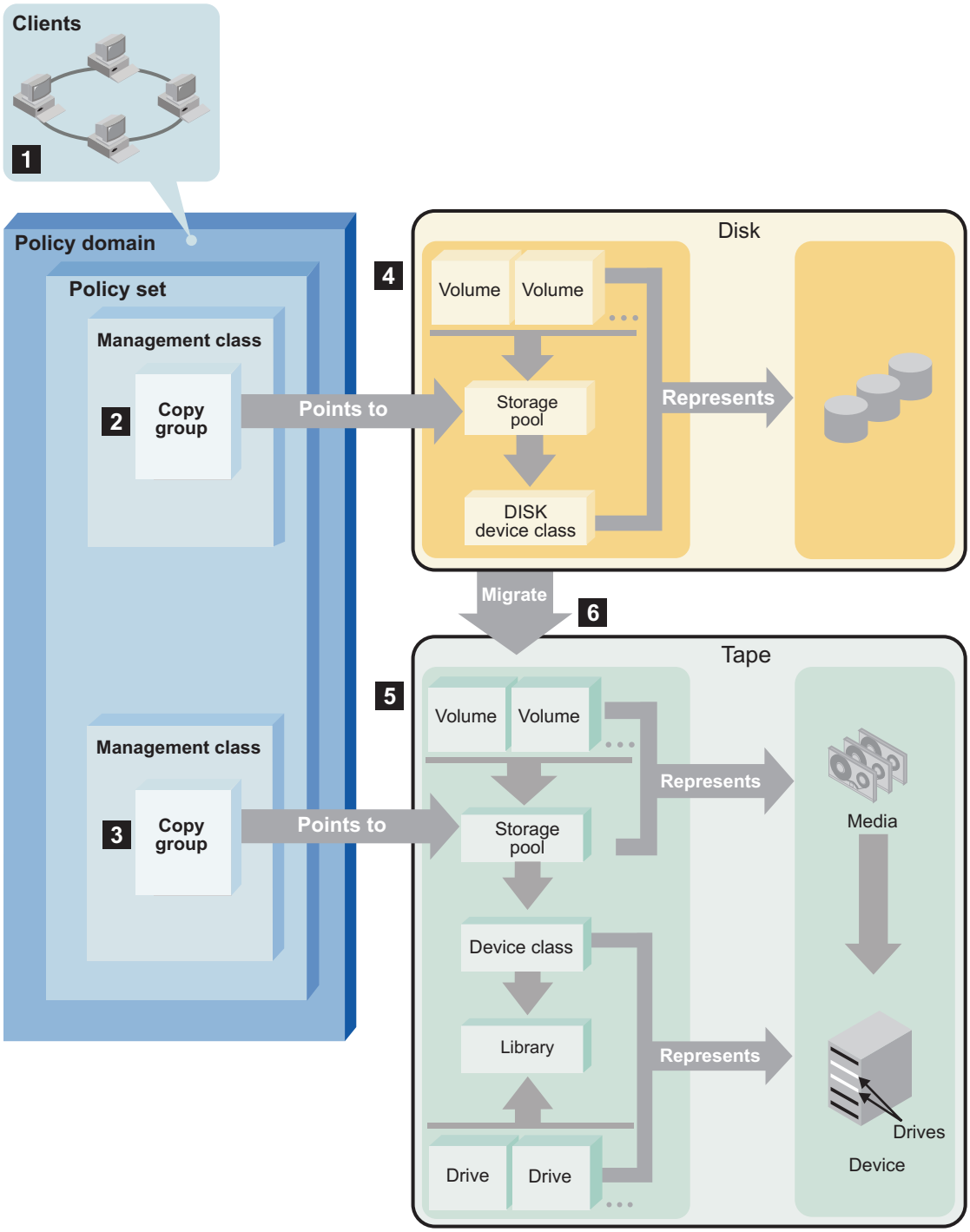


Figure 68. How clients, server storage, and policy work together

1 When clients are registered, they are associated with a policy domain. In a policy domain, there is a 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 its backup and archive copy groups, specifies where files are stored, and how they are managed.

4 , **5**

Storage pools are a collection of volumes where backed-up, archived, or

space-managed files are stored. The copy groups of a management class specify storage pools for backed-up or archived files. For space-managed files, the management classes specify the storage pools.

Storage pools are mapped to device classes, which represent devices. A storage pool contains volumes of the media-type indicated by the associated device class. The following example illustrates this concept:

- A storage pool that is mapped to a device class with a device type of `8MM` contains only 8 mm tapes.

6

Files that are initially stored on disk storage pools can migrate to the following storage pool if the pools are set up in a storage hierarchy:

- Tape storage pools

Figure 68 on page 479 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 416

Related reference:

"Contents of a management class"

"Default management classes" on page 481

"The include-exclude list" on page 482

"How files and directories are associated with a management class" on page 483

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 483

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 69 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 69 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 69 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 69 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 69. 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 459

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.

Procedure

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 506

Rebinding files to management classes

Rebinding is the process of associating all versions of 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.

The most recently backed up files are *active backup versions*. Older copies of your backed up files are *inactive backup versions*. You can configure management classes to save a predetermined number of copies of a file. If a management class is saving five backup copies, there would be one active copy saved and four inactive copies saved. If a file from one management class is bound to a different management class that retains a lesser number of files, inactive files are deleted.

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 522

Full incremental backup

When a user requests a full incremental backup, the IBM Tivoli Storage Manager determines its eligibility.

Procedure

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.

Procedure

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.

Procedure

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.

Results

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.

Procedure

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.

Procedure

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.

Results

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 537

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.

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
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

Task	Required Privilege Class
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 42 shows that an advantage of copying existing policy parts is that some associated parts are copied in a single operation.

Table 42. Cause and effect of copying existing policy parts

If you copy this...	Then you create this...
Policy Domain	A new policy domain with: • 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: • 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 70 on page 492 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.

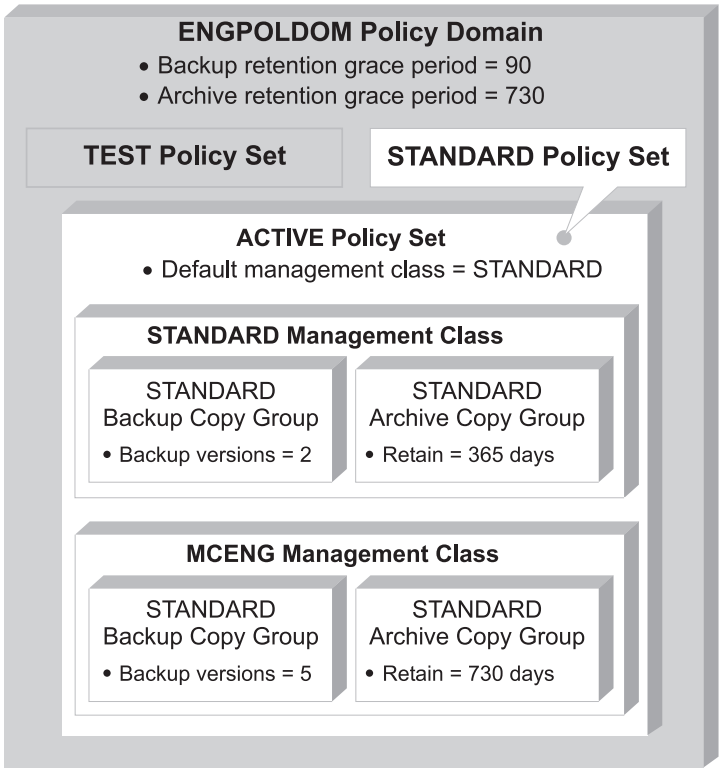


Figure 70. An example of policy objects defined for an engineering department

Related tasks:

"Defining and updating an archive copy group" on page 502

Related reference:

"Defining and updating a policy domain"

"Defining and updating a policy set" on page 494

"Defining and updating a management class" on page 495

"Defining and updating a backup copy group" on page 496

"Assigning a default management class" on page 504

"Activating a policy set" on page 505

"Running expiration processing to delete expired files" on page 506

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.

About this task

When you copy an existing domain, you also copy any associated policy sets, management classes, and copy groups.

Procedure

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 505

Example: defining a policy set

An administrator must develop new policies based on the existing STANDARD policy set.

Procedure

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:

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.

Procedure

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 497

"Retaining backup versions" on page 498

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 43 and Figure 71. 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 44 on page 500 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 43. 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)

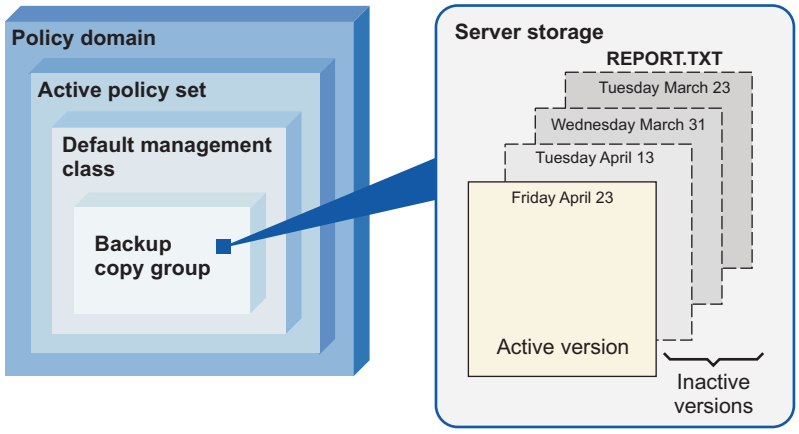


Figure 71. Active and inactive versions of `REPORT.TXT`

Table 44. 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	**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. 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.
NOLIMIT	2 versions	60 days	180 days	**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. 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.
NOLIMIT	NOLIMIT	60 days	180 days	**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. 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.
4 versions	2 versions	NOLIMIT	NOLIMIT	**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. 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.

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 546

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.

About this task

This new copy group must be able to complete 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 (`retonly=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`)

Procedure

Issue the following command to define the backup copy group:

```
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.

Procedure

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 509

Related tasks:

"Using virtual volumes to store data on another server" on page 722

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 481

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.

Procedure

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 494

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 result 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.

Condition	Reason for warning
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.
The current ACTIVE policy set names a management class that is not defined in the policy set being validated.	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. 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.
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 483

"Defining and updating a policy domain" on page 492

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 508

Related tasks:

"Changing policy" on page 472

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 policyset engpoldom standard
   ```

   Examine any messages that result and correct the problems.
2. To activate the STANDARD policy set, enter:

   ```
   activate policyset 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.

About this task

You can use the server command line to assign client nodes to a policy domain.

Procedure

- 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 548

"Deletion hold" on page 509

Running expiration processing automatically

You control automatic expiration processing by using the expiration interval option (EXPINTERVAL) in the server options file (`dsmserv.opt`). You can also control when restartable restore sessions expire with another server option, RESTOREINTERVAL.

You can set the options by editing the `dsmserv.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 that you start the server. After that, the server runs expiration processing at the interval you specified with the option, which is measured from the start time of the server.

Using commands to control expiration processing

You can manually run expiration by issuing the **EXPIRE INVENTORY** command or you can set a specific schedule for expiration processing by using the **DEFINE SCHEDULE** command.

After issuing **EXPIRE INVENTORY**, expired files are deleted from the database according to how you specify parameters on the command.

You can control how long the expiration process runs by using the **DURATION** parameter with the **EXPIRE INVENTORY** command. You can run several (up to 40) expiration processes in parallel by specifying **RESOURCE=***x*, where *x* equals the number of nodes that you want to process. Inventory expiration can also be distributed across more than one resource on a file space level to help distribute the workload for nodes with many file spaces.

You can use the **DEFINE SCHEDULE** command to set a specific schedule for this command. This automatically starts inventory expiration processing. 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.

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 998

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 are satisfied. It is enforced at the server level by 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 is 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 does 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 signaled plus the number of days specified in the **RETVER** parameter. On servers that have retention protection enabled, the following operations will not delete objects whose retention criterion is not 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.

The server does not send a retention value to an EMC Centera storage device if retention protection is not enabled. If this is the case, you can use a Centera storage device as a standard device from which archive and backup files can be deleted.

Related tasks:

Chapter 28, "Protecting and recovering the server infrastructure and client data," on page 857

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 508

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 `RETENTIONEXTENTION` 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.

About this task

When you configure the storage pools this way, you ensure that your data is 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 succeeds, but a warning message is issued.

Procedure

Complete 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 511

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 517

"Policy for logical volume backups" on page 517

"Configuring policy for NDMP operations" on page 518

"Configuring policy for LAN-free data movement" on page 519

"Policy for Tivoli Storage Manager servers as clients" on page 521

"Setting policy to enable point-in-time restore for clients" on page 522

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 can configure a policy to store client data directly in tape storage pools to reduce contention for disk resources.

About this task

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.

At the server command line, you can define a new policy domain that enables client nodes to back up or archive data directly to tape storage pools. For example, you can define a policy domain named DIR2TAPE with the following steps:

Procedure

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 policyset 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 that use the server to store data might require that you configure policies to make the most efficient use of server storage. See the documentation for each application client for the policy requirements.

Some of the application clients include a timestamp in each database backup. Because the default policy for the server keeps one backup version of each unique file, database backups that are managed by default policy are never deleted. The backups are not deleted because each backup is uniquely named with its timestamp. To ensure that the server deletes backups as required, configure policy as recommended by the documentation 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 45 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 45. 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 459

Related reference:

Chapter 9, "Using NDMP for operations with NAS file servers," on page 199

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 125

Related reference:

"Define a new management class in an existing policy domain" on page 521

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.

Procedure

Complete 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 125

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 722

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 691

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     Policy     Mgmt       Copy      Versions  Versions    Retain   Retain
Domain     Set Name   Class      Group        Data      Data     Extra     Only
Name                  Name       Name       Exists   Deleted  Versions  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        Policy        Mgmt          Copy           Retain
Domain        Set Name      Class         Group          Version
Name                        Name          Name
---------     ---------     ---------     ---------     --------
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       Policy       Mgmt         Default      Description
Domain       Set Name     Class        Mgmt
Name                      Name         Class ?
---------    ---------    ---------    ---------    ------------------------
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 policyset 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        Policy        Default       Description
Domain        Set Name      Mgmt
Name                        Class
                            Name
---------     ---------     ---------     ------------------------
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       Activated    Activated     Number of    Description
Domain       Policy       Default      Registered
Name         Set          Mgmt              Nodes
                          Class
---------    ---------    ---------    ----------    ------------------------
APPCLIEN-    BASE         APPCLIEN-             1    Policy domain for
 TS                        TMC                        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

Task	Required Privilege Class
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 508

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.

Before you begin

Move any client nodes to another policy domain, or delete the nodes.

Procedure

As an example, to delete the STANDARD policy domain, complete 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 483

Chapter 14. Managing data for client nodes

You might need help to generate backup sets and enable subfile backups for client nodes. Subfile backups are only applicable to Windows clients. 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 530
“Encrypting data on tape” on page 530
“Setting up shredding” on page 534
“Generating client backup sets on the server” on page 538
“Restoring backup sets from a backup-archive client” on page 542
“Moving backup sets to other servers” on page 542
“Managing client backup sets” on page 543
“Enabling clients to use subfile backup” on page 546
“Managing storage usage for archives” on page 549
Concepts:
“Performance considerations for data validation” on page 530
“Securing sensitive client data” on page 533
“Creating and using client backup sets” on page 537

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 879 to help you determine where to enable data validation.

© Copyright IBM Corp. 1993, 2013

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 878.

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 different methods of drive encryption for the following devices:

- IBM 3592 generation 2 and generation 3
- IBM linear tape open (LTO) generation 4 and generation 5

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 drives that are 3592 generation 2 and later" on page 182
- "Encrypting data using LTO generation 4 tape drives" on page 190
- "Enabling ECARTRIDGE drive encryption" on page 193 and "Disabling ECARTRIDGE drive encryption" on page 194

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

used. 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 use 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 use 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 46 on page 533 illustrates considerations for changing your hardware encryption method.

Table 46. 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

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

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.

Procedure

Complete 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 shreddom
   define policyset shreddom shredpol
   define mgmtclass shreddom shredpol shredclass
   define copygroup shreddom shredpol shredclass type=backup
    destination=shred1
   define copygroup shreddom shredpol shredclass type=archive
    destination=shred1
   activate policyset shreddom shredpol
   ```
5. Identify those client nodes whose data should be shredded after deletion, and assign them to the new domain.

   ```
   update node engineering12 domain=shreddom
   ```

Results

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 72 displays a detailed report for the storage pool.

```
Shredding  Objects  Occupied   Data left
 Active   Awaiting    Space     to shred
           Shred      (MB)       (MB)
--------- --------  --------  ----------
    NO        4        182        364
```

Figure 72. 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 47 describes these types of operations.

Table 47. 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 ACTIVEDATA	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.

Table 47. Commands affecting the shredding process (continued)

Command	Operation
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.
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. • 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, which is 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 that are 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.

Centera storage devices do not support backup sets.

The media might 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 with files that were backed up to Tivoli Storage Manager by using NDMP. However, you can create a backup set with files that were backed up using NetApp SnapShot Difference.

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 "Selection for active file versions" on page 239.

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 does not issue a warning if the backup set operation includes data from shred pools. For more information about shredding, see "Securing sensitive client data" on page 533.

For details about creating and using backup sets, see the following sections:

- "Generating client backup sets on the server" on page 538

- "Restoring backup sets from a backup-archive client" on page 542
- "Moving backup sets to other servers" on page 542
- "Managing client backup sets" on page 543

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.

About this task

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. You must have system or restricted policy over the domain to which the node is assigned to generate a 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.

Procedure

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.

Results

See the following sections:

- "Choosing media for generating the backup set" on page 539
- "Selecting a name for the backup set" on page 539
- "Setting a retention period for the backup set" on page 540
- "Example: generating a client backup set" on page 540
- "Generating backup sets to a specific point-in-time" on page 540
- "Generating backup sets with multiple data types" on page 541

- "Creating a single set of backup volumes with data from multiple nodes" on page 541

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.

About this task

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.

Procedure

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.
- You must restore, with the IBM Tivoli Storage Manager server, backup sets that are written to more than one volume and generated to a REMOVABLEFILE device. Issue the **RESTORE BACKUPSET** command and specify `-location=server` to indicate that the backup set is on the Tivoli Storage Manager server.

Results

For more information, see "Removable file device configuration" on page 119.

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.

Procedure

Complete 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 542.

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 the 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.

About this task

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.

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.

Procedure

Using the example described in "Example: generating a client backup set" on page 540, 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"
```

Results

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.

Important:

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.

About this task

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.

Procedure

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.

About this task

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:    /tsmpool/bksetvol01.ost /tsmpool/bksetvol02.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  Client's Name for File
                        Name
----------------------- ---------- --------------------------
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 Operations Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (??), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

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.

Procedure

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
```

Results

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” on page 547.

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 522

"Policy for logical volume backups" on page 517

Setting up clients to use subfile backup

The subfile backup pertains to the sections of the files that have changed.

Procedure

To enable subfile backup, complete 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`, `SUBFILECACHEPATH`, 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 459 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 506.

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.

Archiving 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 with automated tools that start the command-line client or API, you might encounter large numbers. If performance degrades over time during an archive operation, or you have a large amount of storage that is used by archives, consider advanced techniques. See "Archive operations overview" on page 549 and "Managing storage usage for archives" on page 549.

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"

"Preventing archive of directory-level access permissions" on page 551

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 15. 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.

About this task

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 554
"Defining client schedules" on page 555
"Associating client nodes with schedules" on page 556
"Starting the scheduler on the clients" on page 556
"Displaying information about schedules" on page 563
"Starting the scheduler on the clients" on page 556
"Displaying information about schedules" on page 563
"Creating schedules for running command files" on page 558
"Updating the client options file to automatically generate a new password" on page 559

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 415.
- 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 421 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.

© Copyright IBM Corp. 1993, 2013

Scheduling a client operation

To automate client operations, you can define new schedules.

About this task

When you define a schedule, you assign it to a specific policy domain. You can define more than one schedule for each policy domain.

Procedure

To set up a client schedule on the server, complete the following steps:

1. Define a schedule by using the **DEFINE SCHEDULE** command. See "Defining client schedules" on page 555.
2. Associate client nodes with the schedule by using the **DEFINE ASSOCIATION** command. See "Associating client nodes with schedules" on page 556.
3. Ensure that the clients start the client scheduler. See "Starting the scheduler on the clients" on page 556.
4. View schedule information and check that the schedule completed successfully by using the **QUERY SCHEDULE** and **QUERY EVENT** commands. See "Displaying schedule information" on page 557.

Example

The following example shows the commands that are used to schedule client backups for three registered client nodes that are assigned to the STANDARD policy domain.

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 with the following characteristics:

   - 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 on each client also affects which files are backed up or archived by the two schedules that are defined in the preceding steps. For example, if a file is excluded from backup with an EXCLUDE statement, the file is not backed up when the DAILY_INCR schedule runs.

3. Because the DAILY_INCR schedule is to run daily, you can verify whether it is working on the day after you define the schedule and associate it with clients. If the schedule runs successfully, the status is *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*.

What to do next

You can modify, copy, and delete any schedule that you create. For more information, see Chapter 16, "Managing schedules for client nodes," on page 561.

Defining client schedules

When you schedule client operations, you need to know what operation you want to run and the time, day, and frequency to run your operation.

About this task

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, which is the default.
- The priority for the operation is 5, which 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, which is the default.
- The schedule never expires, which is the default.
- The schedule style is classic, which is the default.

As a best practice, define schedules with durations longer than 10 minutes. This action gives 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.

About this task

You must have system privilege, unrestricted policy, or restricted policy (for the policy domain to which the schedule belongs) to associate client nodes with schedules. Issue the **DEFINE ASSOCIATION** command to associate client nodes with a schedule.

Procedure

Complete the following step to associate the ENGNODE client node with the WEEKLY_BACKUP schedule, both of which belong to the ENGPOLDOM policy domain:

```
define association engpoldom weekly_backup engnode
```

Results

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

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 427 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. You must have administrative privilege to display information about scheduled operations.

About this task

- 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.

About this task

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 565.

- 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.

About this task

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.

About this task

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 16. Managing schedules for client nodes

You can manage and coordinate Tivoli Storage Manager schedules for registered client nodes.

About this task

Administrators can perform the following tasks:

Tasks:
"Managing node associations with schedules" on page 563
"Specifying one-time actions for client nodes" on page 575
"Managing event records" on page 565
"Managing the throughput of scheduled operations" on page 568
"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 415. For information about the scheduler and creating schedules, see Chapter 15, "Scheduling operations for client nodes," on page 553

Managing IBM Tivoli Storage Manager schedules

You can perform the following activities to manage schedules.

About this task

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 and associate schedules by using the Tivoli Storage Manager command-line interface.

About this task

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 555.

© Copyright IBM Corp. 1993, 2013

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.

About this task

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 556.

Modifying schedules

You can modify existing schedules by issuing the **UPDATE SCHEDULE** command.

About this task

You can use the server command line to modify existing schedules in a policy domain.

Procedure

To modify the ENGWEEKLY client schedule in the ENGPOLDOM policy domain, issue the following command:

```
update schedule engpoldom engweekly period=5 perunits=days
```

Results

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.

About this task

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 565.

See "Associating client nodes with schedules" on page 556 for more information.

Displaying information about schedules

You can display information about schedules.

About this task

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.

About this task

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

Task	Required Privilege Class
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.

About this task

To associate client nodes with a schedule, you can use the following method:

Issue the **DEFINE ASSOCIATION** command from the command-line interface.

For more information, see "Associating client nodes with schedules" on page 556.

Moving nodes from one schedule to another

You can move a node from one schedule to another schedule.

Procedure

1. Associate the node to the new schedule. For information, see "Adding new nodes to existing schedules."
2. Delete the association of that node from the original schedule.

Related tasks:

"Associating client nodes with schedules" on page 556

"Removing nodes from schedules" on page 565

Displaying nodes associated with schedules

You can display information about the nodes that are associated with a specific schedule.

About this task

For example, you should query an association before deleting a client schedule.

Figure 73 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 73. 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.

About this task

To delete the association of the ENGNOD client with the ENGWEEKLY schedule, in the policy domain named ENGPOLDOM, enter:

```
delete association engpoldom engweekly engnod
```

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.

About this task

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.

About this task

- 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 file” on page 567.

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*.

About this task

Figure 74 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 74. Events for a node

Displaying events that ended unsuccessfully

You can display information about scheduled events that ended unsuccessfully by using exception reporting.

About this task

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 75 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 75. 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.

About this task

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 file

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.

About this task

The default name for the schedule log 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 Backup-Archive Clients Installation and 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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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 569.

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.

Procedure

To limit the start of backup-archive client sessions to the server only, complete 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 49 on page 570 and Table 48 for the advantages and disadvantages of client-polling and server-prompted modes.

Table 48. Client-Polling mode

How the mode works	Advantages and disadvantages
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 *Backup-Archive Clients Installation and User's Guide*. 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.	• 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 *randomization*, 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 49. Server-Prompted mode

How the mode works	Advantages and disadvantages
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.	• 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.

About this task

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.

About this task

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.

About this task

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 496.

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.

About this task

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.

About this task

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 MAXSCHEDSESSIONS 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 MAXSCHEDSESSIONS command	May reduce the server's ability to process unscheduled operations

For information about the MAXSESSIONS option and the SET MAXSCHEDSESSIONS 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.

About this task

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, specify the percentage of the startup window that the server can use to randomize start times for different client nodes that are 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 that are allowed to make sure that the server can handle the workload. The maximum percentage of randomization that is allowed is 50%. This limit ensures that half of the startup window is available for trying again, the scheduled commands that 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.

About this task

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.

About this task

- 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" on page 574

"Setting the amount of time between retry attempts" on page 574

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.

About this task

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 queryschedperiod 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.

About this task

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.

About this task

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.”

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.

About this task

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.

About this task

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.

© Copyright IBM Corp. 1993, 2013

Chapter 17. Managing server operations

Administrators can perform such server operations as licensing purchased features, starting and halting the server, and monitoring server information.

About this task

When possible, avoid running simultaneous data movement or delete operations, or running simultaneous storage pool data access operations. One or more of the operations might terminate prematurely if the data involved in a data access operation is also involved in a data movement operation.

Examples of a data access operation include restoring, retrieving, or recalling data from a client. Data access operations also include the **GENERATE BACKUPSET** and **EXPORT** commands, as well as storage pool backups. Examples of data movement and delete operations include migration, reclamation, **MOVE DATA**, **EXPIRE INVENTORY**, **DELETE FILESPACE**, and **DELETE VOLUME** when DISCARRDATA=YES.

Also, if a delete operation is processing data involved in either a data access or a data movement operation, either of the operations might terminate prematurely.

See the following topics:

Tasks:
"Licensing IBM Tivoli Storage Manager"
"Starting the Tivoli Storage Manager server" on page 588
"Moving the Tivoli Storage Manager server to another system" on page 599
"Date and time on the server" on page 600
"Managing server processes" on page 600
"Preempting operations" on page 602
"Setting the server name" on page 604
"Add or update server options" on page 605
"Getting help on commands and error messages" on page 607

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.

© Copyright IBM Corp. 1993, 2013

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).

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 media, and are placed on the server during installation. 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 registers 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. To unregister licenses, erase the NODELOCK file found in the server instance directory and reregister the licenses.

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 and Expiration and Deletion Suspension (Deletion Hold).

***.lic** Registers all IBM Tivoli Storage Manager licenses for server components.

Notes:

- The NODELOCK file name is case-sensitive and must be entered in all uppercase letters.
- 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 might need to register your licenses again for any of the following reasons:
  - The server is corrupted.
  - The server is moved to a different computer.
  - 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.

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.

Role of processor value units in assessing licensing requirements

You can obtain information about the number of client and server devices that are managed by Tivoli Storage Manager, and the utilization of processor value units (PVUs) by server devices. Use this data to assess licensing requirements for the Tivoli Storage Manager system.

Important: The PVU calculations that are provided by Tivoli Storage Manager are considered estimates and are not legally binding. The PVU information reported by Tivoli Storage Manager is not considered an acceptable substitute for the IBM License Metric Tool.

Collecting PVU information

Review the following procedure to understand how Tivoli Storage Manager estimates PVUs. The procedure also describes the steps that you can take to make the estimates more accurate.

1. When you install the Tivoli Storage Manager V6.3 server, or upgrade the server from a previous version, Common Inventory Technology is installed automatically. Common Inventory Technology is a software component that collects the metrics that are used to calculate PVUs.
2. When Tivoli Storage Manager clients connect to the Tivoli Storage Manager server, the clients send Common Inventory Technology information to the Tivoli Storage Manager server. This information includes the number of client and server devices, the number of physical processors, the processor vendor and type, and related information. The server will estimate PVUs only for nodes that are classified as server nodes. If you change the default classification for a device, the classification that you specify is used for PVU estimation. For more information about devices and nodes, see Device classification.

   Tip: Common Inventory Technology is not installed automatically on client nodes that are running on Linux operating systems. If you plan to obtain PVU information from a client node that is running on a Linux operating system, install Common Inventory Technology on the client node. For details, see the topic about installing UNIX and Linux backup-archive clients in the *IBM Tivoli Storage Manager Backup-Archive Clients Installation and User's Guide*.
3. The Tivoli Storage Manager server stores the information in the Tivoli Storage Manager database.
4. The Tivoli Storage Manager server generates PVU estimates. The estimates are based on the values that are stored in the Tivoli Storage Manager database and on PVU values that are defined for each processor type in the `IBM_ProcessorValueUnitTable.xml` file. The `IBM_ProcessorValueUnitTable.xml` file is installed automatically during the Tivoli Storage Manager server installation. You also can download the latest version of the file from an IBM FTP site. For more information, see Estimating processor value units.
5. You can browse PVU information by issuing commands that provide PVU estimates. For more information about obtaining and adjusting PVU estimates, see Estimating processor value units.

The following graphic illustrates the PVU estimation process.

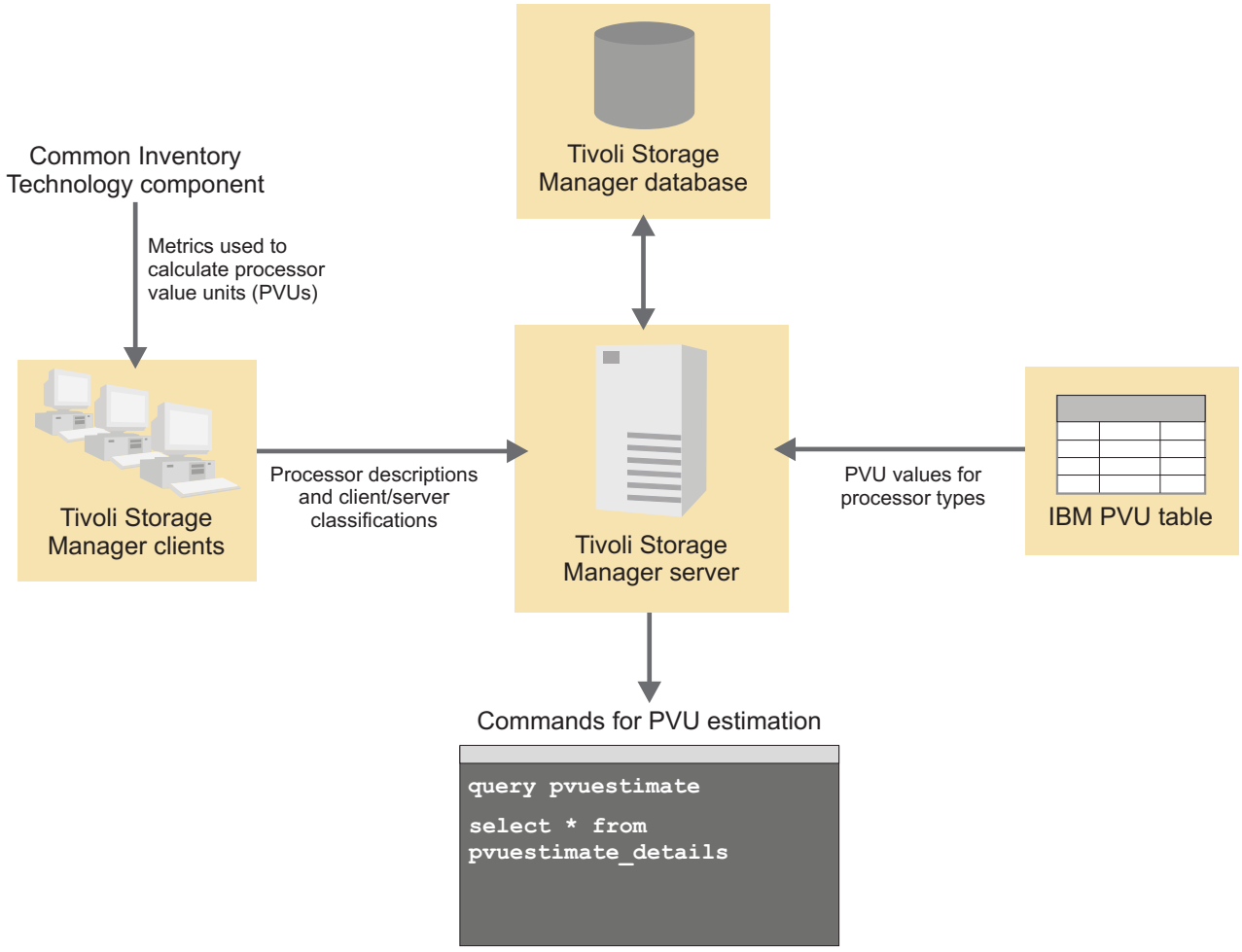


Figure 76. PVU estimation overview

Device classification

For purposes of PVU calculation, you can classify devices, such as workstations and servers, as client nodes, server nodes, or other. By default, devices are classified as client or server:

Client Backup-archive clients that run on Microsoft Windows 7, Microsoft Windows XP Professional, and Apple systems are classified as client devices.

Server Backup-archive clients that run on all platforms except for Microsoft Windows 7, Microsoft Windows XP Professional, and Apple systems are classified as server devices. All other node types are also classified as server devices. The server on which Tivoli Storage Manager is running is classified as a server device.

You can change the node classification to reflect how the device is used in the system. For example, if a node is classified as a server, but functions as a client, you can reclassify it as a client. If a node is not used in the system, you can reclassify it as other.

When you assign a classification, consider the services that are associated with the device. For example, a Microsoft Windows XP Professional notebook might be a

client device if it is distributed to a university staff member as a single-use workstation. The same computer might be a server device if it is distributed to a department administrator, who provides additional services such as file sharing on the computer to multiple workstations.

In a Tivoli Storage Manager system, you can assign multiple client node names to the same physical workstation. For example, a clustering solution can have several node names that are defined in the Tivoli Storage Manager server environment to provide protection if a failover occurs. Redundant node names, or node names that manage data for physical workstations that no longer exist, should not be counted for licensing purposes. In this case, you might classify the node as other by using the **UPDATE NODE** command.

Limitations

The PVU calculations are estimates because the software cannot determine all of the factors that are required for a final number. The following factors affect the accuracy of the calculations:

- PVU estimates are provided only for Tivoli Storage Manager V6.3 server devices that have established a connection with the Tivoli Storage Manager server since the installation of or upgrade to Tivoli Storage Manager V6.3.
- The default classification of nodes is based on assumptions, as described in "Device classification" on page 583.
- The PVU estimate might not reflect the actual number of processors or processor cores in use.
- The PVU estimate might not reflect cluster configurations.
- The PVU estimate might not reflect virtualization, including VMware and AIX LPAR and WPAR.
- Common Inventory Technology might not be able to identify some processors, and some processors might not have corresponding entries in the PVU table.

Formula for PVU estimation

You must have the following information to calculate PVUs:

- The node classification. PVUs are calculated only for server devices.
- The number of processors on the node.
- The processor vendor, brand, type (number of cores), and model. This information is used to look up the PVU value in the IBM PVU table.

Calculate the PVUs for a node by using the following formula:

```
PVUs = number of processors on the node * processor type (core count) * pvu value
```

To calculate the total PVUs, sum the PVUs for all nodes.

Related information

Table 50. Information about PVUs and licensing

Information type	Location
IBM PVU table	ftp://public.dhe.ibm.com/software/tivoli_support/misc/CandO/PVUTable/
PVU calculator	https://www.ibm.com/software/howtobuy/passportadvantage/valueunitcalculator/vucalc.wss

Table 50. Information about PVUs and licensing (continued)

Information type	Location
PVU frequently asked questions	http://www.ibm.com/software/lotus/passportadvantage/pvufaqgen.html
PVU licensing for distributed software	http://www.ibm.com/software/lotus/passportadvantage/pvu_licensing_for_customers.html
PVU terminology	http://www.ibm.com/software/lotus/passportadvantage/pvu_terminology_for_customers.html

Estimating processor value units

To estimate the number of processor value units (PVUs) for your system, use the **QUERY PVUESTIMATE**, **SELECT * FROM PVUESTIMATE_DETAILS**, or the **QUERY NODE** command.

Before you begin

Before you begin, review the information about how PVUs are estimated and what the limitations are. For more information, see "Role of processor value units in assessing licensing requirements" on page 582. Tivoli Storage Manager offers several options for viewing PVU information. Select the option that best meets your needs. To export the PVU estimates to a spreadsheet, use the **SELECT * FROM PVUESTIMATE_DETAILS** command.

Important: The PVU calculations that are provided by Tivoli Storage Manager are considered estimates and are not legally binding.

Procedure

To estimate PVUs, complete the following steps:

1. To ensure that you have the latest IBM PVU table, go to the FTP site at ftp://public.dhe.ibm.com/software/tivoli_support/misc/CandO/PVUTable/.
2. Download the file for the latest table and rename it `IBM_ProcessorValueUnitTable.xml`.
3. Copy the renamed file to the Tivoli Storage Manager server instance directory. Alternatively, you can store the file in the server installation directory, where it replaces the installed file. Then, restart the server.

   Tip: The `IBM_ProcessorValueUnitTable.xml` file name is case-sensitive. Ensure that the file name on your system matches this file name.
   A version of the IBM PVU table is also provided with the Tivoli Storage Manager installation.
4. Obtain the type of PVU information that you require:

Option	Description
Product-based information	To obtain a PVU report that is categorized by product, issue the **QUERY PVUESTIMATE** command. For more information about the **QUERY PVUESTIMATE** command, see the *Tivoli Storage Manager Administrator's Reference*. **Remember:** The reports that are produced by the **QUERY PVUESTIMATE** and **SQL SELECT * FROM PVUESTIMATE_DETAIL** commands provide information about all client and server devices that are managed by the Tivoli Storage Manager server. However, PVU estimates are provided only for the V6.3 server devices that established a connection with the Tivoli Storage Manager server since the installation of or upgrade to Tivoli Storage Manager V6.3.
Node-based information	To obtain a PVU report that provides detailed information at the node level, issue the **SELECT * FROM PVUESTIMATE_DETAILS** command. For more information about the **SELECT** command, see the *Tivoli Storage Manager Administrator's Reference*. After you obtain data from the **SELECT * FROM PVUESTIMATE_DETAILS** command output, you can move the data to a spreadsheet. Then, modify the data to gain a more accurate estimate. To export data to a spreadsheet, use the `-comma` and `> output` redirection options of the **dsmadmc** administrative command-line client with the **SELECT** administrative command. For example, complete the following steps: 1. `dsmadmc -comma select * from pvuestimate_details > pvuestimatefile.csv` 2. Import the resulting CSV file into a spreadsheet program.
Complete information	To view complete information, including PVU-related fields that are provided by Common Inventory Technology, such as processor vendor, brand, type, model, and processor count for selected nodes, specify the `DETAILED` value for the **FORMAT** parameter. For more information about the **QUERY NODE** command, see the *Tivoli Storage Manager Administrator's Reference*.

5. To obtain a more accurate PVU estimate, you might want to change the classifications of nodes. To change node classifications, issue the **UPDATE NODE** command. For more information about the **UPDATE NODE** command, see the *Tivoli Storage Manager Administrator's Reference*.
6. To calculate the PVUs for a node, use the following formula: `PVUs = number of processors on the node * processor type (core count) * pvu value`. To calculate the total PVUs, sum the PVUs for all nodes. For more information about the PVU estimation formula, see Formula for PVU estimation.

7. After you generate a PVU report, more analysis might include removing redundancies, deleting obsolete information from the report, and accounting for known systems that are not logged in to and connected to the server.

   Tip: If you cannot obtain PVU information from a client node that is running on a Linux operating system, ensure that Common Inventory Technology is installed on that client node. After you install Common Inventory Technology, obtain a new PVU estimate.

Collecting processor value unit information in a VMware host environment

If you are installing Tivoli Storage Manager on a system that uses VMware virtualization software, ensure that PVU information is collected in the virtual environment. To collect PVU information in a virtual environment, you must have VMware ESX 4 software installed on your system, and you must set up and run hardware scans.

Procedure

Complete the following steps:

1. Obtain the Tivoli Storage Manager backup-archive client installation package for the Linux x86 operating system.

   For information about obtaining installation packages, see the Passport Advantage website at http://www.ibm.com/software/lotus/passportadvantage/pacustomers.html.
2. Install the Tivoli Storage Manager backup-archive client on a computer in your system.
3. From the `../CIT/VMware/esx-4` directory, download the `dispatcher`, `retrieve.sh`, `wenvmw.sh`, and `cpuid` files.
4. Copy the downloaded files to the ESX 4 host operating system.

   Tip: You can copy the files to any location on the host operating system, but ensure that all files are copied to the same directory.
5. Ensure that guest virtual machines are running. This step is necessary to ensure that the guest virtual machines are detected during the hardware scan.
6. To collect PVU information, issue the following command:

   ```
   retrieve -v
   ```

What to do next

If you restart the host machine or change the configuration, run the **retrieve** command again to ensure that current information is retrieved.

Tip: When the IBM Tivoli Storage Manager for Virtual Environments license file is installed on a VMware vStorage backup server, the platform string that is stored on the Tivoli Storage Manager server is set to TDP VMware for any node name that is used on the server. The reason is that the server is licensed for Tivoli Storage Manager for Virtual Environments. The TDP VMware platform string can be used for PVU calculations. If a node is used to back up the server with standard backup-archive client functions, such as file-level and image backup, interpret the TDP VMware platform string as a backup-archive client for PVU calculations.

Starting the Tivoli Storage Manager server

You can select different methods for starting the Tivoli Storage Manager server, depending on the operating system on which the server is installed.

To review the methods that are available if the server is installed on AIX, HP-UX, Linux, or Solaris, see "Starting the server on AIX, HP-UX, Linux, and Solaris."

Starting the server on AIX, HP-UX, Linux, and Solaris

You can start the Tivoli Storage Manager server by using the instance user ID or the root user ID. In addition to selecting a user for starting the server, you can also select one of several modes for starting the server.

About this task

The preferred way to start the server is by using the instance user ID. By using the instance user ID, you simplify the setup process and avoid potential issues. However, in some cases, it might be necessary to start the server with the root user ID. For example, you might want to use the root user ID to ensure that the server can access specific devices.

Procedure

To start the server, take one of the following actions:

- To start the server by using the instance user ID, complete the following steps:
  1. Ensure that you set access permissions and user limits correctly, as described in "Verifying access rights and user limits" on page 589.
  2. Start the server as described in "Starting the server from the instance user ID" on page 590.
- To start the server by using the root user ID, complete the following steps:
  1. Ensure that you set access permissions and user limits correctly, as described in "Verifying access rights and user limits" on page 589.
  2. Authorize the root user ID to start the server, as described in "Authorizing root user IDs to start the server" on page 591.
  3. Start the server as described in "Starting the server from the root user ID" on page 592.
- To start the server by using the instance user ID or the root user ID and an alternative method, follow the steps for the selected method:
  - Start the server automatically, as described in "Automatically starting AIX, HP-UX, and Solaris servers" on page 593.
  - Start the server in background mode, as described in "Running the server in background mode" on page 594.
  - Start the server in stand-alone mode, as described in "Stand-alone mode for server startup" on page 594.
  - Start the server in another mode, as described in "Starting the server with other options and modes" on page 596.

Related reference:

"rmstdev (Detect and delete device special files)" on page 86

Verifying access rights and user limits

Before you start the Tivoli Storage Manager server on an AIX, HP-UX, Linux, or Solaris operating system, verify access rights and user limits.

About this task

If you do not verify user limits, also known as *ulimits*, you might experience server instability or a failure of the server to respond. You must also verify the system-wide limit for the maximum number of open files. The system-wide limit must be greater than or equal to the user limit.

Procedure

1. Verify that the server instance user ID has permissions to start the server.
2. For the server instance that you plan to start, ensure that you have authority to read and write files in the server instance directory. Verify that the `dsmserv.opt` file exists in the server instance directory, and that the file includes parameters for the server instance.
3. If the server is attached to a tape drive, medium changer, or removable media device, and you plan to start the server by using the instance user ID, grant read/write access to the instance user ID for these devices. To set permissions, take one of the following actions:
   - If the system is dedicated to Tivoli Storage Manager and only the Tivoli Storage Manager administrator has access, make the device special file world-writable:

     ```
     chmod +w /dev/rmtX
     ```
   - If the system has multiple users, you can restrict access by making the Tivoli Storage Manager instance user ID the owner of the special device files:

     ```
     chmod u+w /dev/rmtX
     ```
   - If multiple user instances are running on the same system, change the group name, for example TAPEUSERS, and add each Tivoli Storage Manager instance user ID to that group. Then, change the ownership of the device special files to belong to the group TAPEUSERS, and make them group writable:

     ```
     chmod g+w /dev/rmtX
     ```
4. To prevent server failures during interaction with DB2, tune the kernel parameters.

   For instructions, see the section about tuning kernel parameters in the *Installation Guide*.
5. Verify the following user limits based on the guidelines in the table.

Table 51. User limit (ulimit) values

User limit type	Standard value	Command to query value
Maximum size of core files created	Unlimited	`ulimit -Hc`
Maximum size of a data segment for a process	Unlimited	`ulimit -Hd`
Maximum file size	Unlimited	`ulimit -Hf`
Maximum number of open files	65536	`ulimit -Hn`
Maximum amount of processor time in seconds	Unlimited	`ulimit -Ht`

To modify user limits, follow the instructions in the documentation for your operating system.

Tip: If you plan to start the server automatically by using a script, you can set the user limits in the script.

6. Verify the system-wide value for the maximum number of open files. The system-wide value must be equal to or greater than the user limit for the maximum number of open files.
7. Ensure that the user limit of maximum user processes (`nproc`) is set to the minimum suggested value of 16384.
   a. To verify the current user limit, issue the following command by using the instance user ID:

      ```
      ulimit -Hu
      ```

      For example:

      ```
      [user@Machine ~]$ ulimit -Hu
      16384
      ```

   b. If the limit of maximum user processes is not set to 16384, set the value to 16384:

      Add a line to `/etc/security/limits.conf`.

Starting the server from the instance user ID

To start the server from the instance user ID, log in with the instance user ID and issue the appropriate command from the server instance directory.

Before you begin

Ensure that access rights and user limits are set correctly. For instructions, see "Verifying access rights and user limits" on page 589.

Procedure

1. Log in to the system where Tivoli Storage Manager is installed by using the instance user ID for the Tivoli Storage Manager server.
2. If you do not have a user profile that runs the `db2profile` script, issue the following command:

   ```
   . /home/tsminst1/sqllib/db2profile
   ```

   Tip: For instructions about updating the user ID login script to run the `db2profile` script automatically, see the DB2 documentation.
3. Start the server by issuing the following command from the server instance directory:

   ```
   /usr/bin/dsmserv
   ```

   Tip: The command runs in the foreground so that you can set an administrator ID and connect to the server instance.

   For example, if the name of the Tivoli Storage Manager server instance is `tsminst1` and the server instance directory is `/tsminst1`, you can start the instance by issuing the following commands:

   ```
   cd /tsminst1
    . ~/sqllib/db2profile
   /usr/bin/dsmserv
   ```

Alternatively, if you set up the Tivoli Storage Manager server as a service by using the Solaris Service Management Facility (SMF), you can start the service by issuing the **svcadm enable** command. For instructions, see Technote 7021102 (http://www.ibm.com/support/docview.wss?uid=swg27021102).

Authorizing root user IDs to start the server

A user that is not the instance owner must be explicitly granted authority to start a Tivoli Storage Manager server.

About this task

The preferred way to start the server is by using the instance user ID. By using the instance user ID, you simplify the setup process and avoid potential issues. However, in some cases, it might be necessary to start the server with the root user ID. For example, you might want to use the root user ID to ensure that the server can access specific devices. To start the server, the root user ID must meet the following requirements:

- Have authority to issue the start command for the server and database manager
- Belong to the group that is specified by the database manager SYSADM_GROUP setting
- Have database administration authority (DBADM) granted by the instance user ID
- Have authority to access the server database
- Have authority to use all files, directories, and devices that are required by the server

Before you start the server with the root user ID, explicitly grant server database authority to the root user ID and verify all other authorities for the root user ID.

Procedure

Log in, connect to the database, and issue the **DB2 GRANT** command.
In the following example, the user ID is tsminst1:

1. Log in by using the root user ID or the instance user ID.
   ```
   # su - tsminst1
   ```
2. Start DB2.
   ```
   $ db2start
   ```
3. Connect to the TSMDB1 database.
   ```
   $ db2 connect to tsmdb1
   ```
4. Grant the root user ID authority to the database.
   ```
   $ db2 grant dbadm on database to user root
   ```
5. Optional: Verify that the root user ID has the required database authority by issuing the following command:
   ```
   db2 get dbm cfg | grep SYSADM_GROUP
   ```
   Review the SYSADM_GROUP setting and verify that the root user ID belongs to the group.
6. Ensure that the root user ID has access to all files, directories, and devices that are required by the server.

Starting the server from the root user ID

The standard way to start the server is from the instance user ID, but the root user ID can be used in some cases. For example, it might be necessary to start the server from the root user ID to access specific devices.

Before you begin

Ensure that the root user ID meets the requirements for starting the server. Follow the instructions in Authorizing root users to start the server. Ensure that access rights and user limits are set correctly. Follow the instructions in "Verifying access rights and user limits" on page 589.

Procedure

1. Log in by using the root user ID.
2. Change the `.profile` file so that the root user ID can run the **db2profile** script for the instance user ID. Issue the following command:

   ```
   . db2_instance_home_directory/sqllib/db2profile
   ```

   where *db2_instance_home_directory* specifies the home directory of the DB2 instance. For example, if the instance name is tsminst1, issue the following command:

   ```
   . /home/tsminst1/sqllib/db2profile
   ```
3. Change to the instance directory. For example, if the instance directory is /tsminst1, issue the following command:

   ```
   # cd /tsminst1
   ```
4. Start the server instance:
   - To start the tsminst1 server by using the root user ID and run it with the instance user ID, use the `-u` option:

     ```
     # nohup /user/bin/dsmserv -u tsminst1 -q &
     ```

     With this command, the server program runs in the background.
   - To start the tsminst1 server by using the root user ID and run it with the root user ID, issue the following command. If the root user ID is a member of the primary group of instance user IDs, the root user ID has authority to start the database manager.

     ```
     # /user/bin/dsmserv
     ```

     The database and log files are written by the instance user ID, not the root user ID. Ensure that the permissions on the database and log directories are set to allow read and write access by the instance user ID.
5. Optional: Start the server automatically by using the root user ID. Use the script that is provided for starting the server automatically.

   Follow the instructions in "Automatically starting AIX, HP-UX, and Solaris servers" on page 593.

Automatically starting AIX, HP-UX, and Solaris servers

You can configure the server to start automatically at system startup. Use the `rc.dsmserv` script, which is provided for this purpose.

Before you begin

Ensure that access rights and user limits are set correctly. For instructions, see "Verifying access rights and user limits" on page 589.

About this task

The `rc.dsmserv` script is in the server installation directory, for example, in the `/opt/tivoli/tsm/server/bin` directory.

Procedure

If you did not use a wizard to configure the 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. For more information about multiuser mode and run levels, see the documentation for your operating system.
- 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 correct syntax for the entry by consulting documentation for your operating system.

Example

Automatically starting a server instance

In this example, the instance user ID is tsminst1; the server instance directory is `/home/tsminst1/tsminst1`; the run level is 3; and the process ID is tsm1. The following entry is added to the `/etc/inittab` file, on one line:

```
tsm1:3:once:/opt/tivoli/tsm/server/bin/rc.dsmserv -u tsminst1
   -i /home/tsminst1/tsminst1 -q >/dev/console 2>&1
```

Automatically starting several server instances

This example uses the following instance user IDs:

- tsminst1
- tsminst2

This example uses the following instance directories:

- `/home/tsminst1/tsminst1`
- `/home/tsminst2/tsminst2`

This example uses the following process IDs:

- tsm1
- tsm2

The run level is 3. In this example, the following entries are added to the `/etc/inittab` file. Each entry is on one line.

```
tsm1:3:once:/opt/tivoli/tsm/server/bin/rc.dsmserv -u tsminst1
   -i /home/tsminst1/tsminst1 -q >/dev/console 2>&1
tsm2:3:once:/opt/tivoli/tsm/server/bin/rc.dsmserv -u tsminst2
   -i /home/tsminst2/tsminst2 -q >/dev/console 2>&1
```

Running the server in background mode

When you run the server in the background, you control the server through the administrative client.

Before you begin

Verify that the following conditions exist:

- An administrative node is registered and granted system authority, as described in "Registering administrator IDs" on page 837.
- The administrative client options file is updated with the correct **SERVERNAME** and **TCPPORT** options.
- The administrative client can access the Tivoli Storage Manager server.

If these conditions are not met, and you run the server in background mode, you cannot control the server. You can stop the server only by canceling the process based on the process number that is displayed at server startup. You might not be able to stop the server cleanly without this process number.

Ensure that access rights and user limits are set correctly. For instructions, see "Verifying access rights and user limits" on page 589.

Ensure that you are running the Tivoli Storage Manager instance with the instance user ID.

Procedure

Change to the server instance directory and issue the following command:

```
/user/bin/dsmserv -q &
```

Stand-alone mode for server startup

Some Tivoli Storage Manager server operations require that you start the server in the single-user, stand-alone mode. It might be useful to start the server in this way for maintenance procedures or for a major reconfiguration of the server.

Here are some examples of operations that require that you start the server in stand-alone mode:

- Verifying the Tivoli Storage Manager server operations after a server upgrade.
- Verifying the Tivoli Storage Manager server operations after one of the following operations:
  - Restoring the server database by using the **DSMSERV RESTORE DB** utility.
  - Dumping, reinitializing, and reloading the server database if a serious error occurs (recovery log corruption, for example), and if the **DSMSERV RESTORE DB** utility cannot be used.
- Running Tivoli Storage Manager recovery utilities, as requested by IBM Software Support.

Before you start these tasks, 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

Running the server in stand-alone mode:

To run a Tivoli Storage Manager server in stand-alone mode, edit the server options file. Then, start the server and prevent new client sessions, administrative sessions, and server-to-server sessions.

Before you begin

Ensure that access rights and user limits are set correctly. For instructions, see "Verifying access rights and user limits" on page 589.

Procedure

1. Edit the server options file, `dsmserv.opt`, 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 server on AIX, HP-UX, Linux, and Solaris" on page 588.
3. Prevent new client sessions, administrative sessions, and server-to-server sessions by issuing the following command:

   ```
   disable sessions all
   ```

   Tip: You can continue to access the server. Any current client activities complete unless a user logs off or you cancel a client session.

What to do next

You can complete maintenance, reconfiguration, or recovery operations, and then halt the server. To restart the server, complete the following steps:

1. To return the server options to their original settings, edit the `dsmserv.opt` file.
2. Start the server, as described in "Starting the server on AIX, HP-UX, Linux, and Solaris" on page 588.
3. Enable client sessions, administrative sessions, and server-to-server sessions by issuing the following command:

   ```
   enable sessions all
   ```

Starting the server with other options and modes

When you start the server with the **dsmserv** command, you can specify various options and modes.

Before you begin

Ensure that access rights and user limits are set correctly, as described in Verifying access rights and setting user limits.

Procedure

Issue the **dsmserv** command with the appropriate option and mode:

```
dsmserv option mode
```

where *option* specifies an option for starting the server and *mode* specifies a startup mode. Not all options are compatible with all modes.

For example, specify one of the following options:

-i *instance_directory*
: Specifies the instance directory, which becomes the working directory of the server.

-noexpire
: Specifies that expiration processing is suppressed when the server is started.

-o *file_name*
: Specifies an options file to use if several servers are running.

-q
: Starts the server as a daemon program. The server runs as a background process, and does not read commands from the server console. Messages to the console are suppressed.

  If you plan to start the server as a daemon program, you must have an administrative client that is registered and authorized with system authority. The administrative client must be started. Otherwise, the server runs in quiet mode and you will not be able to access the server.

-S
: Specifies that ID verification for the database will be skipped.

For example, specify one of the following modes:

display
: Displays information about the database and recovery log.

extend
: Extends the size of the recovery log.

format
: Formats the database and recovery log for normal use.

loadformat
: Formats the database and recovery log to enable the loading of data into the database.

restore
: Restores the database from a backup copy.

runfile *file_name*
: Starts the server, runs the commands that are defined in the specified file, and stops the server.

Running multiple server instances on a single system

To run multiple server instances on a single system, set up separate database and recovery log directories for each server instance. Also, set up an instance directory for each server instance to contain the server options file and other files that are required to run the instance.

About this task

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 the files that are used by another server instance on the same system, and separately from the server program files. Database and recovery log files are stored in separate directories, not in the server instance directory.

As part of server configuration, you create a directory to store the files for the server instance. The following files are stored in the server instance directory:

- The server options file, `dsmserv.opt`
- The server key database file, `cert.kdb`, and the `.arm` files, which are used by clients and other servers to import the Secure Sockets Layer certificates of the server
- 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
- The volumes for storage pools of the FILE device type, if the directory for the device class is not fully specified, or not fully qualified
- The user exits
- The trace output, if it is not fully qualified
- The log file for first-failure data capture, `dsmffdc.log`

Procedure

1. Set up database and recovery log directories for each server instance.
2. Set up an instance directory for each server instance. For example, to run two server instances, `tsminst1` and `tsminst2`, create instance directories such as `/tsminst1` and `/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. Ensure that all required files are included in the instance directory.
3. To manage the system memory that is used by each server on a system, use the **DBMEMPERCENT** server option. 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.
4. Optional: Set up the server instances to start automatically.

   For instructions, see "Automatically starting AIX, HP-UX, and Solaris servers" on page 593.

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 are finished or canceled. To shut down the server without severely impacting administrative and client node activity with the server, you must complete the following steps:

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 whether any sessions are running, use the QUERY SESSION command. If a session is running, a table is displayed showing the session number on the leftmost of the screen.
4. Find out whether 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 you halt 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 waiting for a tape volume to be mounted (for example, a process that is initiated by EXPORT, IMPORT, or MOVE DATA commands), the mount request is automatically canceled. If a volume associated with the process is being mounted by an *automated* library, the cancel might 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 by using the ALIASHALT server option. By using the server option, you can define a term other than HALT that completes the same function. The HALT command functions, however the server option provides an extra 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.

Moving the Tivoli Storage Manager server to another system

You can move your Tivoli Storage Manager server from one computer to another.

Before you begin

The following prerequisites 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 the code level on the server that was backed up.

Procedure

To move the database to another system, complete the following steps:

1. Install Tivoli Storage Manager on the target server. Ensure that you install the instance as if you were creating a new server. See the *Installation Guide* for details.
2. Back up the database to sequential media. For example, issue the following command:

   ```
   backup db devclass=lto4 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. Ensure that you issue the following commands as the instance user. 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 must be restored on separate lines. 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 669

"Starting the server from the instance user ID" on page 590

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 an automated library
- 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 Process Description      Status
  Number
-------- ----------------------- ---------------------------------------------
       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.

About this task

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.

Preempting 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. When an operation is preempted, it is canceled.

You can use the **QUERY MOUNT** command to see the status of the volume for the mount point.

By default, preemption is enabled on the server. To disable preemption, specify NOPREEMPT in the server options file. If you specify this option, the **BACKUP DB** command, and the export and import commands are the only operations that can preempt other operations.

Mount point preemption

If a high-priority operation requires a mount point that is in a specific device class and all the mount points in the device class are in use, the high-priority operation can preempt a mount point from a lower-priority operation.

Mount points can be preempted only when the device class of the operation preempting and the operation that is being preempted is the same.

The following high-priority operations can preempt other operations for a mount point.

- Back up database operations
- Retrieve, restore, or HSM recall operations that are initiated by clients
- Restore operations by using a remote data mover
- Export operations
- Import operations
- Backup set generation operations

The following server operations cannot preempt other operations or be preempted:

- Audit a volume
- Restore data from a copy or active-data pool
- Prepare a recovery plan file
- Store data by using a remote data mover

The following operations can be preempted and are listed in order of priority, from highest priority to lowest priority. The server selects the lowest priority operation to preempt, for example, identify duplicates.

- Replicate nodes
- Back up to a copy storage pool
- Copy active data to an active data pool
- Move data on a storage pool volume
- Migrate data from disk to sequential media
- Migrate data from sequential media to sequential media
- Back up, archive, or HSM migrate operations that are initiated by clients
- Reclaims volumes in a sequential-access storage pool
- Identify duplicates

Volume access preemption

If a high-priority operation requires access to a specific volume and that volume is in use, the high-priority operation can preempt the lower-priority operation for that volume.

For example, if a restore request requires access to a volume in use by a reclamation operation and a drive is available, the reclamation operation is canceled.

The following high-priority operations can preempt operations for access to a specific volume.

- Back up database operations
- Retrieve, restore, or HSM recall operations that are initiated by clients
- Restore operations by using a remote data mover
- Export operations
- Import operations
- Backup set generation operations

The following operations cannot preempt other operations or be preempted:

- Audit volume
- Restore data from a copy or active-data pool
- Prepare a recovery plan
- Store data by using a remote data mover

The following operations can be preempted, and are listed in order of priority, from highest priority to lowest priority. The server selects the lowest priority operation to preempt, for example, identify duplicates.

- Replicate nodes
- Back up to a copy storage pool
- Copy active data to an active data pool
- Move data on a storage pool volume
- Migrate data from disk to sequential media
- Migrate data from sequential media to sequential media
- Back up, archive, or HSM migrate data that is initiated by client
- Reclaim volumes in a sequential-access storage pool
- Identify duplicates

Changing names in Tivoli Storage Manager

When you change the server name, it can affect communication from the server to backup-archive nodes. When you change the host name for the server, it can affect the connection between the database and the server.

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 issue the QUERY STATUS command to see the name of the server.

To specify the server name you must have system privileges. For example, to change the server name to WELLS_DESIGN_DEPT., 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 682 for details. On a network where clients connect to multiple servers, it is preferable that all 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.

You can change the server name with the SET SERVERNAME command. But you might have unfortunate results that, according to the platform, can vary. 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 you take specific steps to ensure that the database continues to function properly. If you do not complete the extra steps, you can receive errors when you are starting the server.

Procedure

Change the host name by completing the following steps:

1. Stop any Tivoli Storage Manager servers that are running on the system.
2. Change the host name by using the procedures that are defined for your operating system.
3. From the root user ID on the system, issue the following command:

   ```
   db2set -g DB2SYSTEM=newhostname
   ```

where *newhostname* is the new host name for the server.

Tip: The **db2set** command is in the /opt/tivoli/tsm/db2/adm directory.

4. Verify that the DB2SYSTEM value was changed by issuing the following command:

   ```
   db2set -all
   ```

   This command shows all configuration settings that are used by the database.

5. In the *instance directory*/sqllib directory, locate the db2nodes.cfg file. The file contains an entry that shows the previous host name, for example:

   ```
   0 tsmmon TSMMON 0
   ```

   a. Update the entry with the new host name. The entry is similar to the following entry:

      ```
      0 tsmnew newhostname 0
      ```

   b. Save and close the changed file.

Add or update server options

If you have system privileges, you can add or update server options in the dsmserv.opt file.

To add or update server options, use the **SETOPT** command to edit the dsmserv.opt file.

For information about editing the server options file, refer to the *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, 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.

Deleting information about volume history

Periodically delete outdated information from the volume history file.

About this task

The contents of the volume history file are created by using the volume history table in the server database. When opening a volume, the server might check the table to determine whether the volume is already used. If the table is large, it can take a long time to search. Other sessions or processes, such as backups and other processes that use multiple sequential volumes, can be delayed due to locking.

For example, if you keep backups for seven days, information older than seven days is not needed. If 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.

Procedure

To delete volume history, issue the **DELETE VOLHISTORY** command. For example, to delete volume history that is seven days old or older, issue the following command:

```
delete volhistory type=all todate=today-8
```

When deleting information about volume history, keep in mind the following guidelines:

- Ensure that you delete volume history entries such as STGNEW, STGDELETE, and STGREUSE that are older than the oldest database backup that is required to perform a point-in-time database restore. If necessary, you can delete other types of entries.
- Existing volume history files are *not* automatically updated with the **DELETE VOLHISTORY** command.
- Do not delete information about sequential volumes until you no longer need that information. For example, do not delete information about the reuse of storage volumes unless you backed up the database after the time that was specified for the delete operation.
- Do not delete the volume history for database backup or export volumes that are stored in automated libraries unless you want to return the volumes to scratch status. When the **DELETE VOLHISTORY** command removes information for such volumes, the volumes automatically return to scratch status. The volumes are then available for reuse by the server and the information stored on them can be overwritten.
- To ensure that you have a backup from which to recover, you cannot remove the most current database snapshot entry by deleting volume history. Even if a more current, standard database backup exists, the latest database snapshot is not deleted.
- To display volume history, issue the **QUERY VOLHISTORY** command. For example, to display volume history up to yesterday, issue the following command:

  ```
  query volhistory enddate=today-1
  ```

Results

DRM: DRM automatically expires database backup series and deletes the volume history entries.

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.

About this task

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 18. 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.

About this task

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:

Procedure

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*.

© Copyright IBM Corp. 1993, 2013

What to do next

See the following topics for more information:

Concepts:
"Automating a basic administrative command schedule"
"Tailoring schedules" on page 611
"Copying schedules" on page 614
"Deleting schedules" on page 615
"Managing scheduled event records" on page 615
"Tivoli Storage Manager server scripts" on page 616
"Using macros" on page 627

Automating a basic administrative command schedule

You can set up a basic administrative command schedule by using Tivoli Storage Manager defaults.

Notes:

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 Tivoli Storage Manager activity log" on page 789.
2. You cannot schedule MACRO or QUERY ACTLOG commands.

To later update or tailor your schedules, see "Tailoring schedules" on page 611.

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.

About this task

Include the following parameters when you define a schedule with the **DEFINE SCHEDULE** command:

- Specify the administrative command to be issued (CMD=).
- Specify whether the schedule is activated (ACTIVE=).

Issue the following example command:

```
define schedule backup_archivepool type=administrative
cmd='backup stgpool archivepool recoverypool' active=yes
```

To receive the following results:

- 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.

About this task

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-     09/04/2002 14:08:11        1 H      1 D    Any
     POOL
```

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 615.

Tailoring schedules

To control your schedules more precisely, specify values for the schedule parameters instead of accepting the default settings when you define or update schedules.

You can specify the following values when you issue the **DEFINE SCHEDULE** or **UPDATE SCHEDULE** command:

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, 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. Issue 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 615.

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.

About this task

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 startime=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
startime=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 auxilarypool'
active=yes startime=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
startime=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.

About this task

When you copy a schedule, Tivoli Storage Manager copies the following information:

- A description of the schedule
- All parameter values from the original schedule

Procedure

To copy the BACKUP_ARCHIVEPOOL administrative schedule and name the new schedule BCKSCHED, issue the following command:

```
copy schedule backup_archivepool bcksched type=administrative
```

Deleting schedules

You can delete the ENGBKUP administrative schedule.

About this task

Procedure

To delete the administrative schedule ENGBKUP, issue the following command:

```
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.

About this task

- 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 77 on page 616 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-     Completed
                                                  VEPOOL
```

Figure 77. 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.

About this task

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
```

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 Tivoli Storage Manager scripts” on page 787.

The administrator can 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.

About this task

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 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.

Restriction: You cannot redirect the output of a command within a Tivoli Storage Manager script. Instead, run the script and then specify command redirection. For example, to direct the output of **script1** to the `c:\temp\test.out` directory, run the script and specify command redirection as in the following example:

```
run script1 > c:\temp\test.out
```

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 623.

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.

About this task

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.

Before you begin

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.

Procedure

Complete 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**.

What to do next

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.

Before you begin

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.

Procedure

Complete 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.

What to do next

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 in a script serially, in parallel, or serially and in parallel by using the **SERIAL** or **PARALLEL** script commands in the **COMMAND_LINE** parameter of **DEFINE** and **UPDATE SCRIPT**. You can run multiple commands in parallel and wait for them to complete before proceeding to the next command.

About this task

Running commands serially in a script ensures that any preceding commands are complete before proceeding and ensures that any following commands are run serially. When a script starts, all commands are run serially until a **PARALLEL** command is encountered. Multiple commands running in parallel and accessing common resources, such as tape drives, can run serially.

Script return codes remain the same before and after a **PARALLEL** command is run. When a **SERIAL** command is encountered, the script return code is set to the maximum return code from any previous commands run in parallel.

When using server commands that support the **WAIT** parameter after a **PARALLEL** command, the behavior is as follows:

- If you specify (or use the default) **WAIT=NO**, a script does not wait for the completion of the command when a subsequent **SERIAL** command is encountered. The return code from that command reflects processing only up to the point that the command starts a background process. The final return code from the command is not available to your script.
- If you specify **WAIT=YES**, your script waits for the completion of the command when a subsequent **SERIAL** command is encountered. The return code from that command reflects processing for the entire command.

In most cases, you can use **WAIT=YES** on commands that are run in parallel.

The following example illustrates how the **PARALLEL** command is used to back up, 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, relcaim 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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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.

About this task

The following is an example of querying the SRTL2 script and directing the output to newscript.script:

```
query script srtl2 format=raw outputfile=newscript.script
```

You can then edit the newscript.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.

About this task

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.

About this task

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.

About this task

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.

About this task

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."
- Comments

  For more information on comments, see "Writing comments in a macro" on page 628.
- Continuation characters

  For more information on continuation characters, see "Using continuation characters" on page 628.
- Variables

  For more information on variables, see "Using substitution variables in a macro" on page 629.

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 630.

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.

About this task

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."

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.

About this task

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.

About this task

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" on page 629).

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.

About this task

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.

About this task

If the macro does not contain substitution variables (such as the REG.MAC macro described in the "Writing commands in a macro" on page 627), 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 629), 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 600.

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 19. 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.

About this task

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 642
"Estimating recovery log space requirements" on page 646
"Monitoring the database and recovery log" on page 661
"Increasing the size of the database" on page 662
"Reducing the size of the database" on page 666
"Increasing the size of the active log" on page 668
"Backing up the server database" on page 858
"Restoring the database" on page 888
"Moving the database and recovery log on a server" on page 669
"Adding optional logs after server initialization" on page 674
"Transaction processing" on page 675

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.

Tivoli Storage Manager version 6.3 is installed with the IBM DB2 database application. Users who are experienced DB2 administrators can choose to perform advanced SQL queries and use DB2 tools to monitor the database. However, do not use DB2 tools to change DB2 configuration settings from those settings that are preset by Tivoli Storage Manager. Do not alter the DB2 environment for Tivoli Storage Manager in other ways, such as with other products. The Tivoli Storage Manager Version 6.3 server was built and tested with the data definition language (DDL) and database configuration that Tivoli Storage Manager deploys.

© Copyright IBM Corp. 1993, 2013

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.

Ensure that you do not do any of the following:

- 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.
- 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.
- Use other interfaces to make changes to the server database.

Database: Overview

The database does not store client data; it points to the locations of the client files in the storage pools. The Tivoli Storage Manager database contains information about the Tivoli Storage Manager server. The database also contains information about the data that is managed by the Tivoli Storage Manager server.

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)
- Intermediate results for queries

The maximum size of the Tivoli Storage Manager database is 4 TB.

The database can be distributed across up to 128 directories. 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 by using hardware mirroring, such as Redundant Array of Independent Disks (RAID) 5.

Attention: If the database is unusable, the Tivoli Storage Manager server is unavailable. You must backup the database to ensure that data that is managed by the server can be recovered. Encrypt sensitive data by using the Tivoli Storage Manager client or a storage device, unless the storage media is physically secured. Security can be compromised even if data is not recovered. If a database is lost and cannot be recovered, it might be difficult or impossible to recover data that is managed by that server. Fragments of data or complete files might be read from storage pool volumes that are not encrypted. See Part 6, “Protecting the server,” on page 819 for steps that you can take to protect your database.

The database manager manages database volumes, and there is no need to format them. 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 that is used since the last backup, which triggers a full database backup
- The active log utilization ratio, which triggers an incremental database backup

The database manager automatically backs up the database when the archive log usage is 80% full.

Automatic statistics collection
Automatic statistics collection helps to improve database performance by collecting up-to-date table statistics. The database manager determines which statistics must be updated.

Automatic database reorganization
Reorganization of table data can be initiated by the server, or by DB2. If server-initiated reorganization is enabled, based on table activity, the server analyzes selected database tables and their indexes to determine when reorganization is required. The database manager runs a reorganization while server operations continue.

Attention: If reorganization by DB2 is enabled, DB2 controls the reorganization process. Reorganization by DB2 is not recommended.

Multiple data streams for database backup and restore
Using a single data stream to back up databases of multiple terabytes can take many hours. It can also affect the administrator's ability to schedule database backups effectively. The time to recover the server by using a single data stream might not be enough to meet disaster recovery objectives for the server. The Tivoli Storage Manager server provides a multiple data stream capability for backups and restores.

SQL queries
The database makes more sophisticated SQL queries on the data possible. To take advantage of these functions, you must use SQL to develop new tools and create 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.

Related tasks:

"Scheduling table and index reorganization" on page 666

Connecting the server to the database with TCP/IP

The default configuration for the Tivoli Storage Manager V6.1 and V6.2 servers is to use interprocess communications (IPC) to communicate with the database manager. With Tivoli Storage Manager V6.3, the server can also connect to the database manager by using TCP/IP.

Using TCP/IP to communicate with DB2 can greatly extend the number of concurrent connections. The TCP/IP connection is part of the default configuration. When the Tivoli Storage Manager V6.3 server is started for the first time, it inspects the current configuration of the DB2 instance. It then makes any necessary changes to ensure that both IPC and TCP/IP can be used to communicate with the database manager. Any changes are made only as needed. For example, if the TCP/IP node exists and has the correct configuration, it is not changed. If the node was cataloged but has an incorrect IP address or port, it is deleted and replaced by a node having the correct configuration.

When cataloging the remote database, the Tivoli Storage Manager server generates a unique alias name based on the name of the local database. By default, a remote database alias of TSMAL001 is created to go with the default database name of TSMDB1.

Tip: Tivoli Storage Manager disables the TCP/IP connections if it cannot find an alias in the range TSMAL001-TSMAL999 that is not already in use.

By default, the Tivoli Storage Manager server uses IPC to establish connections for the first two connection pools, with a maximum of 480 connections for each pool. After the first 960 connections are established, the Tivoli Storage Manager server uses TCP/IP for any additional connections.

You can use the `DBMTCPPORT` server option to specify the port on which the TCP/IP communication driver for the database manager waits for requests for client sessions. The port number must be reserved for use by the database manager.

If Tivoli Storage Manager cannot connect to the database by using TCP/IP, it issues an error message and halts. The administrator must determine the cause of the problem and to correct it before restarting the server. The server verifies that it can connect by using TCP/IP at startup even if it is configured to initially favor IPC connections over TCP/IP connections.

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 when you restart the Tivoli Storage Manager or the database, and is required if you must restore the database.

When you issue a command to make changes, the changes are committed to the database to complete. A committed change is permanent and cannot be rolled back. 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 reapplied and committed again.

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.

Ensure that the recovery log has enough space. 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.

Related concepts:

"Transaction processing" on page 675

"Active log"

Recovery log mode

The recovery log mode for the Tivoli Storage Manager is the roll-forward mode. The recovery log stores data that is required to back up a restored database to the most recent committed transaction. You must have the most recent recovery logs to use the roll-forward mode.

Changes to the database are recorded in the recovery log to maintain a consistent database image. You can restore the server to the latest time possible, by using the active and archive log files, which are included in database backups.

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. You can also set these values by specifying the **ACTIVELOGDIRECTORY** and the **ACTIVELOGSIZE** parameters of the **DSMSERV FORMAT** or **DSMSERV LOADFORMAT** utilities. 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 668. To change the location of the active log directory, see "Moving only the active log, archive log, or archive failover log" on page 671.

For information about the space required for the active log directory, see "Active and archive log space" on page 646.

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 **DSMSERV LOADFORMAT** utility instead of the wizard to configure the server, specify the **MIRRORLOGDIRECTORY** parameter. If the log mirror directory is not created at that time, you can create it later by specifying the `MIRRORLOGDIRECTORY` 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 and 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 set the location of the archive log directory during initial configuration of a new or upgraded server. You can also specify the **ARCHLOGDIRECTORY** parameter of the **DSMSERV FORMAT** or **DSMSERV LOADFORMAT** utility. The location of the log can be changed later.

Related concepts:

"Active and archive log space" on page 646

Related tasks:

"Moving only the active log, archive log, or archive failover log" on page 671

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. You can also specify its location with the **ARCHFAILOVERLOGDIRECTORY** parameter of the **DSMSERV FORMAT** or **DSMSERV LOADFORMAT** utility. If it is not created through the utilities, it can be created later by specifying the ARCHFAILOVERLOGDIRECTORY option in the server options file, dsmserv.opt. See "Adding optional logs after server initialization" on page 674 for details.

For information about the space required for the log, see "Archive failover log space" on page 659.

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. It is important to monitor the archive log directory to ensure that there is space in the active log.

The Tivoli Storage Manager database manager can move active log files to the failover archive log. The database manager automatically manages the space that is available to the directories as database space. The database manager determines when database backups are required and automatically initiates them.

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.

Setting the user data limit for the database

You can set the user data limit and private memory for the database to ensure that private memory is not exhausted. Exhausting private memory can result in errors, less than optimal performance, and instability.

About this task

The user data limit that is displayed when you issue the ulimit -d command is the soft user data limit. It is not necessary to set the hard user data limit for DB2. The default soft user data limit is 128 MB. This is equivalent to the value of 262,144 512-byte units as set in /etc/security/limits folder, or 131,072 KB units as displayed by the ulimit -d command. This setting limits private memory usage to about one half of what is available in the 256 MB private memory segment available for a 32-bit process on AIX.

Note: A DB2 server instance cannot make use of the Large Address Space or of very large address space AIX 32-bit memory models due to shared memory requirements. On some systems, for example those requiring large amounts of sort memory for performance, it is best to increase the user data limit to allow DB2 to allocate more than 128 MB of memory in a single process.

You can set the user data memory limit to "unlimited" (a value of "-1"). This setting is not recommended for 32-bit DB2 because it allows the data region to overwrite the stack, which grows downward from the top of the 256 MB private memory segment. The result would typically be to cause the database to end abnormally. It is, however, an acceptable setting for 64-bit DB2 because the data region and stack are allocated in separate areas of the very large address space available to 64-bit AIX processes.

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 is 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 isolated, 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.

Capacity planning

Capacity planning for Tivoli Storage Manager includes managing resources such as the database and recovery log. To maximize resources as part of capacity planning, you must estimate space requirements for the database and the recovery log.

Procedure

For information about the benefits of deduplication and guidance on how to make effective use of the Tivoli Storage Manager deduplication feature, see Optimizing Performance.

Estimating database space requirements

To estimate space requirements for the database, you can use the maximum number of files that can be in server storage at one time or you can use storage pool capacity.

About this task

Consider using at least 25 GB for the initial database space. Provision file system space appropriately. A database size of 25 GB is adequate for a test environment or a library-manager-only environment. For a production server supporting client workloads, the database size is expected to be larger. If you use random-access disk (DISK) storage pools, more database and log storage space is needed than for sequential-access storage pools.

The maximum size of the Tivoli Storage Manager database is 4 TB.

For information about sizing the database in a production environment that is based on the number of files and on storage pool size, see the following topics.

Estimating database space requirements based on the number of files

If you can estimate the maximum number of files that might be in server storage at a time, you can use that number to estimate space requirements for the database.

About this task

To estimate space requirements that is based on the maximum number of files in server storage, use the following guidelines:

- 600 - 1000 bytes for each stored version of a file, including image backups.

  Restriction: The guideline does not include space that is used during data deduplication.
- 100 - 200 bytes for each cached file, copy storage pool file, active-data pool file, and deduplicated file.
- Additional space is required for database optimization to support varying data-access patterns and to support server back-end processing of the data. The amount of extra space is equal to 50% of the estimate for the total number of bytes for file objects.

In the following example for a single client, the calculations are based on the maximum values in the preceding guidelines. The examples do not take into account that you might use file aggregation. In general, when you aggregate small files, it reduces the amount of required database space. File aggregation does not affect space-managed files.

Procedure

1. Calculate the number of file versions. Add each of the following values to obtain the number of file versions:
   a. Calculate the number of backed-up files. For example, as many as 500,000 client files might be backed up at a time. In this example, storage policies are set to keep up to three copies of backed up files:

      ```
      500,000 files * 3 copies = 1,500,000 files
      ```

   b. Calculate the number of archive files. For example, as many as 100,000 client files might be archived copies.

c. Calculate the number of space-managed files. For example, as many as 200,000 client files might be migrated from client workstations.

Using 1000 bytes per file, the total amount of database space that is required for the files that belong to the client is 1.8 GB:

```
(1,500,000 + 100,000 + 200,000) * 1000 = 1.8 GB
```

2. Calculate the number of cached files, copy storage-pool files, active-data pool files, and deduplicated files:
   a. Calculate the number of cached copies. For example, caching is enabled in a 5 GB disk storage pool. The high migration threshold of the pool is 90% and the low migration threshold of the pool is 70%. Thus, 20% of the disk pool, or 1 GB, is occupied by cached files.

      If the average file size is about 10 KB, approximately 100,000 files are in cache at any one time:

      ```
      100,000 files * 200 bytes = 19 MB
      ```

   b. Calculate the number of copy storage-pool files. All primary storage pools are backed up to the copy storage pool:

      ```
      (1,500,000 + 100,000 + 200,000) * 200 bytes = 343 MB
      ```

   c. Calculate the number of active storage-pool files. All the active client-backup data in primary storage pools is copied to the active-data storage pool. Assume that 500,000 versions of the 1,500,000 backup files in the primary storage pool are active:

      ```
      500,000 * 200 bytes = 95 MB
      ```

   d. Calculate the number of deduplicated files. Assume that a deduplicated storage pool contains 50,000 files:

      ```
      50,000 * 200 bytes = 10 MB
      ```

   Based on the preceding calculations, about 0.5 GB of extra database space is required for the client's cached files, copy storage-pool files, active-data pool files, and deduplicated files.
3. Calculate the amount of extra space that is required for database optimization. To provide optimal data access and management by the server, extra database space is required. The amount of extra database space is equal to 50% of the total space requirements for file objects.

   ```
   (1.8 + 0.5) * 50% = 1.2 GB
   ```

4. Calculate the total amount of database space that is required for the client. The total is approximately 3.5 GB:

   ```
   1.8 + 0.5 + 1.2 = 3.5 GB
   ```

5. Calculate the total amount of database space that is required for all clients. If the client that was used in the preceding calculations is typical and you have 500 clients, for example, you can use the following calculation to estimate the total amount of database space that is required for all clients:

   ```
   500 * 3.5 = 1.7 TB
   ```

Results

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. Periodically monitor your database and adjust its size as necessary.

What to do next

During normal operations, the Tivoli Storage Manager server might require temporary database space. This space is needed for the following reasons:

- To hold the results of sorting or ordering that are not already being kept and optimized in the database directly. The results are temporarily held in the database for processing.
- To give administrative access to the database through one of the following methods:
  - A DB2 open database connectivity (ODBC) client
  - An Oracle Java database connectivity (JDBC) client
  - Structured Query Language (SQL) to the server from an administrative-client command line

Consider using an extra 50 GB of temporary space for every 500 GB of space for file objects and optimization. See the guidelines in the following table. In the example that is used in the preceding step, a total of 1.7 TB of database space is required for file objects and optimization for 500 clients. Based on that calculation, 200 GB is required for temporary space. The total amount of required database space is 1.9 TB.

Database size	Minimum temporary-space requirement
< 500 GB	50 GB
≥ 500 GB and < 1 TB	100 GB
≥ 1 TB and < 1.5 TB	150 GB
≥ 1.5 and < 2 TB	200 GB
≥ 2 and < 3 TB	250 - 300 GB
≥ 3 and < 4 TB	350 - 400 GB

Estimating database space requirements based on storage pool capacity

To estimate database space requirements based on storage pool capacity, use a ratio of 1 - 5%. For example, if you require 200 TB of storage pool capacity, the size of your database is expected to be 2 - 10 TB. As a general rule, make your database as large as possible to prevent running out of space. If you run out of database space, server operations and client-store operations can fail.

Database space required when running database operations

The database manager of the Tivoli Storage Manager server manages and allocates system memory and disk space for the database. The amount of database space that the system requires depends on the amount of system memory that is available and the server workload.

For example, expiration processing can use a large amount of database space. If there is not enough system memory in the database to store the files identified for expiration, some of the data is allocated to temporary disk space. During expiration processing, if a node or file space is selected that is too large to process, the database manager cannot sort the data.

To run database operations, consider adding more database space for the following scenarios:

- The database has a small amount of space and the server operation that requires temporary space uses the remaining free space.
- The file spaces are large, or the file spaces have a policy assigned to it that creates many file versions.
- The Tivoli Storage Manager server must run with limited memory.
- An `out of database space` error is displayed when you deploy a Tivoli Storage Manager V6 server.

Attention: Do not install or upgrade to a different version, release, or fix pack of DB2 software because doing so can damage the database.

The database manager and temporary space

The Tivoli Storage Manager server database manager manages and allocates system memory and disk space for the database. The amount of database space you require depends on the amount of system memory available and the server workload.

The database manager sorts data in a specific sequence, as per the SQL statement that you issue to request the data. Depending on the workload on the server, and if there is more data than the database manager can manage, the data (that is ordered in sequence) is allocated to temporary disk space. Data is allocated to temporary disk space when there is a large result set. The database manager dynamically manages the memory used when data is allocated to temporary disk space.

For example, expiration processing can produce a large result set. If there is not enough system memory on the database to store the result set, some of the data is allocated to temporary disk space. During expiration processing, if a node or file space are selected that are too large to process, the database manager does not have enough memory to sort the data.

To run database operations, consider adding more database space for the following scenarios:

- The database has a small amount of space and the server operation that requires temporary space uses the remaining free space.
- The file spaces are large, or the file spaces has a policy assigned to it which creates many file versions.
- The Tivoli Storage Manager server must run with limited memory. The database uses the Tivoli Storage Manager server main memory to run database operations. However, if there is insufficient memory available, the Tivoli Storage Manager server allocates temporary space on disk to the database. For example, if 10G of memory is available and database operations require 12G of memory, the database uses temporary space.
- An `out of database space` error is displayed when you deploy a Tivoli Storage Manager V6 server. Monitor the server activity log for messages related to database space.

Important: Do not change the DB2 software that is installed with the Tivoli Storage Manager installation packages and fix packs. Do not install or upgrade to a different version, release, or fix pack, of DB2 software to avoid damage to the database.

Estimating recovery log space requirements

In Tivoli Storage Manager, the term *recovery log* comprises the active log, the archive log, the active log mirror, and the archive failover log. The amount of space that you require for the recovery log depends on various factors, including, for example, the amount of client activity with the server.

Active and archive log space

When you estimate space requirements for active and archive logs, include some extra space for contingencies such as occasional heavy workloads and failovers.

In Tivoli Storage Manager servers V6.1 and later, the active log can be a maximum size of 128 GB. The archive log size is limited to the size of the file system that it is installed on.

Use the following general guidelines when you estimate the size of the active log:

- The suggested starting size for the active log is 16 GB.
- Ensure that the active log is at least large enough for the amount of concurrent activity that the server typically handles. As a precaution, try to anticipate the largest amount of work that the server manages at one time. Provision the active log with extra space that can be used if needed. Consider using 20% of extra space.
- Monitor used and available active log space. Adjust the size of the active log as needed, depending upon factors such as client activity and the level of server operations.
- Ensure that the directory that holds the active log is as large as, or larger than, the size of the active log. A directory that is larger than the active log can accommodate failovers, if they occur.
- Ensure that the file system that contains the active log directory has at least 8 GB of free space for temporary log movement requirements.

The suggested starting size for the archive log is 48 GB.

The archive log directory must be large enough to contain the log files that are generated since the previous full backup. For example, if you perform a full backup of the database every day, the archive log directory must be large enough to hold the log files for all the client activity that occurs during 24 hours. To recover space, the server deletes obsolete archive log files after a full backup of the database. If the archive log directory becomes full and a directory for archive failover logs does not exist, log files remain in the active log directory. This condition can cause the active log directory to fill up and stop the server. When the server restarts, some of the existing active-log space is released.

After the server is installed, you can monitor archive log utilization and the space in the archive log directory. If the space in the archive log directory fills up, it can cause the following problems:

- The server is unable to perform full database backups. Investigate and resolve this problem.
- Other applications write to the archive log directory, exhausting the space that is required by the archive log. Do not share archive log space with other applications including other Tivoli Storage Manager servers. Ensure that each server has a separate storage location that is owned and managed by that specific server.

For guidance about the layout and tuning of the active log and archive log, see Optimizing Performance.

Related tasks:

"Increasing the size of the active log" on page 668

Example: Estimating active and archive log sizes for basic client-store operations:

Basic client-store operations include backup, archive, and space management. Log space must be sufficient to handle all store transactions that are in progress at one time.

To determine the sizes of the active and archive logs for basic client-store operations, use the following calculation:

```
number of clients  x  files stored during each transaction
  x  log space needed for each file
```

This calculation is used in the example in the following table.

Table 52. Basic client-store operations

Item	Example values	Description
Maximum number of client nodes that back up, archive, or migrate files concurrently at any time	300	The number of client nodes that back up, archive, or migrate files every night.
Files stored during each transaction	4096	The default value of the server option TXNGROUPMAX is 4096.
Log space that is required for each file	3053 bytes	The value of 3053 bytes for each file in a transaction represents the log bytes that are needed when backing up files from a Windows client where the file names are 12 - 120 bytes. This value is based on the results of tests performed under laboratory conditions. The tests consisted of backup-archive clients performing backup operations to a random-access disk (DISK) storage pool. DISK pools result in more log use than sequential-access storage pools. Consider a value larger than 3053 bytes if the data being stored has file names that are longer than 12 - 120 bytes.
Active log: Suggested size	19.5 GB [1]	Use the following calculation to determine the size of the active log. One GB equals 1,073,741,824 bytes. (300 clients x 4096 files stored during each transaction x 3053 bytes for each file) ÷ 1,073,741,824 bytes = 3.5 GB Increase that amount by the suggested starting size of 16 GB: 3.5 + 16 = 19.5 GB

Table 52. Basic client-store operations (continued)

Item	Example values	Description

Archive log: Suggested size	58.5 GB [1]	Because of the requirement to be able to store archive logs across three server database-backup cycles, multiply the estimate for the active log by 3 to estimate the total archive log requirement.
		
		`3.5 x 3 = 10.5 GB`
		
		Increase that amount by the suggested starting size of 48 GB:
		
		`10.5 + 48 = 58.5 GB`

[1] The example values in this table are used only to illustrate how the sizes for active logs and archive logs are calculated. In a production environment that does not use deduplication, 16 GB is the suggested minimum size for an active log. The suggested minimum size for an archive log in a production environment that does not use deduplication is 48 GB. If you substitute values from your environment and the results are larger than 16 GB and 48 GB, use your results to size the active log and archive log. Monitor your logs and adjust their size if necessary.		

Example: Estimating active and archive log sizes for clients that use multiple sessions:

If the client option `RESOURCEUTILIZATION` is set to a value that is greater than the default, the concurrent workload for the server increases.

To determine the sizes of the active and archive logs when clients use multiple sessions, use the following calculation:

```
number of clients  x  sessions for each client  x  files stored
  during each transaction  x  log space needed for each file
```

This calculation is used in the example in the following table.

Table 53. Multiple client sessions

Item	Example values		Description
Maximum number of client nodes that back up, archive, or migrate files concurrently at any time	300	1000	The number of client nodes that back up, archive, or migrate files every night.
Possible sessions for each client	3	3	The setting of the client option `RESOURCEUTILIZATION` is larger than the default. Each client session runs a maximum of three sessions in parallel.
Files stored during each transaction	4096	4096	The default value of the server option `TXNGROUPMAX` is 4096.

Table 53. Multiple client sessions (continued)

Item	Example values		Description
Log space that is required for each file	3053	3053	The value of 3053 bytes for each file in a transaction represents the log bytes needed when backing up files from a Windows client where the file names are 12 - 120 bytes. This value is based on the results of tests performed under laboratory conditions. Tests consisted of clients performing backup operations to a random-access disk (DISK) storage pool. DISK pools result in more log use than sequential-access storage pools. Consider a value larger than 3053 bytes if the data being stored has file names that are longer than 12 - 120 bytes.

Active log: Suggested size	26.5 GB [1]	51 GB [1]	The following calculation was used for 300 clients. One GB equals 1,073,741,824 bytes.
			
			`(300 clients x 3 sessions for each client x 4096 files stored during each transaction x 3053 bytes for each file) ÷ 1,073,741,824 = 10.5 GB`
			
			Increase that amount by the suggested starting size of 16 GB:
			
			`10.5 + 16 = 26.5 GB`
			
			The following calculation was used for 1000 clients. One GB equals 1,073,741,824 bytes.
			
			`(1000 clients x 3 sessions for each client x 4096 files store during each transaction x 3053 bytes for each file) ÷ 1,073,741,824 = 35 GB`
			
			Increase that amount by the suggested starting size of 16 GB:
			
			`35 + 16 = 51 GB`

Archive log: Suggested size	79.5 GB [1]	153 GB [1]	Because of the requirement to be able to store archive logs across three server-database backup cycles, the estimate for the active log is multiplied by 3:
			
			`10.5 x 3 = 31.5 GB`
			
			`35 x 3 = 105 GB`
			
			Increase those amounts by the suggested starting size of 48 GB:
			
			`31.5 + 48 = 79.5 GB`
			
			`105 + 48 = 153 GB`

[1] The example values in this table are used only to illustrate how the sizes for active logs and archive logs are calculated. In a production environment that does not use deduplication, 16 GB is the suggested minimum size for an active log. The suggested minimum size for an archive log in a production environment that does not use deduplication is 48 GB. If you substitute values from your environment and the results are larger than 16 GB and 48 GB, use your results to size the active log and archive log. Monitor your active log and adjust its size if necessary.			

Example: Estimating active and archive log sizes for simultaneous write operations:

If client backup operations use storage pools that are configured for simultaneous write, the amount of log space that is required for each file increases.

The log space that is required for each file increases by about 200 bytes for each copy storage pool that is used for a simultaneous write operation. In the example in the following table, data is stored to two copy storage pools in addition to a primary storage pool. The estimated log size increases by 400 bytes for each file. If you use the suggested value of 3053 bytes of log space for each file, the total number of required bytes is 3453.

This calculation is used in the example in the following table.

Table 54. Simultaneous write operations

Item	Example values	Description
Maximum number of client nodes that back up, archive, or migrate files concurrently at any time	300	The number of client nodes that back up, archive, or migrate files every night.
Files stored during each transaction	4096	The default value of the server option `TXNGROUPMAX` is 4096.
Log space that is required for each file	3453 bytes	3053 bytes plus 200 bytes for each copy storage pool. The value of 3053 bytes for each file in a transaction represents the log bytes that are needed when backing up files from a Windows client where the file names are 12 - 120 bytes. This value is based on the results of tests performed under laboratory conditions. The tests consisted of backup-archive clients performing backup operations to a random-access disk (DISK) storage pool. DISK pools result in more log use than sequential-access storage pools. Consider a value larger than 3053 bytes if the data being stored has file names that are longer than 12 - 120 bytes.

Active log: Suggested size	20 GB [1]	Use the following calculation to determine the size of the active log. One GB equals 1,073,741,824 bytes.
		
		`(300 clients x 4096 files stored during each transaction x 3453 bytes for each file) ÷ 1,073,741,824 bytes = 4.0 GB`
		
		Increase that amount by the suggested starting size of 16 GB:
		
		`4 + 16 = 20 GB`

Archive log: Suggested size	60 GB [1]	Because of the requirement to be able to store archive logs across three server database-backup cycles, multiply the estimate for the active log by 3 to estimate the archive log requirement:
		
		`4 GB  x  3  = 12 GB`
		
		Increase that amount by the suggested starting size of 48 GB:
		
		`12 + 48 = 60 GB`

Table 54. Simultaneous write operations (continued)

Item	Example values	Description
[1] The example values in this table are used only to illustrate how the sizes for active logs and archive logs are calculated. In a production environment that does not use deduplication, 16 GB is the suggested minimum size for an active log. The suggested minimum size for an archive log in a production environment that does not use deduplication is 48 GB. If you substitute values from your environment and the results are larger than 16 GB and 48 GB, use your results to size the active log and archive log. Monitor your logs and adjust their size if necessary.		

Example: Estimating active and archive log sizes for basic client store operations and server operations:

Migration of data in server storage, identification processes for data deduplication, reclamation, and expiration might run concurrently with client store operations. Administrative tasks such as administrative commands or SQL queries from administrative clients can also run concurrently with client store operations. Server operations and administrative tasks that run concurrently can increase the active log space that is required.

For example, migration of files from the random-access (DISK) storage pool to a sequential-access disk (FILE) storage pool uses approximately 110 bytes of log space for each file that is migrated. For example, suppose that you have 300 backup-archive clients and each one of them backs up 100,000 files every night. The files are initially stored on DISK and then migrated to a FILE storage pool. To estimate the amount of active log space that is required for the data migration, use the following calculation. The number of clients in the calculation represents the maximum number of client nodes that back up, archive, or migrate files concurrently at any time.

```
300 clients  x 100,000 files for each client   x  110 bytes =  3.1 GB
```

Add this value to the estimate for the size of the active log that calculated for basic client store operations.

Example: Estimating active and archive log sizes under conditions of extreme variation:

Problems with running out of active log space can occur if you have many transactions that complete quickly and some transactions that take much longer to complete. A typical case occurs when many workstation or file-server backup sessions are active and a few very large database server-backup sessions are active. If this situation applies to your environment, you might need to increase the size of the active log so that the work completes successfully.

Example: Estimating archive log sizes with full database backups:

The Tivoli Storage Manager server deletes unnecessary files from the archive log only when a full database backup occurs. Consequently, when you estimate the space that is required for the archive log, you must also consider the frequency of full database backups.

For example, if a full database backup occurs once a week, the archive log space must be able to contain the information in the archive log for a full week.

The difference in archive log size for daily and full database backups is shown in the example in the following table.

Table 55. Full database backups

Item	Example values	Description
Maximum number of client nodes that back up, archive, or migrate files concurrently at any time	300	The number of client nodes that back up, archive, or migrate files every night.
Files stored during each transaction	4096	The default value of the server option `TXNGROUPMAX` is 4096.
Log space that is required for each file	3453 bytes	3053 bytes for each file plus 200 bytes for each copy storage pool. The value of 3053 bytes for each file in a transaction represents the log bytes needed when backing up files from a Windows client where the file names are 12 - 120 bytes. This value is based on the results of tests performed under laboratory conditions. Tests consisted of clients performing backup operations to a random-access disk (DISK) storage pool. DISK pools result in more log use than sequential-access storage pools. Consider a value larger than 3053 bytes if the data being stored has file names that are longer than 12 - 120 bytes.

Active log: Suggested size	20 GB [1]	Use the following calculation to determine the size of the active log. One GB equals 1,073,741,824 bytes.
		
		`(300 clients x 4096 files per transaction x 3453 bytes per file) ÷ 1,073,741,824 bytes = 4.0 GB`
		
		Increase that amount by the suggested starting size of 16 GB:
		
		`4 + 16 = 20 GB`

Archive log: Suggested size with a full database backup every day	60 GB [1]	Because of the requirement to be able to store archive logs across three backup cycles, multiply the estimate for the active log by 3 to estimate the total archive log requirement:
		
		`4 GB x 3 = 12 GB`
		
		Increase that amount by the suggested starting size of 48 GB:
		
		`12 + 48 = 60 GB`

Table 55. Full database backups (continued)

Item	Example values	Description

Archive log: Suggested size with a full database every week	132 GB [1]	Because of the requirement to be able to store archive logs across three server database-backup cycles, multiply the estimate for the active log by 3 to estimate the total archive log requirement. Multiply the result by the number of days between full database backups:
		`(4 GB x 3 ) x 7 = 84 GB`
		Increase that amount by the suggested starting size of 48 GB:
		`84 + 48 = 132 GB`

[1] The example values in this table are used only to illustrate how the sizes for active logs and archive logs are calculated. In a production environment that does not use deduplication, 16 GB is the suggested minimum size for an active log. The suggested starting size for an archive log in a production environment that does not use deduplication is 48 GB. If you substitute values from your environment and the results are larger than 16 GB and 48 GB, use your results to size the active log and archive log. Monitor your logs and adjust their size if necessary.		

Example: Estimating active and archive log sizes for data deduplication operations:

If you deduplicate data, you must consider its effects on space requirements for active and archive logs.

The following factors affect requirements for active and archive log space:

The amount of deduplicated data

The effect of data deduplication on the active log and archive log space depends on the percentage of data that is eligible for deduplication. If the percentage of data that can be deduplicated is relatively high, more log space is required.

The size and number of extents

Approximately 1,500 bytes of active log space are required for each extent that is identified by a duplicate-identification process. For example, if 250,000 extents are identified by a duplicate-identification process, the estimated size of the active log is 358 MB:

```
250,000 extents identified during each process x 1,500 bytes
 for each extent = 358 MB
```

Consider the following scenario. Three hundred backup-archive clients back up 100,000 files each night. This activity creates a workload of 30,000,000 files. The average number of extents for each file is two. Therefore, the total number of extents is 60,000,000, and the space requirement for the archive log is 84 GB:

```
60,000,000 extents x 1,500 bytes for each extent = 84 GB
```

A duplicate-identification process operates on aggregates of files. An aggregate consists of files that are stored in a given transaction, as specified by the `TXNGROUPMAX` server option. Suppose that the `TXNGROUPMAX` server option is set to the default of 4096. If the average number of extents for each file is two, the total number of extents in each aggregate is 8192, and the space required for the active log is 12 MB:

```
8192 extents in each aggregate x 1500 bytes for each extent =
   12 MB
```

The timing and number of the duplicate-identification processes

The timing and number of duplicate-identification processes also affects the size of the active log. Using the 12 MB active-log size that was calculated in the preceding example, the concurrent load on the active log is 120 MB if 10 duplicate-identification processes are running in parallel:

```
12 MB for each process x 10 processes = 120 MB
```

File size

Large files that are processed for duplicate identification can also affect the size of the active log. For example, suppose that a backup-archive client backs up an 80 GB, file-system image. This object can have a high number of duplicate extents if, for example, the files included in the file system image were backed up incrementally. For example, assume that a file system image has 1.2 million duplicate extents. The 1.2 million extents in this large file represent a single transaction for a duplicate-identification process. The total space in the active log that is required for this single object is 1.7 GB:

```
1,200,000 extents x 1,500 bytes for each extent = 1.7 GB
```

If other, smaller duplicate-identification processes occur at the same time as the duplicate-identification process for a single large object, the active log might not have enough space. For example, suppose that a storage pool is enabled for deduplication. The storage pool has a mixture of data, including many relatively small files that range from 10 KB to several hundred KB. The storage pool also has few large objects that have a high percentage of duplicate extents.

To take into account not only space requirements but also the timing and duration of concurrent transactions, increase the estimated size of the active log by a factor of two. For example, suppose that your calculations for space requirements are 25 GB (23.3 GB + 1.7 GB for deduplication of a large object). If deduplication processes are running concurrently, the suggested size of the active log is 50 GB. The suggested size of the archive log is 150 GB.

The examples in the following tables show calculations for active and archive logs. The example in the first table uses an average size of 700 KB for extents. The example in the second table uses an average size of 256 KB. As the examples show, the average deduplicate-extent size of 256 KB indicates a larger estimated size for the active log. To minimize or prevent operational problems for the server, use 256 KB to estimate the size of the active log in your production environment.

Table 56. Average duplicate-extent size of 700 KB

Item	Example values		Description
Size of largest single object to deduplicate	800 GB	4 TB	The granularity of processing for deduplication is at the file level. Therefore, the largest single file to deduplicate represents the largest transaction and a correspondingly large load on the active and archive logs.
Average size of extents	700 KB	700 KB	The deduplication algorithms use a variable block method. Not all deduplicated extents for a given file are the same size, so this calculation assumes an average size for extents.

Table 56. Average duplicate-extent size of 700 KB (continued)

Item	Example values		Description
Extents for a given file	1,198,372 bits	6,135,667 bits	Using the average extent size (700 KB), these calculations represent the total number of extents for a given object. The following calculation was used for an 800 GB object: `(800 GB ÷ 700 KB) = 1,198,372 bits` The following calculation was used for a 4 TB object: `(4 TB ÷ 700 KB) = 6,135,667 bits`
Active log: Suggested size that is required for the deduplication of a single large object during a single duplicate-identification process	1.7 GB	8.6 GB	The estimated active log space that are needed for this transaction.

Active log: Suggested total size	66 GB [1]	79.8 GB [1]	After considering other aspects of the workload on the server in addition to deduplication, multiply the existing estimate by a factor of two. In these examples, the active log space required to deduplicate a single large object is considered along with previous estimates for the required active log size.
			
			The following calculation was used for multiple transactions and an 800 GB object:
			
			`(23.3 GB + 1.7 GB) x 2 = 50 GB`
			
			Increase that amount by the suggested starting size of 16 GB:
			
			`50 + 16 = 66 GB`
			
			The following calculation was used for multiple transactions and a 4 TB object:
			
			`(23.3 GB + 8.6 GB) x 2 = 63.8 GB`
			
			Increase that amount by the suggested starting size of 16 GB:
			
			`63.8 + 16 = 79.8 GB`

Archive log: Suggested size	198 GB [1]	239.4 GB [1]	Multiply the estimated size of the active log by a factor of 3.
			
			The following calculation was used for multiple transactions and an 800 GB object:
			
			`50 GB x 3 = 150 GB`
			
			Increase that amount by the suggested starting size of 48 GB:
			
			`150 + 48 = 198 GB`
			
			The following calculation was used for multiple transactions and a 4 TB object:
			
			`63.8 GB x 3 = 191.4 GB`
			
			Increase that amount by the suggested starting size of 48 GB:
			
			`191.4 + 48 = 239.4 GB`

Table 56. Average duplicate-extent size of 700 KB (continued)

Item	Example values	Description
[1] The example values in this table are used only to illustrate how the sizes for active logs and archive logs are calculated. In a production environment that uses deduplication, 32 GB is the suggested minimum size for an active log. The suggested minimum size for an archive log in a production environment that uses deduplication is 96 GB. If you substitute values from your environment and the results are larger than 32 GB and 96 GB, use your results to size the active log and archive log. Monitor your logs and adjust their size if necessary.		

Table 57. Average duplicate-extent size of 256 KB

Item	Example values		Description
Size of largest single object to deduplicate	800 GB	4 TB	The granularity of processing for deduplication is at the file level. Therefore, the largest single file to deduplicate represents the largest transaction and a correspondingly large load on the active and archive logs.
Average size of extents	256 KB	256 KB	The deduplication algorithms use a variable block method. Not all deduplicated extents for a given file are the same size, so this calculation assumes an average extent size.

Extents for a given file	3,276,800 bits	16,777,216 bits	Using the average extent size, these calculations represent the total number of extents for a given object.
			
			The following calculation was used for multiple transactions and an 800 GB object:
			
			`(800 GB ÷ 256 KB) = 3,276,800 bits`
			
			The following calculation was used for multiple transactions and a 4 TB object:
			
			`(4 TB ÷ 256 KB) = 16,777,216 bits`

Active log: Suggested size that is required for the deduplication of a single large object during a single duplicate-identification process	4.5 GB	23.4 GB	The estimated size of the active log space that is required for this transaction.

Table 57. Average duplicate-extent size of 256 KB (continued)

Item	Example values		Description

Active log: Suggested total size	71.6 GB [1]	109.4 GB [1]	After considering other aspects of the workload on the server in addition to deduplication, multiply the existing estimate by a factor of 2. In these examples, the active log space required to deduplicate a single large object is considered along with previous estimates for the required active log size.
			
			The following calculation was used for multiple transactions and an 800 GB object:
			
			`(23.3 GB + 4.5 GB) x 2 = 55.6 GB`
			
			Increase that amount by the suggested starting size of 16 GB:
			
			`55.6 + 16 = 71.6 GB`
			
			The following calculation was used for multiple transactions and a 4 TB object:
			
			`(23.3 GB + 23.4 GB) x 2 = 93.4 GB`
			
			Increase that amount by the suggested starting size of 16 GB:
			
			`93.4 + 16 = 109.4 GB`

Archive log: Suggested size	214.8 GB [1]	328.2 GB [1]	The estimated size of the active log multiplied by a factor of 3.
			
			The following calculation was used for an 800 GB object:
			
			`55.6 GB x 3 = 166.8 GB`
			
			Increase that amount by the suggested starting size of 48 GB:
			
			`166.8 + 48 = 214.8 GB`
			
			The following calculation was used for a 4 TB object:
			`93.4 GB x 3 = 280.2 GB`
			
			Increase that amount by the suggested starting size of 48 GB:
			
			`280.2 + 48 = 328.2 GB`

[1] The example values in this table are used only to illustrate how the sizes for active logs and archive logs are calculated. In a production environment that uses deduplication, 32 GB is the suggested minimum size for an active log. The suggested minimum size for an archive log in a production environment that uses deduplication is 96 GB. If you substitute values from your environment and the results are larger than 32 GB and 96 GB, use your results to size the active log and archive log. Monitor your logs and adjust their size if necessary.			

Archive log space for online database reorganization

Tivoli Storage Manager manages table and index reorganization. The Tivoli Storage Manager server requires additional archive log space if an online table must be reorganized.

The Tivoli Storage Manager logs reorganization activity. No information is lost if there is a database error that requires crash recovery.

The space required for an online table reorganization is determined by the following factors:

- The number of rows to be reorganized
- The number of indexes
- The size of the index keys
- The current organization of the table

Typically, every row in a table is moved twice during an online table reorganization. For each index, each table row must update the index key to reflect the new location. After all accesses to the old location are complete, the index key is updated again to remove references to the old location. When the row is moved back, updates to the index key are performed again. This activity is logged to make online table reorganization fully recoverable. There is a minimum of two data log records (each including the row data) and four index log records (each including the key data) for each row (assuming one index).

Clustering indexes are prone to filling up the index pages, causing index splits and merges that must also be logged. A number of the tables implemented by the server have more than one index. A table that has four indexes would require 16 index log records for each row that is moved for the reorganization.

The server monitors characteristics of the database, the active log, and the archive log to determine if a database backup is needed. For example, during an online table reorganization, if the file system for the archive log space begins to fill up, the server triggers a database backup. When a database backup is started, any online table reorganization in progress is paused so that the database backup can operate without contending for resources with the reorganization.

Active log mirror space

The active log can be mirrored so that the mirrored copy can be used if the active log files cannot be read. There can be only one active log mirror.

Creating a log mirror is a suggested option. 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 when deciding whether to create a log mirror.

If the mirror log directory becomes full, the server issues error messages to the activity log and to the `db2diag.log`. Server activity continues.

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. For example, suppose the drive or file system where the archive log directory is located becomes full and the archive failover log directory does not exist or is full. If this situation occurs, the log files that are ready to be moved to the archive log 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 **DSMSERV LOADFORMAT** utility instead of the wizard to configure the server, you specify the **ARCHLOGDIRECTORY** parameter for the archive log directory. In addition, you specify the **ARCHFAILOVERLOGDIRECTORY** parameter for the archive failover log directory. If the archive failover log is not created at initial configuration, you can create it by specifying the `ARCHFAILOVERLOGDIRECTORY` option in the server options file.

Monitoring space utilization for the database and recovery logs

To determine the amount of used and available active log space, you issue the **QUERY LOG** command. To monitor space utilization in the database and recovery logs, you can also check the activity log for messages.

Active log

If the amount of available active log space is too low, the following messages are displayed in the activity log:

ANR4531I: IC_AUTOBACKUP_LOG_USED_SINCE_LAST_BACKUP_TRIGGER
This message is displayed when the active log space exceeds the maximum specified size. The Tivoli Storage Manager server starts a full database backup.

To change the maximum log size, halt the server. Open the `dsmserv.opt` file, and specify a new value for the `ACTIVELOGSIZE` option. When you are finished, restart the server.

ANR0297I: IC_BACKUP_NEEDED_LOG_USED_SINCE_LAST_BACKUP
This message is displayed when the active log space exceeds the maximum specified size. You must back up the database manually.

To change the maximum log size, halt the server. Open the `dsmserv.opt` file, and specify a new value for the `ACTIVELOGSIZE` option. When you are finished, restart the server.

ANR4529I: IC_AUTOBACKUP_LOG_UTILIZATION_TRIGGER
The ratio of used active-log space to available active-log space exceeds the log utilization threshold. If at least one full database backup has occurred, the Tivoli Storage Manager server starts an incremental database backup. Otherwise, the server starts a full database backup.

ANR0295I: IC_BACKUP_NEEDED_LOG_UTILIZATION
The ratio of used active-log space to available active-log space exceeds the log utilization threshold. You must back up the database manually.

Archive log

If the amount of available archive log space is too low, the following message is displayed in the activity log:

ANR0299I: IC_BACKUP_NEEDED_ARCHLOG_USED
The ratio of used archive-log space to available archive-log space exceeds the log utilization threshold. The Tivoli Storage Manager server starts a full automatic database backup.

Database

If the amount of space available for database activities is too low, the following messages are displayed in the activity log:

ANR2992W: IC_LOG_FILE_SYSTEM_UTILIZATION_WARNING_2
The used database space exceeds the threshold for database space utilization. To increase the space for the database, use the **EXTEND DBSPACE** command, the **EXTEND DBSPACE** command, or the DSMSERV FORMAT utility with the **DBDIR** parameter.

ANR1546W: FILESYSTEM_DBPATH_LESS_1GB
The available space in the directory where the server database files are located is less than 1 GB.

When a Tivoli Storage Manager server is created with the DSMSERV FORMAT utility or with the configuration wizard, a server database and recovery log are also created. In addition, files are created to hold database information used by the database manager. The path specified in this message indicates the location of the database information used by the database manager. If space is unavailable in the path, the server can no longer function.

You must add space to the file system or make space available on the file system or disk.

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.

About this task

You can monitor the database and recovery log space whether the server is online or offline.

Procedure

- 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 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
      Package Cache Hit Ratio: 100.0
  Last Database Reorganization: 06/25/2009 01:33:11
       Full Device Class Name: LTO1_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 database 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 Tivoli Storage Manager database by creating directories and adding them to the database.

About this task

The Tivoli Storage Manager server can use all of 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 size of the Tivoli Storage Manager database is 4 TB.

Use the **QUERY DBSPACE** command to display the number of free pages in the table space and the free space that is available to the database. If the number of free pages is low and there is plenty of free space available, the database allocates more space. However, if free space in drives or file systems is low, it might not be possible to expand the database.

You must format the database with Tivoli Storage Manager V6.2 server or later. If you are using a Tivoli Storage Manager V6.1 server or earlier, you must manually upgrade table spaces to the DB2 9.7 version of table spaces. The instructions for manually upgrading the table spaces Tivoli Storage Manager V6.1 server and earlier is different for each version level. Contact IBM Software Support for the instructions for your version of Tivoli Storage Manager.

If you want to increase space for the database, you can create new directories and add them by using the **EXTEND DBSPACE** command. You can also use the **DSMSERV EXTEND DBSPACE** utility when the server is offline. When you issue these commands with the default parameter values, data is redistributed across the new database directories and storage space in the old directories is reclaimed. This action makes the new directories available for use immediately and parallel I/O performance is improved.

To redistribute data to new directories, the storage paths must be able to host the directories and data. Make sure that sufficient disk space is available for the operation. New directories must be empty.

For example, assume that your table space has four directories over two existing storage paths (two directories per path), and you add another storage path to the

database. When data is redistributed for the table space, two new directories of the same size as the existing directories are created on the new storage path. If your table space has a size of approximately 100 GB, then you need 50 GB of free space on the new storage path for redistribution to be successful. After redistribution is complete and space is reclaimed, the total used space is still the same amount of 100 GB, until you begin adding more data.

The following image shows the example table space before and after data is redistributed.

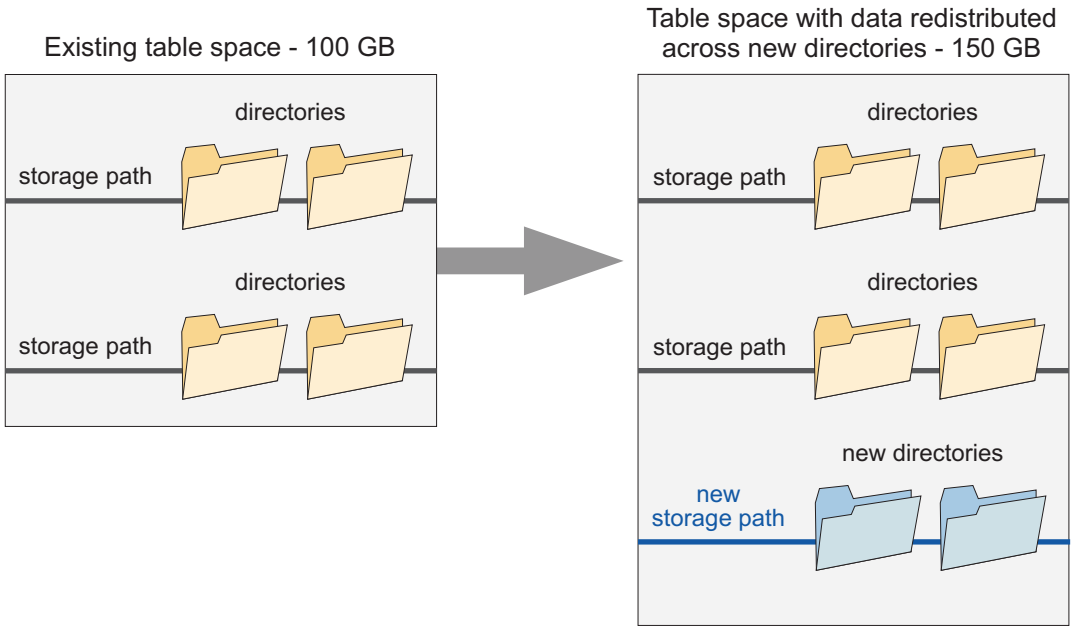


Figure 78. Redistributing data for table spaces

The process of redistributing data and reclaiming space uses considerable resources. Ensure that you plan ahead when you want to add space to the database. For planning requirements, review the **EXTEND DBSPACE** command.

If you do not want to redistribute data at the same time that you add directories, you can set the **RECLAIMSTORAGE** parameter in the **EXTEND DBSPACE** command to `No`. You can perform the tasks to redistribute data and reclaim space after the database size is increased, but the steps must be done manually.

Procedure

To add space to the database, complete the following steps:

1. Create one or more directories for the database on separate drives or file systems.

   Review best practices for creating database directories in the *Optimizing Performance* guide.
2. Issue the **EXTEND DBSPACE** command to add the directory or directories to the database. The directories must be accessible to the user ID of the database manager. By default, data is redistributed across all database directories and space is reclaimed.

Remember: The time that is needed to complete redistribution of data and reclaiming of space is variable, depending on the size of your database. Make sure that you plan adequately.

3. Halt and restart the server to fully use the new directories.

Example

AIX, HP-UX, Linux, and Solaris: For example, to add two directories to the storage space for the database and then redistribute data and reclaim space, issue the following command:

```
extend dbspace /tsmdb005,/tsmdb006
```

For example, to add two directories to the storage space for the database and then redistribute data and reclaim space, issue the following command:

```
extend dbspace /tsmdb005,/tsmdb006
```

To increase the size of the database without redistributing data and reclaiming space, issue the following command:

```
extend dbspace /tsmdb005,/tsmdb006 reclaim=no
```

Manually redistributing data and reclaiming space after you add directories to the database

If you add directories to the Tivoli Storage Manager database with the **EXTEND DBSPACE** command and choose not to redistribute data and reclaim space at the same time, you can complete these tasks later by using DB2 commands.

Before you begin

Restriction: The redistribution process, also known as rebalancing, only works with DB2 version 9.7 or later table spaces, which are created when you format a new Tivoli Storage Manager version 6.2 or later server. If you upgraded or restored your Tivoli Storage Manager server from version 6.1, and you want to rebalance your table spaces, contact IBM Software Support for instructions.

Redistribution uses considerable resources. Review the following guidelines before you start the procedure:

- Run the process when the server is not handling a heavy workload.
- To redistribute data to new directories, the storage paths must be able to host the directories and data. Make sure that sufficient disk space is available for the operation.
- The time that is required to redistribute data and reclaim space might vary. File system layout, the ratio of new paths to existing storage paths, server hardware, and concurrent operations are all factors in determining the time requirement. Start the process with one small and one medium-sized table space and then try a larger table space. Use your results as a reference to estimate the time that is needed to process remaining table spaces.
- Do not interrupt the process. If you try to stop it, for example, by halting the process that is completing the work, you must stop and restart the DB2 server. When the server is restarted, it will go into crash recovery mode, which takes several minutes, after which the process resumes.

Procedure

Complete the following steps to redistribute data and then reclaim space for each table space. For the best performance, rebalance a table space and then reduce the size for that table space. While the size for the first table space is being reduced, you can start rebalancing on the second table space, and so on.

1. Open the DB2 command line processor and issue the following command:

   ```
   db2 connect to tsmdb1
   ```

2. List DB2 table spaces by issuing the following command. To display details about each table space, including the total size of a table space and how many bytes are used in each file system where the table space is located, include `show detail`.

   ```
   db2 list tablespaces show detail
   ```

   You only need to redistribute data on Database Managed Space (DMS) table spaces. The following example output shows where the table space type is identified:

   ```
   Tablespaces for Current Database

    Tablespace ID            = 0
    Name                     = SYSCATSPACE
    Type                     = Database managed space  <---DMS table space.
    Contents                 = All permanent data. Regular table space.
    State                    = 0x0000
    Detailed explanation:
    Normal
   ```

3. Use the list that you obtained in Step 2 to identify each DMS table space. For each DMS table space, issue the following command to start redistribution of data for the first DMS space:

   ```
   db2 alter tablespace tablespace_name rebalance
   ```

4. Monitor the data redistribution progress by issuing the following command:

   ```
   db2list utilities show detail
   ```

   If the rebalance process is running, the command output shows `Type = REBALANCE`, and also indicates how many extents are moved and how many remain to be moved.

   The following example output shows where these details are displayed:

   ```
   ID = 6219
   Type = REBALANCE                    <--- Data is being redistributed.
   Database Name = AX4
   Partition Number = 0
   Description = Tablespace ID: 37
   Start Time = 04/27/2009 21:37:37.932471
   State = Executing
   Invocation Type = User
   Throttling:
   Priority = Unthrottled
   Progress Monitoring:
   Estimated Percentage Complete = 15
   Total Work = 22366 extents          <--- Total extents to be moved.
   Completed Work = 3318 extents       <--- Total extents moved.
   Start Time = 04/27/2009 21:37
   ```

   The value in the `Completed Work` field should increase as the redistribution progresses. The `db2diag` log also records status about the process, including start and complete time and what percentage of the process is complete at a certain time.

5. After the redistribution process is completed, reduce the size for each table space. During and after the operation, table spaces have a much larger total size because directories are added. Issue the following command:

   ```
   db2 alter tablespace tablespace_name reduce max
   ```

Reducing the size of the database

If a significant amount of data has been deleted from the database, consider reducing the database size.

Procedure

1. Create a file that contains a list of directories that represent the new directories. For example, `dbdirs.txt`.
2. Run a full database backup by issuing the following command:

   ```
   backup db devclass=tapeclass type=full
   ```

3. Halt the server.
4. Remove the database instance by issuing the following command:

   ```
   dsmserv removedb TSMDB1
   ```

5. Restore the database specifying the file that contains the directories to be used. For example, you can issue the following command:

   ```
   dsmserv restore db todate=today on=dbdirs.txt
   ```

6. Restart the server.

Scheduling table and index reorganization

Unexpected database growth, unexpected active and archive log space requirements, and reduced server performance can occur over time if tables or the indexes associated with tables are not reorganized. Tables and indexes are reorganized by default by Tivoli Storage Manager. If automatic reorganization is affecting server performance, you can schedule reorganization manually.

Before you begin

Reorganization of table data can be initiated by the Tivoli Storage Manager server or by DB2. If server-initiated reorganization is enabled. The server analyzes selected database tables and indexes based on table activity, and determines when reorganization is required. The database manager runs a reorganization while server operations continue. If reorganization by DB2 is enabled, DB2 controls the reorganization process. Reorganization by DB2 is not recommended.

Server-initiated table reorganization is enabled by default. If you are deduplicating data, consider enabling server-initiated index reorganization. However, table and index reorganization requires locks on the database that might interfere with server operations. Reorganization also requires significant processor, active log, and archive log resources. If reorganization is causing performance problems, you can manually schedule table and index reorganizations.

The best time to start a reorganization is when server activity is low and when access to the database is optimal. Schedule table reorganization for databases on servers that are not running deduplication. Schedule table and index reorganization on servers that are running deduplication.

Important: Ensure that the system on which the Tivoli Storage Manager server is running has sufficient memory and processor resources. To assess how busy the

system is over time, use operating system tools to assess the load on the system. You can also review the db2diag.log file and the server activity log. If the system does not have sufficient resources, reorganization processing might be incomplete, or it might degrade or destabilize the system.

Restrictions to table and index reorganization

If you manually schedule a table or index reorganization, be aware of the restrictions that can affect performance.

Table reorganization

Under certain conditions, table reorganization can pause if reorganization activity exceeds the time specified by the REORGDURATION server option. Table reorganization can also pause to make active log space available. If table reorganization is paused for either reason, reorganization starts again if the interval is not exceeded. Otherwise, reorganization remains paused until the following day at the next scheduled reorganization.

Do not schedule database backups during a table reorganization interval. Database backup takes precedence over table reorganization:

- If a database backup is running, table reorganization cannot start until the database backup completes. After the database backup completes, table reorganization can start if the start time is within the interval specified by the REORGDURATION server option.
- If a running table reorganization is paused, the server checks to determine whether a database backup is required. If a backup is required, it starts automatically.

Index reorganization

If a database backup is running, index reorganization cannot start until the database backup completes. After the database backup completes, index reorganization can start if the start time is within the interval specified by the REORGDURATION server option.

Unlike table reorganization, index reorganization cannot be paused. If an index reorganization is running, the server cannot start a full database backup. If an index is large, reorganization can take a long time to complete and can generate a large amount of data in the active log and the archive log. To reduce the possibility of having to manually start a full database backup, take one or more of the following steps:

- Consider allocating relatively large amounts of archive log space that the server can use during index reorganization.
- To ensure that running index reorganizations complete, do not schedule a full database backup for least an hour after the interval during which an index reorganization can start.
- Ensure that the value of the REORGDURATION server option limits when index reorganizations can start.

Continue to monitor database and log space usage during index reorganization. If the server starts to run out of archive log space, begin a full database backup.

Scheduling table or index reorganization

To schedule table or index reorganization, you specify the earliest time and an interval after that time during which reorganization can start.

Procedure

To schedule table or index reorganization, complete the following steps:

1. Enable table or index reorganization by setting one or both of the following server options to `yes` in the `dsmserv.opt` file.
   - `ALLOWREORGTABLE`
   - `ALLOWREORGINDEX`
2. Specify the time that the server can start table or index reorganization by setting the `REORGBEGINTIME` server option. Specify the time by using a 24-hour format. The best time to start a reorganization is when server activity is low and when access to the database is optimal. Schedule table reorganization for databases on servers that are not running data deduplication. Schedule table and index reorganization on servers that are running data deduplication.
3. Specify the interval during which table or index reorganization must start by setting the `REORGDURATION` server option. Specify the time as a number of hours.

Results

If you set only the `REORGBEGINTIME` option, reorganization is enabled for an entire day. If you do not specify the `REORGBEGINTIME` option, but you specify a value for the `REORGDURATION` option, the reorganization interval starts at 6:00 a.m. and runs for the specified number of hours.

If your system experiences performance problems, modify the values of the `REORGBEGINTIME` option and the `REORGDURATION` option. When you modify the values, you ensure that table or index reorganization is not running during periods of peak server activity.

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.

Procedure

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` 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 plan to use a new active log directory, update the directory name specified in the ACTIVELOGDIRECTORY server option. The new directory must be empty and must be accessible to the user ID of the database manager.
6. Restart the server.

Results

Log files of 512 MB are automatically defined until the size specified in the ACTIVELOGSIZE option is reached. If a log mirror exists, log files are also defined automatically.

Reducing the size of the active log

You can reduce the size of the active log during the startup of the Tivoli Storage Manager server.

Procedure

If you have too much active log space, you can reduce the size of the active log by completing the following steps:

1. Stop the Tivoli Storage Manager server.
2. In the dsmserv.opt file, change the ACTIVELOGSIZE option to the new size of the active log, in megabytes. For example, to reduce the active log by 8 GB, enter the following server option:

   ```
   dsmserv activelogsize 8000
   ```

3. Restart the server.

Results

When you reduce the size of the active log, you must restart the Tivoli Storage Manager server twice. The first restart changes the DB2 parameters. The second restart removes the log files that are no longer required on the disk.

Moving the database and recovery log on a server

You can move the database and logs to different locations on the server.

About this task

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 670
- "Moving only the database" on page 670
- "Moving only the active log, archive log, or archive failover log" on page 671

For information about moving a Tivoli Storage Manager server to another machine, see "Moving the Tivoli Storage Manager server to another system" on page 599

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.

Procedure

1. Back up the database by issuing the following command:

   ```
   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 is used if the database must be restored. Enter each location on a separate line. For example, these are the contents of the `dbdirs.txt` file:

   ```
   /tsmdb005
   /tsmdb006
   /tsmdb007
   /tsmdb008
   ```

5. Remove the database instance by issuing the following command:

   ```
   dsmserv removedb TSMDB1
   ```

6. Issue the **DSMSERV RESTORE DB** utility to move the database and create the new active log. For example:

   ```
   dsmserv restore db todate=today on=dbdirs.txt
      activelogdir=/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.

Procedure

To move the database from one location on the server to another location, follow this procedure:

1. Back up the database by issuing the following command:

   ```
   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 by issuing the following command:

```
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, archive log, or archive failover log

You can move only the active log, archive log, or archive failover log from one directory to another. You can move the logs as part of performance tuning, replacing a hard disk drive, or changing to a file system with more space.

Procedure

1. Stop the server.
2. Create a directory by using the following commands for the log that you want to move.

Table 58. Commands for new directories

Log name	Command
Active log	`mkdir /activelog2`
Archive log	`mkdir /archivelog2`
Archive failover log	`mkdir /tsmserver1/archfaillog2`

3. Update the `dsmserv.opt` option file for the new directory by using one of the following commands.

Table 59. Server option example to update the `dsmserv.opt` option file

Log name	Server option example
Active log	`activelogdirectory /activelog2`
Archive log	`archlogdirectory /archivelog2`
Archive failover log	`archfailoverlogdirectory /tsmserver1/archfaillog2`

4. Restart the server.
5. Move or copy the logs from the old directory to the new directory.

Table 60. Commands to move or copy the logs

Log name	Command
Active log	No action is necessary. The active logs are automatically moved by the database manager from the old directory to the new directory.
Archive log	Preserve the subdirectory structure. Use and verify a recursive copy directory command. For example: `cp -r /archivelog/* /archivelog2`

Table 60. Commands to move or copy the logs (continued)

Log name	Command
Archive failover log	Preserve the subdirectory structure. Use and verify a recursive copy directory command. For example: `cp -r /tsmserver1/archfaillog/* /tsmserver1/archfaillog2`

6. Optional: Remove the old directory by using the **rmdir** command.

Specifying alternative locations for database log files

A database backup includes a database image and associated recovery-log files. During a database restore operation, the recovery-log files are temporarily stored in a subdirectory called `RstDbLog`. Log files that are created during day-to-day operations and that cannot be stored in the active log directory or in the archive log directory can be temporarily stored in a subdirectory called `failarch`.

Procedure

To specify alternative locations for the database log files, complete the following steps:

1. To specify the location of subdirectories `RstDbLog` and `failarch`, use the **ARCHFAILOVERLOGDIRECTORY** server option. The Tivoli Storage Manager server creates the `RstDbLog` and `failarch` subdirectories in the directory that is specified by the server option.

   Restriction: If you do not specify the location of the subdirectories, the Tivoli Storage Manager server automatically creates the two subdirectories under the archive log directory

   If the archive log directory becomes full, it can limit the amount of space that is available for archived log files. If you must use the archive log directory, you can increase its size to accommodate both the `RstDbLog` and `failarch` directories.
2. Use a file system that is different from the file system that is specified by the **ACTIVELOGDIRECTORY** and **ARCHLOGDIRECTORY** parameters.

   Tip: If you do not set the **ARCHFAILOVERLOGDIRECTORY** option, the Tivoli Storage Manager server creates the `RstDbLog` and `failarch` subdirectories automatically in the directory that is specified for the **ARCHLOGDIRECTORY** parameter on the **DSMSERV FORMAT** or **DSMSERV LOADFORMAT** command. You must specify the **ARCHLOGDIRECTORY** parameter for these commands.
3. For a database restore operation, you can specify the location of the `RstDbLog` subdirectory, but not the `failarch` subdirectory, by using the **RECOVERYDIR** parameter on the **DSMSERV RESTORE DB** command. Consider allocating a relatively large amount of temporary disk space for the restore operation. Because database restore operations occur relatively infrequently, the `RstDbLog` subdirectory can contain many logs from backup volumes that are stored in preparation for pending roll-forward-restore processing.

Specifying an alternative location with the ARCHFAILOVERLOGDIRECTORY server option or parameter

If you specify the ARCHFAILOVERLOGDIRECTORY server option or the **ARCHFAILOVERLOGDIRECTORY** parameter in the **DSMSERV FORMAT** or **DSMSERV LOADFORMAT** command, the server creates the RstDbLog and failarch subdirectories in the archive failover-log directory.

About this task

The server also updates the DB2 parameter **OVERFLOWLOGPATH** that points to the RstDbLog subdirectory and the DB2 parameter **FAILARCHPATH**, that points to the failarch subdirectory. For details about these parameters, see the DB2 information center at http://pic.dhe.ibm.com/infocenter/db2luw/v10r5.

For example, suppose that you specify archlogfailover as the value of the **ARCHFAILOVERLOGDIRECTORY** parameter on the **DSMSERV FORMAT** command:

```
dsmserv format
   dbdir=/tsmdb001
   activelogdirectory=/home/tsminst1/inst1/activelog
   archlogdirectory=/home/tsminst1/inst1/archlog
   archfailoverlogdirectory=/home/tsminst1/inst1/archlogfailover
```

The server creates the subdirectories RstDbLog and failarch in the parent directory archlogfailover. The server also updates the following DB2 parameters:

```
OVERFLOWLOGPATH=/home/tsminst1/inst1/archlogfailover/RstDbLog
FAILARCHPATH=/home/tsminst1/inst1/archlogfailover/failarch
```

The server also updates the value of the ARCHFAILOVERLOGDIRECTORY option in the server options file, dsmserv.opt:

```
ARCHFAILOVERLOGDIRECTORY /home/tsminst1/inst1/archlogfailover
```

Specifying an alternative location with the ARCHLOGDIRECTORY server option or parameter

If you do not set the **ARCHFAILOVERLOGDIRECTORY** parameter or server option, the server automatically creates the subdirectories RstDbLog and failarch in the directory that is specified for the **ARCHLOGDIRECTORY** parameter. In addition, the server updates the DB2 parameters **OVERFLOWLOGPATH** and **FAILARCHPATH** to point to those subdirectories.

About this task

For details about these parameters, see the DB2 Information Center at http://pic.dhe.ibm.com/infocenter/db2luw/v10r5.

For example, suppose that you specify a value of archlog for the **ARCHLOGDIRECTORY** parameter in a **DSMSERV FORMAT** command. You do not specify the **ARCHFAILOVERLOGDIRECTORY** parameter:

```
dsmserv format
   dbdir=/tsmdb001
   activelogdirectory=/home/tsminst1/inst1/activelog
   archlogdirectory=/home/tsminst1/inst1/archlog
```

The Tivoli Storage Manager server creates the subdirectories RstDbLog and failarch under the archlog parent directory. The server also updates the following DB2 parameters:

```
OVERFLOWLOGPATH=/home/tsminst1/inst1/archlog/RstDbLog
FAILARCHPATH=/home/tsminst1/inst1/archlog/failarch
```

The server also updates the value of the ARCHLOGDIRECTORY option in the server options file, dsmserv.opt:

```
ARCHLOGDIRECTORY /home/tsminst1/inst1/archlog
```

Specifying the location of RstDbLog using the RECOVERYDIR parameter

For a database restore operation, you can specify a temporary location for the RstDbLog subdirectory by using the **RECOVERYDIR** parameter on the **DSMSERV RESTORE DB** command. The Tivoli Storage Manager server creates the RstDbLog subdirectory in the directory that is specified by that parameter. The server does not create the failarch subdirectory.

About this task

The server also updates the DB2 parameter, **OVERFLOWLOGPATH**, that points to RstDbLog. For details about this parameter, see the DB2 Information Center at http://pic.dhe.ibm.com/infocenter/db2luw/v10r5.

Restriction: The location specified by the **RECOVERYDIR** parameter takes precedence over the location specified by the ARCHFAILOVERLOGDIRECTORY or ARCHLOGDIRECTORY server option or parameter.

For example, for a point-in-time database restore, you can issue the following command:

```
dsmserv restore db
   todate=5/12/2011
   totime=14:45
   recoverydir=/home/tsminst1/inst1/recovery
```

The server creates the RstDbLog subdirectory in the parent recovery directory. The server also updates the **OVERFLOWLOGPATH** parameter:

```
OVERFLOWLOGPATH=/home/tsminst1/inst1/recovery/RstDbLog
```

After the database is restored, the RstDbLog subdirectory reverts to its location as specified by the server option ARCHFAILOVERLOGDIRECTORY or ARCHLOGDIRECTORY in the server options file, dsmserv.opt.

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.

Procedure

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:

   ```
   archfailoverlogdirectory /archfailoverlog
   mirrorlogdirectory /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. 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 for the TXNGROUPMAX option can have the following impact:

- Affect the performance of client backup, archive, restore, and retrieve operations.
- 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 661 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 20. 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*10*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.

- 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*10*5=100 000
  ```

  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.

Remember: 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 61. 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. 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.

Refer to the **REGISTER NODE** command and the server options in the *Administrator's Reference*.

Chapter 20. 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.

About this task

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 678

Tasks:
"Setting up communications among servers" on page 682
"Setting up communications for enterprise configuration and enterprise event logging" on page 682
"Setting up communications for command routing with multiple source servers" on page 687
"Completing tasks on multiple servers" on page 715
"Using virtual volumes to store data on another server" on page 722

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.

© Copyright IBM Corp. 1993, 2013

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 579.

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 79 on page 679 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 691.

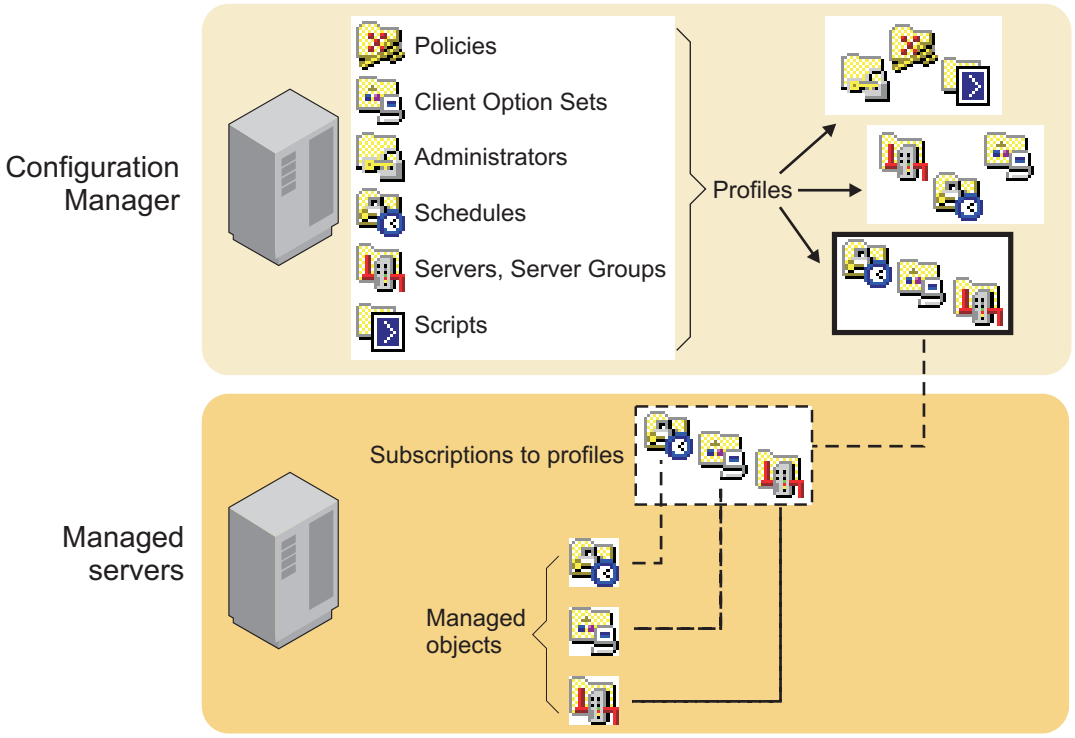


Figure 79. Enterprise configuration

Command routing

Use the 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 718 and "Routing commands" on page 716.

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 811. For information about communications setup, see "Setting up communications for enterprise configuration and enterprise event logging" on page 682.
- Allowing designated administrators to log in to any of the servers in the network with a single user ID and password.

  See "Managing multiple servers through a web-based interface" on page 715.
- 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 716. For information about communications setup, see "Setting up communications for enterprise configuration and enterprise event logging" on page 682.

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.

The data can also be a recovery plan file created by using disaster recovery manager (DRM). 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.

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 579.

For information about using virtual volumes with DRM, see Chapter 30, "Disaster recovery manager," on page 979.

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.

About this task

Communication setup for server-to-server virtual volumes is described in "Setting up source and target servers for virtual volumes" on page 724. See "Setting the server name" on page 604 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.

About this task

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:

Procedure

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).

Results

Figure 80 on page 684 shows the servers and the commands issued on each:

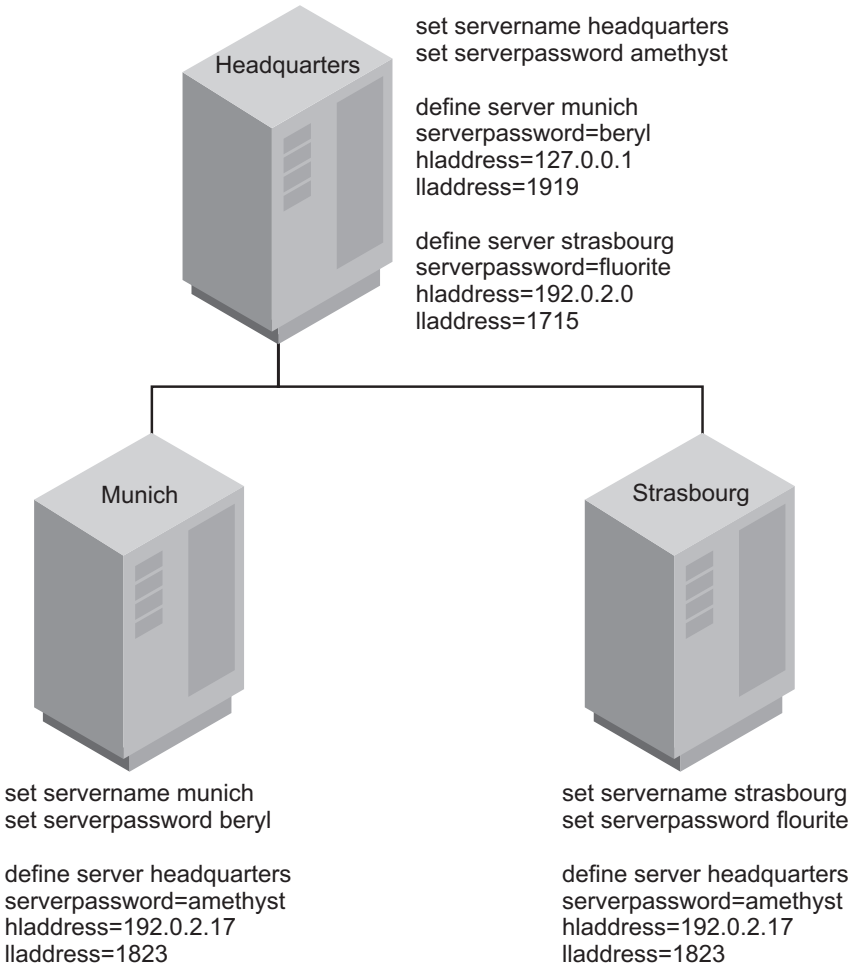


Figure 80. Communication configuration with separate server definitions

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 81 on page 685 shows the servers and the commands issued on each:

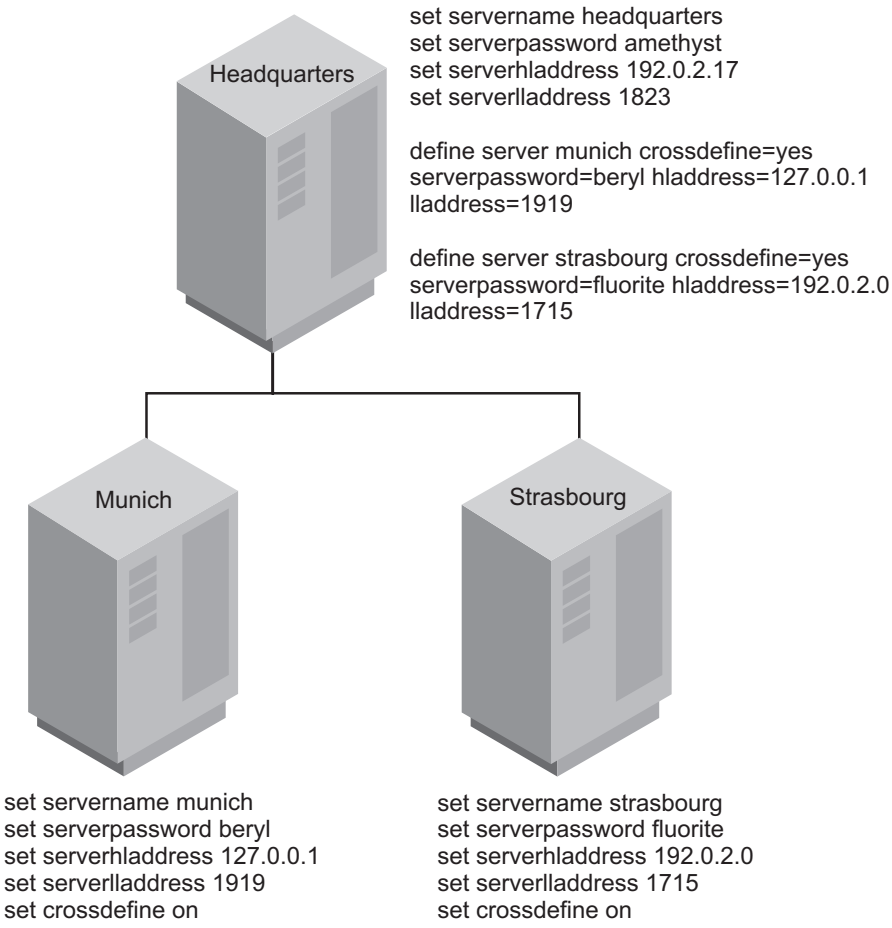


Figure 81. Communication configuration with cross definition

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.

About this task

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.

About this task

The example that is provided shows you how you can set up communications for administrator HQ on the server HEADQUARTERS who routes commands to the servers MUNICH and STRASBOURG. Administrator HQ has the password SECRET and has system privilege class.

Procedure

The procedure for setting up communications for command routing with one source server is shown in the following steps.

- 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=beryl hladdress=9.115.2.223 lladdress=1919
define server strasbourg serverpassword=fluorite hladdress=9.115.2.178
 lladdress=1715
```

  Note: Command routing uses the ID and password of the Administrator. It does not use the password or server password that is set in the server definition.
- On MUNICH and STRASBOURG, register administrator HQ with the specific 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.

About this task

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 you are setting up communications for command routing, you can define each server to each of the other servers.

Procedure

To create separate definitions, complete the following steps:

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.

Results

Figure 82 on page 688 shows the servers and the commands that are issued on each.

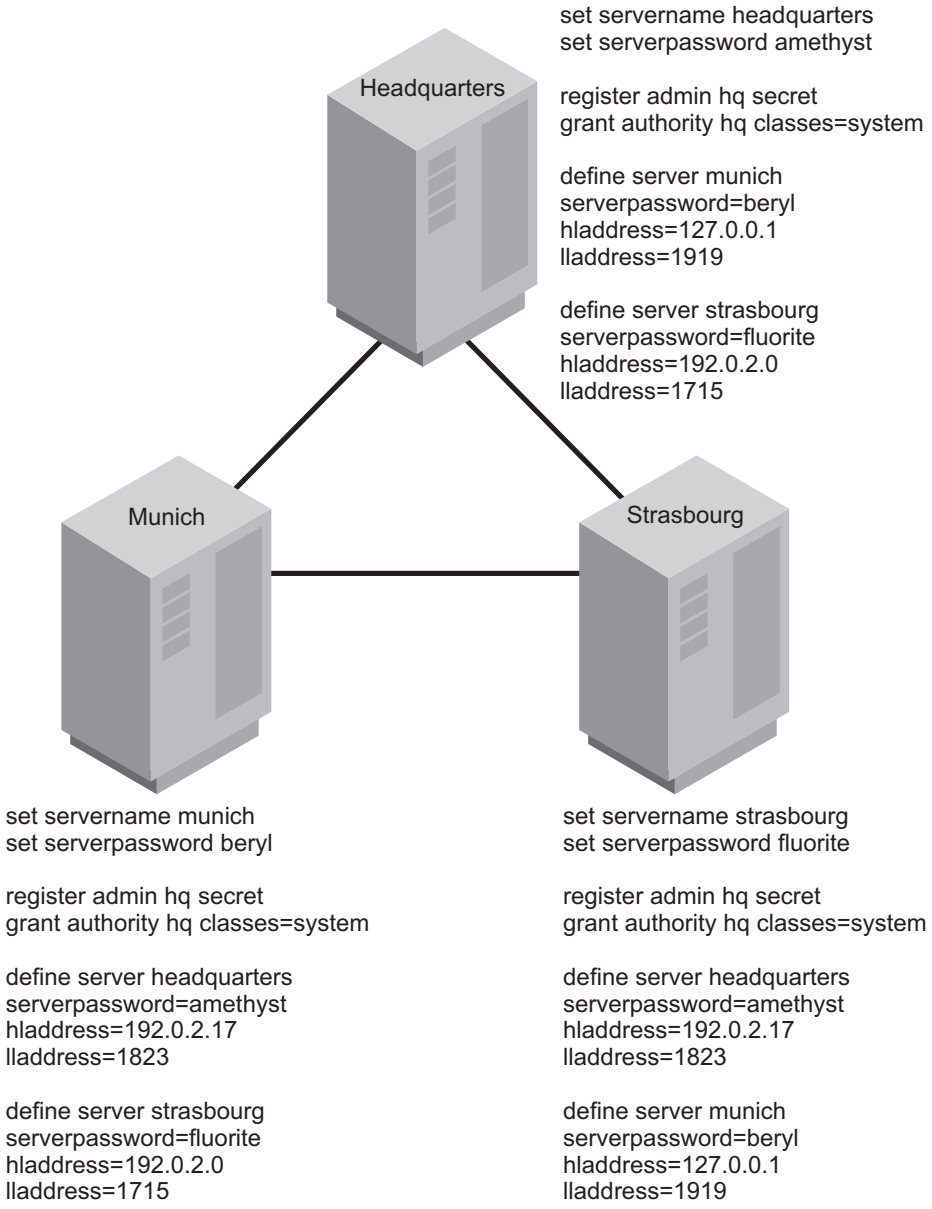


Figure 82. Communication configuration with separate server definitions

Creating cross definitions:

When you are setting up communications for command routing, you can cross-define the other servers.

Procedure

To create cross definitions, complete the following steps:

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 is to be completed.
3. On MUNICH, define STRASBOURG, specifying that cross define is to be completed.

Results

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 83 on page 690 shows the servers and the commands that are issued on each.

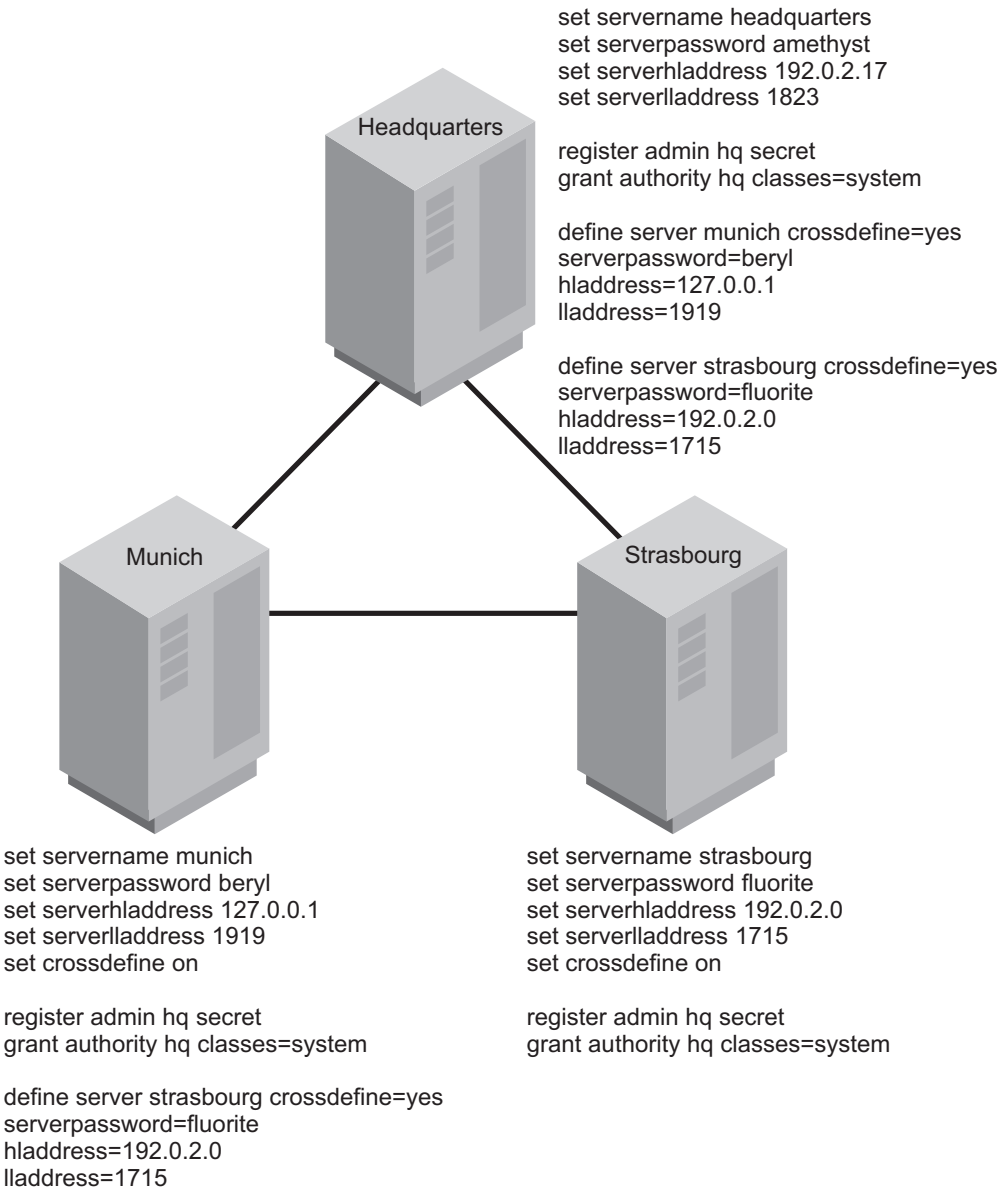


Figure 83. Communication configuration with cross definitions

Updating and deleting servers

You can update and delete server definitions for server-to-server virtual volumes, enterprise configuration, and enterprise event logging.

About this task

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 718 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.

About this task

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.

If you use an LDAP directory server to authenticate passwords, any target servers must be configured for LDAP passwords. Data that is replicated from a node that authenticates with an LDAP directory server is inaccessible if the target server is not properly configured. If your target server is not configured, replicated data from an LDAP node can still go there. But the target server must be configured to use LDAP in order for you to access the data.

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 697. After you set up server communication as described in "Setting up communications for enterprise configuration and enterprise event logging" on page 682, 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.

About this task

Figure 84 shows the hierarchy that you want to set up.

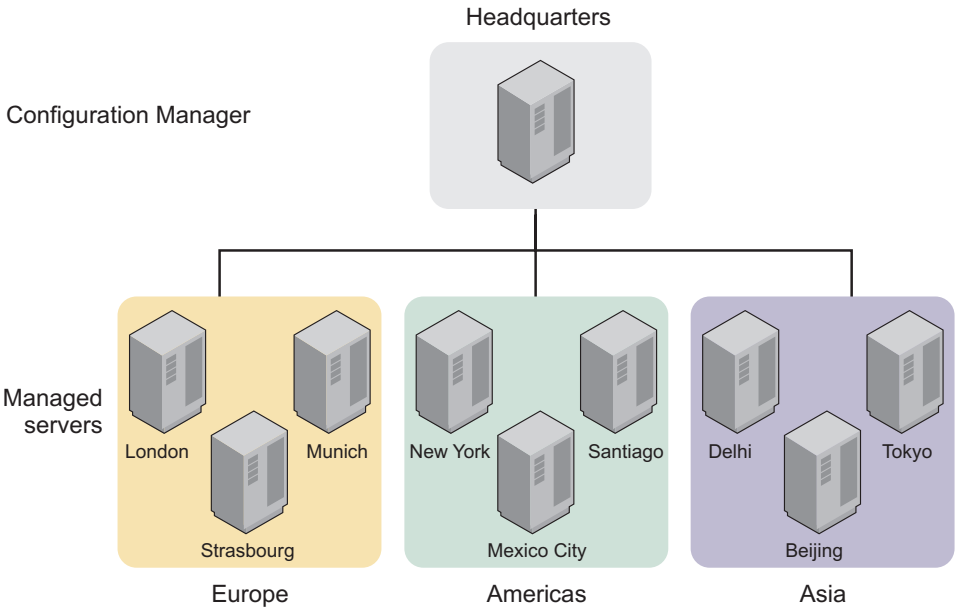


Figure 84. A scenario for implementing enterprise configuration

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.

About this task

Figure 85 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.

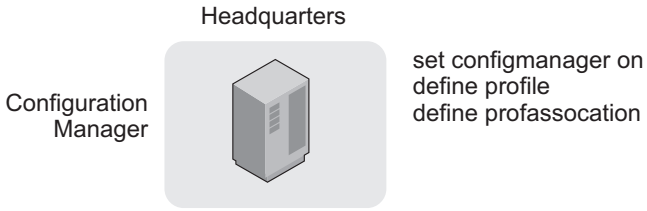


Figure 85. Setting up a configuration manager

Procedure

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 682 for details. For more information, see “Creating the default profile on a configuration manager” on page 696. See “Defining a server group and members of a server group” on page 719 for details. For details, see “Creating and changing configuration profiles” on page 697.

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.

About this task

Figure 86 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.

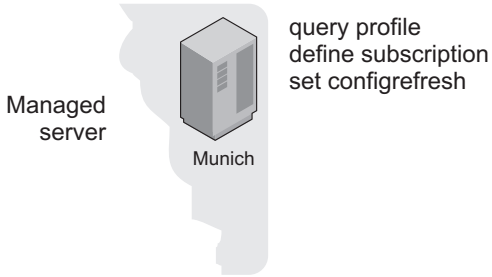


Figure 86. Setting up a managed server

A server becomes a managed server when that server first subscribes to a profile on a configuration manager.

Procedure

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 611.

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 697
   - "Defining storage pools" on page 241
   - "Getting information about profiles" on page 704
   - "Refreshing configuration information" on page 711
   - "Renaming storage pools" on page 405
   - "Subscribing to a profile" on page 707

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.

About this task

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.

If the configuration manager is on a V6.3.3 or later server, password authentication methods can cause conflicts between the configuration manager and the managed servers. Administrator IDs with LDAP-authenticated passwords on a V6.3.3 or later server cannot be accessed on a managed server that is on a pre-V6.3.3 server.

Creating and changing configuration profiles

You create configuration profiles on a configuration manager, which distributes the information that is associated with the profiles to any managed server that subscribes to those profiles.

Procedure

Create a configuration profile by completing the following steps:

1. Defining the profile.
2. Associating the configuration information with the profile.

Results

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 that is 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.

About this task

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.

About this task

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 701
- "Configuring information for policy domains" on page 699
- "Configuring information for servers and server groups" on page 700
- "Configuration information for Tivoli Storage Manager administrators"
- "Tivoli Storage Manager server scripts" on page 616
- "Managing client option sets" on page 461

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.

About this task

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:

Procedure

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.

About this task

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 62 summarizes what happens when servers or server groups being distributed have the same names as servers or server groups on the managed server.

Table 62. 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.

About this task

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.

About this task

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 702 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.

About this task

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.

About this task

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 711 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.

About this task

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:

Procedure

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
   ```

Results

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 710.

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.

About this task

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 710 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.

About this task

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       Profile name        Locked?
manager
---------------     ---------------     -------
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.

About this task

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.

About this task

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:

Procedure

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
   ```

Results

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.

About this task

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                                      date/time
---------------     ---------------     -------------------
HEADQUARTERS        ADMINISTRATORS      06/04/2002 17:51:49
HEADQUARTERS        ADMINS_1            06/04/2002 17:51:49
HEADQUARTERS        ADMINS_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.

About this task

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.

About this task

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.

About this task

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 697 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.

About this task

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:

Procedure

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
```

Results

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.

About this task

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

About this task

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:

Procedure

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.

Results

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.

Procedure

Complete the following steps to switch a managed server.

1. Query profiles on the server that is 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 are deleted by querying subscriptions.
3. Change server communications as needed. Define the server that is 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).

About this task

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.

Procedure

To rename a managed server, complete 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.

Results

See “Setting the server name” on page 604 for more information before you use the**SET SERVERNAME** command.

Completing tasks on multiple servers

To easily complete tasks with multiple servers, Tivoli Storage Manager provides you with the Operations Center, command routing, and server group definitions that you can use to simplify command routing.

Managing multiple servers through a web-based interface

Tivoli Storage Manager provides a web-based interface for managing and monitoring multiple Tivoli Storage Manager servers.

About this task

Through the Operations Center, you can manage multiple Tivoli Storage Manager servers, issue commands to those servers, and access web clients.

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.

About this task

If you have set up your servers as described in "Setting up communications for command routing" on page 686, 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 718)

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.

About this task

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.

About this task

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.

About this task

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" 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.

About this task

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.

Procedure

To use server groups, you must complete the following tasks:

1. Define the server groups.
2. Add the servers as members of the appropriate group.

Results

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.

About this task

Task	Required Privilege Class
Define a server group	System
Define a server group member	System

Procedure

To route commands to a server group, you must complete 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.

Results

The following example shows how to create a server group that is called 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 686)

Managing server groups

You can query, copy, rename, update, and delete server groups as necessary.

About this task

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 by issuing the **QUERY SERVERGROUP** command.

Procedure

To query the WEST_COMPLEX server group, issue the following command:

```
query servergroup west_complex
```

Results

The following sample output shows the results of the **QUERY SERVERGROUP** command.

```
Server Group      Members        Description          Managing profile
--------------------------------------------------------------------------
WEST_COMPLEX      BLD12, BLD13
```

Copying a server group:

You can copy a server group by issuing the **COPY SERVERGROUP** command.

Procedure

To copy the entire server group contents of WEST_COMPLEX to the NEWWEST server group, issue the following command:

```
copy servergroup west_complex newwest
```

Results

This command creates the new group. If the new group exists, the command fails.

Renaming a server group:

You can rename a server group by issuing the **RENAME SERVERGROUP** command.

Procedure

For example, to rename an existing server group NORTH_COMPLEX to NORTH, issue the following command:

```
rename servergroup north_complex north
```

Updating a server group description:

You can update a server group by issuing the **UPDATE SERVERGROUP** command.

Procedure

For example, to update the NORTH server group to modify its description, issue the following command:

```
update servergroup north description="Northern marketing region"
```

Deleting a server group:

You can delete a server group by issuing the **DELETE SERVERGROUP** command.

Procedure

To delete the WEST_COMPLEX server group from the Tivoli Storage Manager server, issue the following command:

```
delete servergroup west_complex
```

Results

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.

About this task

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 by issuing the **MOVE GRPMEMBER** command.

Procedure

To move group member TRADE5 from the NEWWEST group to the NORTH_COMPLEX group, issue the following command:

```
move grpmember trade5 newwest north_complex
```

Deleting a group member from a group:

You can delete group members from a group by issuing the **DELETE GROUPMEMBER** command.

Procedure

To delete group member BLD12 from the NEWWEST server group, issue the following command:

```
delete grpmember newwest bld12
```

Results

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 by issuing the **PING SERVER** command.

Procedure

For example, to ping the server GEO2, issue the following command:

```
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 that is being pinged, the ping fails even if the server is 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.

About this task

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
- DRM **PREPARE** command

The data is stored as *virtual volumes*, which look like sequential media volumes on the source server, but are stored as archive files on a target server. The following list includes the virtual volumes that can store data.

- Database backups
- Storage pool backups
- Data that is backed up, archived, or space managed from client nodes
- Client data that is migrated from storage pools on the source server
- Any data that can be moved by EXPORT and IMPORT commands
- DRM plan files

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 are 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 following conditions 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. However, after the data movement or copy operation, the amount of data that is reported as moved or copied is the amount of deduplicated data. For example, if a storage pool contains 20 GB of deduplicated data that represents 50 GB of total file data. If the data is moved or copied, the server reports that 20 GB was moved or copied, even though 50 GB of data was sent.
- 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 databases are backed up in that manner, 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.
- You cannot use a Centera storage pool as the target for virtual volumes.
- 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.
- To enable data validation between a source and target server, issuing both the **DEFINE SERVER** and **REGISTER NODE** commands. For more information, see "Validating a node's data" on page 529 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), you can reset the code by issuing the **UPDATE SERVER** command with the **FORCESYNC** parameter set to YES.

For details, see "Reconciling virtual volumes and archive files" on page 729.

Related concepts:

"Performance limitations for virtual volume operations" on page 725

Related tasks:

"Setting up source and target servers for virtual volumes" on page 724

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.

About this task

In the following example (illustrated in Figure 87 on page 725), 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.

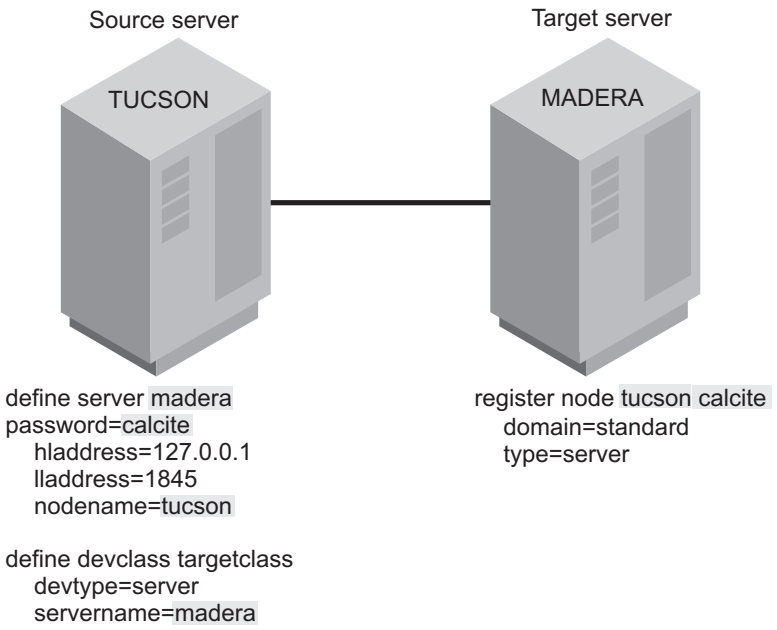


Figure 87. Communication configuration for virtual volumes

Related tasks:

"Changing policy" on page 472

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.

About this task

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
- DRM prepare

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.

About this task

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 DRMDBBACKUPEXPIREDAYS command

See "Moving copy storage pool and active-data pool volumes on-site" on page 998 for more information.

Backing up a storage pool

You can back up a storage pool of a source server to a target server.

About this task

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.

About this task

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 363.

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.

About this task

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.

About this task

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.

About this task

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.

About this task

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 63.

Table 63. 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 21. 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 734
"Preparing to export to another server for immediate import" on page 738
"Monitoring the server-to-server export process" on page 740
Tasks for Exporting to Sequential Media:
"Exporting and importing data using sequential media volumes" on page 743
"Exporting tasks" on page 745
"Importing data from sequential media volumes" on page 748

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:

© Copyright IBM Corp. 1993, 2013

- 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 that are 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. The data is not protected by retention when it is imported to another server.
- You cannot export nodes of type NAS (network-attached storage). Export processing excludes these nodes.
- Exporting data to a Centera device class or importing data from a Centera device class is not supported. However, files that are stored in Centera storage pools can be exported and files that must be imported can be stored on a Centera storage device.
- The **EXPORT NODE** and **EXPORT SERVER** commands do not export data from a shred pool unless you explicitly 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 does not issue a warning if the export operation includes data from shred pools.

Related concepts:

"Securing sensitive client data" on page 533

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 533

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 734

Related tasks:

"Preventing client nodes from accessing the server" on page 734

Related reference:

"Options to consider before exporting" on page 735

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 463

"Locking and unlocking administrator IDs from the server" on page 842

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.

About this task

To prevent users from accessing the server during export operations, cancel existing client sessions.

Procedure

After you cancel the client sessions, you can complete 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.

What to do next

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 MERGEFILESPACES=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 756

Related tasks:

"Querying the activity log for export or import information" on page 762

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 750

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:

"Selection for active file versions" on page 239

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 Suspendible
: 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: "Identifying and exporting eligible files" or phase 3: "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 737: "Identifying and exporting eligible files" or phase 3 on page 737: "File list complete. Exporting eligible files."

Suspended
The operation is not currently running. It may still be in either phase 2 on page 737 or phase 3 on page 737.

An export operation that fails while it is in phase 1 on page 737 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 MERGEFILESPACES=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 MERGEFILESPACES 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.

About this task

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:

Procedure

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 20, "Managing a network of Tivoli Storage Manager servers," on page 677

"Setting up communications among servers" on page 682

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.

About this task

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:

Procedure

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.
You can also view the results on the following applications:

- Server console

Related tasks:

"Requesting information about an export or import process" on page 760

"Canceling server processes" on page 601

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.

About this task

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.

Procedure

To direct messages to an output file named `IMPSERV.OUT`, issue the following command:

```
> dsmadmc -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.

About this task

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.

Procedure

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.

About this task

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

Procedure

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
```

Results

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.

About this task

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

Procedure

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.

Procedure

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.

Procedure

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 760

"Canceling server processes" on page 601

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 to store the exported data.

Procedure

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 that is available to both servers.

   Tip:

   a. If the mount limit for the device class is exceeded 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 process.

   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 that you must label.

   To estimate the number of removable media volumes that you must label, divide the number of bytes to be moved by the estimated capacity of a volume in the storage pool.

   You can estimate the following forms of removable media volumes:

   • The number of tapes that are required to store the exported data

   For example, you have LTO Generation 1 tapes with an estimated capacity of 100 GB. If the preview shows that you need to transfer 150 GB of data, then label at least two tape volumes before you export the data.

3. Use scratch media. You can use scratch media to ensure that you have sufficient space to store all the exported data. If you use scratch media, record the label names and the order in which they were mounted.

   Or, you can use the **USEDVOLUMELIST** parameter on the **EXPORT** command to create a file that contains a list of volumes that were 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 the volumes 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.

   When you export data, record the date and time for each labeled volume. Store this information in a safe location because you need the information when you import the data. Alternatively, if you used the **USEDVOLUMELIST** parameter on the export command, save the generated output file. This file can be used on the import command volumes parameter.

   For more information about exporting data, see **EXPORT SERVER** in the *Administrative's Reference*.

Related tasks:

"Using virtual volumes to store data on another server" on page 722

"Defining device classes" on page 174

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 744

Exporting administrator information

When you issue the **EXPORT ADMIN** command, the server exports administrator definitions.

About this task

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.

Procedure

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.

About this task

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

Procedure

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:

"Selection for active file versions" on page 239

Exporting policy information

You must issue the **EXPORT POLICY** command to export policy information.

About this task

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.

Procedure

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.

About this task

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.

Procedure

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:

"Selection for active file versions" on page 239

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.

Procedure

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.

Results

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 MERGEFILESPACES=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 756

Related tasks:

"Querying the activity log for export or import information" on page 762

Determining whether to replace existing definitions:

By using the **REPLACEDEFS** 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 REPLACEDEFS=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.

About this task

When you set PREVIEW=YES, tape operators must mount export tape volumes so that the target server can calculate the statistics for the preview.

Procedure

Issue the following design to preview information for the **IMPORT SERVER** command:

```
import server devclass=tapeclass preview=yes
volumenames=dsm001,dsm002,dsm003,dsm004
```

Figure 88 on page 752 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 88. 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 88 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 755

"Using preview before exporting or importing data" on page 743

Related reference:

"Monitoring export and import processes" on page 760

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.

About this task

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:

Procedure

1. Read the following topics:
   - "Determining whether to replace existing definitions" on page 750
   - "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 754

"Importing server control information" on page 755

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.

About this task

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.

Procedure

To direct messages to an output file named `IMPSERV.OUT`, issue the following command:

```
> dsmadmc -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.

About this task

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.

Procedure

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 previously imported definition information, you can use the reports that are generated by the import process to help you tailor storage for the target server.

Procedure

To tailor server storage definitions on the target server, complete the following steps:

1. Identify any storage destinations that are 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 that are generated during validation include whether any management classes or copy groups refer to storage pools that do not exist on the target server. If you directed console messages to an output file, you have a copy of these messages in a file.
   - Query management class and copy group definitions to compare the storage destinations that are 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, complete 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. complete 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 needed
c. Activate the new policy set

Results

Depending on the amount of client file data that you expect to import, you might want to examine the storage hierarchy to ensure that sufficient storage space is available. Storage pools that are specified as storage destinations by management classes and copy groups might fill up with data. For example, you might need to define extra storage pools to which data can migrate from the initial storage destinations.

Related tasks:

"Directing import messages to an output file" on page 754

"Defining storage pools" on page 241

Related reference:

"Defining and updating a policy set" on page 494

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 ACTIVEDATA** 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.

About this task

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.

Procedure

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 64 on page 759 for the commands that you can use to import a subset of exported information to a target server.

Table 64. 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 756

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.

About this task

You can query an export or import process by specifying the process ID number.

Procedure

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 781

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 89 on page 761 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 89. 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.

Procedure

1. To start an administrative session in console mode, issue the following command:

   ```
   > dsmadmc -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:

   ```
   > dsmadmc -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.

About this task

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.

Procedure

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
```

Results

Figure 90 on page 763 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 90. 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 722

"Exporting and importing data using sequential media volumes" on page 743

Part 5. Monitoring operations

Monitor client operations, server operations, and maintenance processes to ensure that applications and systems are protected as expected.

© Copyright IBM Corp. 1993, 2013

Chapter 22. Daily monitoring tasks

As a Tivoli Storage Manager administrator, monitor operations daily to ensure that the Tivoli Storage Manager system is functioning properly. Daily monitoring tasks focus on examining server processes, server database, storage pools, and scheduled operations.

About this task

You can complete the monitoring tasks by using the command-line interface (CLI) or the Operations Center, which also includes a CLI.

The following list describes some of the items that are important to monitor daily. Instructions for monitoring these items, and other monitoring tasks can be found in the topics in this section. Not all of these tasks apply to all environments.

- Verify that the database file system has enough space.
- Examine the database percent utilization, available free space, and free-pages.
- Verify that there is enough disk space in the file systems that contain these log files.
  - Active log
  - Archive log
  - Mirror log
  - Archive failover log
- Verify that the instance directory file system has enough space.
- Verify that the database backups completed successfully, and that they are running frequently enough.
- Check the database and recovery log statistics.
- Verify that you have current backup files for device configuration and volume history information. You can find the file names for the backups by looking in the `dsmserv.opt` file for the **DEVCONFIG** and **VOLUMEHISTORY** options. Ensure that file systems where the files are stored have sufficient space.
- Search the summary table for failed processes.
- Search the activity log for error messages.
- For storage pools that have deduplication enabled, ensure that processes are completing successfully.
- Check the status of your storage pools to ensure that there is enough space available.
- Check for any failed storage pool migrations.
- Check the status of sequential access storage pools.
- Check how many scratch volumes are available.
- Determine if there are any tape drives offline, or their paths that are offline.
- Determine if there are any libraries offline, or their paths that are offline.
- Verify that all of the tapes have the appropriate write-access.
- Verify the status and settings for disaster recovery manager (DRM).
- Check for failed or missed schedules.
- Check the summary table for scheduled client operations such as backup, restore, archive, and retrieve.

© Copyright IBM Corp. 1993, 2013

- Check the summary table for scheduled server operations such as migration, reclamation, and expiration.

Monitoring operations using the command line

You can use the command-line interface to monitor your server processes, database, storage pools, and scheduled operations.

Monitoring your server processes daily

Monitor server processes daily to ensure that your server is functioning properly, and that no problems exist.

About this task

For detailed information about the commands mentioned here, see the *Administrator's Reference*.

The examples used here are based on a 24-hour period, but your values can differ depending on the time frame you specify.

The following steps describe the commands that you can use to monitor server processes:

Procedure

1. Search the summary table for any server processes that failed within the previous 24-hour period:

   ```
   select activity as process, number as processnum from summary where
    activity in ('EXPIRATION','RECLAMATION','MIGRATION','STGPOOL BACKUP',
    'FULL_DBBACKUP','INCR_DBBACKUP','REPLICATION') and successful='NO'
    and end_time> (current_timestamp - interval '24' hours)
   ```

   This example output indicates that backup storage pool process number 7 failed:

   ```
      PROCESS: STGPOOL BACKUP
   PROCESSNUM: 7
   ```

2. Search the activity log for the messages associated with the failed process number that was indicated in the output of the command in Step 1.

   ```
   select message from actlog where process=7 and date_time>(current_timestamp
    - interval '24' hours) and severity in ('W','E','S')
   ```

   Example output:

   ```
   MESSAGE: ANR1221E BACKUP STGPOOL: Process 7 terminated - insufficient space in
            target storage pool FILECOPYPOOL. (SESSION: 1, PROCESS: 7)

   MESSAGE: ANR1893E Process 7 for BACKUP STORAGE POOL  completed with a
            completion state of FAILURE. (SESSION: 1, PROCESS: 7)
   ```

3. Examine the frequency of your successful database backups to determine if they are running often enough to clear archive log space, provide sufficient recovery points, and allow for appropriate number of volumes to be saved for disaster recovery purposes.

Tip: This SELECT statement is searching the last 24 hours for a full or incremental database backup. You can adjust this statement to search the last 48 hours or longer, depending on the frequency that you have scheduled TYPE=FULL database backups.

```
select count(*) as frequency from summary where end_time>(current_
timestamp - interval '24' hours) and successful='YES' and activity in
 ('FULL_DBBACKUP','INCR_DBBACKUP')
```

Example output:

```
FREQUENCY
------------
    3
```

4. If storage pools are enable for deduplication, ensure that the duplicate-identification processes are completing successfully and processing the expected amount of data:

```
select activity,number,sum(examined) as filesprocessed,sum(affected)
 as duplicateextents,sum(bytes) as duplicatebytes,successful from summary
 where activity='IDENTIFY' and entity='STORAGE_POOL_NAME'
 and end_time>(current_timestamp - interval '24' hours) group by
 activity,number,successful
```

Example output:

```
       ACTIVITY: IDENTIFY
         NUMBER: 5
  FILESPROCESSED: 12946
DUPLICATEEXTENTS: 10504
  DUPLICATEBYTES: 127364341
      SUCCESSFUL: YES
```

Related tasks:

"Monitoring your database daily"

"Monitoring disk storage pools daily" on page 772

"Monitoring sequential access storage pools daily" on page 773

"Monitoring scheduled operations daily" on page 776

Monitoring your database daily

Monitor the database daily to ensure that the database is functioning properly, and that no problems exist.

About this task

For detailed information about the commands mentioned here, see the *Administrator's Reference*.

The following steps describe the commands that you can use to monitor the database:

Procedure

1. Use the **QUERY DBSPACE** command, and then examine the file system information reported through the query to ensure that the file system has adequate space. Examine the total, used, and free space.

```
query dbspace
```

Example output:

```
Location                        Total Space(MB)  Used Space(MB)  Free Space(MB)
------------------------------  -----------      --------------- ---------------
/fvt/kolty/srv/db2/mgsA3/pat-h1 253,952.00       62,066.64       191,821.36
```

2. Examine the file systems where the database is located, using the appropriate operating system commands for the following:
   - Ensure that the file systems are not approaching full.
   - Ensure that other applications, or unexpected users of the file system space are not storing data in the server database directories.
   - Check the operating system and device error logs for any early signs or indications of device failures.
3. Query the database to ensure that the percent utilization is acceptable, and that the remaining space is sufficient for the next few days or weeks of expected activity. This includes examining the free space available, and the free-pages values. If you find that you are approaching your space limits, take action to ensure that you get additional space provisioned to avoid any potential problems.

   ```
   query db format=detailed
   ```

   Example output:

```
                Database Name: mgsA2
Total Size of File System (MB): 253,952
    Space Used by Database(MB): 544
     Free Space Available (MB): 191,821
                   Total Pages: 40,964
                  Usable Pages: 40,828
                    Used Pages: 33,116
                    Free Pages: 7,712
         Buffer Pool Hit Ratio: 97.7
         Total Buffer Requests: 102,279
                Sort Overflows: 0
       Package Cache Hit Ratio: 78.9
  Last Database Reorganization: 08/24/2011 17:28:28
        Full Device Class Name: FILECLASS
  Incrementals Since Last Full: 1
Last Complete Backup Date/Time: 08/25/2011 15:02:31
```

4. Monitor the file systems to ensure that they are not running out of space. Verify that there is enough disk space in the file systems that contain these log files:
   - Active log
   - Archive log
   - Mirror log
   - Archive failover log

   If the archive log directory fills up it will overflow to the active log directory. If you see archive log space file systems filling up, it might be an indication that a database backup is not being run, or not being run often enough. It might also be an indication that the space is shared with other applications that are contending for the same space.

   Issue this command to look at the total space used, free space, and so on.

   ```
   query log format=detailed
   ```

   Example output:

```
              Total Space(MB): 1,000
               Used Space(MB): 0
               Free Space(MB): 995
         Active Log Directory: /fvt/kolty/srv/db2/mgsA3/alog
        Archive Log Directory: /fvt/kolty/srv/db2/mgsA3/archlog
         Mirror Log Directory:
Archive Failover Log Directory: /fvt/kolty/srv/db2/mgsA3/flog
```

5. Examine the instance directory to ensure that it has enough space. If there is insufficient space in this directory, the Tivoli Storage Manager server fails to start.

   You should also examine the `instance_dir/sqllib/db2dump` directory and delete `*.trap.txt` and `*.dump.bin` files regularly.

   V6.2 and later servers:

   - The `db2diag.log` file is managed by Tivoli Storage Manager and is automatically pruned periodically.
   - The **DB2DIAGSIZE** option can be used to control the maximum size of the diagnostic log files.

   V6.1 servers:

   - Servers that are running version 6.1 must periodically delete the `db2diag.log` file.

6. Verify that the database backups completed successfully, and examine the details to determine if there are any problems:

   ```
   select * from summary where end_time>(current_timestamp - interval
   '24' hours) and activity in ('FULL_DBBACKUP','INCR_DBBACKUP')
   ```

   If there are no results to this select command, then there were no database backups in the previous 24-hour period.

   a. Issue the **QUERY PROCESS** command to look at current status of an active backup:

      ```
      query process
      ```

      Example output:

      ```
      Process      Process Description   Status
      Number
      --------     -------------------- -------------------------------------------------
      5            Database Backup       TYPE=FULL in progress. 62,914,560 bytes
                                         backed up to volume /fvt/kolty/srv/Storage/143-
                                         12072.DSS .
      ```

7. Check to ensure that the **DEVCONFIG** and **VOLUMEHISTORY** files configured in the `dsmserv.opt` file are current and up-to-date. Ensure that the file systems where these files are being written to are not running out of space. If there are old or unnecessary volume history entries, consider pruning the old entries using the **DELETE VOLHISTORY** command.

   Important: Save the volume history file to multiple locations. Ensure that these different locations represent different underlying disks and file systems.

Related tasks:

"Monitoring your server processes daily" on page 768

"Monitoring disk storage pools daily" on page 772

"Monitoring sequential access storage pools daily" on page 773

"Monitoring scheduled operations daily" on page 776

Monitoring disk storage pools daily

Monitor disk storage pools daily to ensure that there are no problems, and that your storage pools are fully operational and capable of handling the required workload.

About this task

For detailed information about the commands mentioned here, see the *Administrator's Reference*.

The following steps describe the commands that you can use to monitor disk storage pools:

Procedure

1. Check the status of storage pools, and ensure that there is enough space available.
   - Examine the percent utilization to ensure that the amount of space is sufficient for ingestion rates.
   - The high and low migration thresholds should be set to values that will allow for proper migration cycles.
   - If the storage pool is set to CACHE=YES, the percent migration should be approaching zero.
   - This indicates that items are being cleared out of the pool appropriately.

   Issue the **QUERY STGPOOL** command to display information about one or more storage pools.

   ```
   query stgpool
   ```

   Example output:

   ```
   Storage      Device       Estimated    Pct    Pct  High  Low  Next
   Pool         Class        Capacity     Util   Migr Mig   Mig  Storage-
   Name         Name                                 Pct   Pct  Pool
   -----------  ----------  ----------  -----  -----  ----  ---  -----------
   ARCHIVEPOOL  DISK         1,000.0 M    0.0    0.0    90   70  storage_pool
   BACKUPPOOL   DISK         1,000.0 M    0.0    0.0     5    1  storage_pool
   ```

2. Check the status of the disk volumes. Issue the **SELECT** command and specify a particular device class name:

   ```
   select volume_name, status from volumes
    where devclass_name='devclass name'
   ```

   Example output:

   ```
   VOLUME_NAME: /fvt/kolty/srv/Storage/ar1
        STATUS: ONLINE

   VOLUME_NAME: /fvt/kolty/srv/Storage/bk1
        STATUS: ONLINE
   ```

3. Check for any failed migrations that could contribute to space not being released in a timely manner:

   ```
   select start_time,end_time,activity as process,number as processnum,
    entity as poolname from summary where activity='MIGRATION' and successful='NO'
    and end_time>(current_timestamp - interval '24' hours)
   ```

   Example output:

```
START_TIME: 2011-08-23 14:53:37.000000
  END_TIME: 2011-08-23 14:53:38.000000
   PROCESS: MIGRATION
PROCESSNUM: 7
  POOLNAME: storage_pool_example
```

Related tasks:

"Monitoring your server processes daily" on page 768

"Monitoring your database daily" on page 769

"Monitoring sequential access storage pools daily"

"Monitoring scheduled operations daily" on page 776

Monitoring sequential access storage pools daily

Monitor sequential access storage pools daily to ensure that there are no problems.

About this task

For detailed information about the commands mentioned here, see the *Administrator's Reference*.

The following steps describe the commands that you can use to monitor sequential access storage pools:

Procedure

1. Check the status of your storage pools, and ensure that there is enough space available. Examine the percent utilization to ensure that the amount of space is sufficient for the amount of data that is being taken in. Set the high and low migration thresholds to values that will allow for proper migration cycles.

   Issue the **QUERY STGPOOL** command to display information about one or more storage pools.

   ```
   query stgpool
   ```

   Example output:

   ```
   Storage      Device       Estimated    Pct    Pct  High  Low  Next Storage-
   Pool Name    Class Name   Capacity     Util   Migr Mig   Mig  Pool
                                                     Pct   Pct
   -----------  ----------  ----------  -----  -----  ----  ---  -----------
   ARCHIVEPOOL  DISK         1,000.0 M    0.0    0.0    90   70  storage_pool
   BACKUPPOOL   DISK         1,000.0 M    0.0    0.0     5    1  storage_pool
   ```

2. Check the status of the sequential access storage pool volumes with this **SELECT** command:

   ```
   select volume_name,status,access,write_errors,read_errors,
    error_state from volumes where stgpool_name='STORAGE_POOL_NAME'
   ```

   The select statement can be modified to limit the results based on error state, read-write errors, or current-access state. Example output:

```
VOLUME_NAME: /fvt/kolty/srv/Storage/00000152.BFS
      STATUS: FULL
      ACCESS: READONLY
WRITE_ERRORS: 0
 READ_ERRORS: 0
 ERROR_STATE: NO

 VOLUME_NAME: /fvt/kolty/srv/Storage/00000153.BFS
      STATUS: FULL
      ACCESS: READWRITE
WRITE_ERRORS: 0
 READ_ERRORS: 0
 ERROR_STATE: NO
```

3. Verify that all of the tapes have the appropriate write-access by issuing this command:

```
select volume_name,access from volumes
 where stgpool_name='TAPEPOOL' and access!='READWRITE'
```

For example, this output indicates that the following volumes are not available for use:

```
VOLUME_NAME: A00011L4
     ACCESS: DESTROYED

VOLUME_NAME: KP0033L3
     ACCESS: UNAVAILABLE
```

4. Use the **QUERY DIRSPACE** command to display information about free space in the directories that are associated with a device class with a device type of FILE.

```
query dirspace
```

Example ouput:

```
Device Class   Directory                          Estimated  Estimated
Name                                              Capacity   Available
------------   ---------------------------------  ---------  ---------
FILECLASS      /fvt/kolty/srv/Storage             253,952 M  185,616 M
```

Tip: Ensure that the amount of available space is higher than the total capacity of all storage pools assigned to the device class or classes using that directory.

5. Determine how many scratch volumes are available in tape libraries with this **SELECT** command:

```
select library_name,count(*) "Scratch volumes" from libvolumes
 where status='Scratch' group by library_name
```

Example output:

```
LIBRARY_NAME                Scratch volumes
-------------------------   ---------------
TS3310                           6
```

6. Determine how many scratch volumes can be potentially allocated out of the storage pools using those tape libraries.

```
select stgpool_name,(maxscratch-numscratchused)
 as "Num Scratch Allocatable" from stgpools
 where devclass='DEVICE_CLASS_NAME'
```

Example output:

```
STGPOOL_NAME                        Num Scratch Allocatable
--------------------------------    -----------------------
COPYPOOL                                                 4
TAPEPOOL                                                 2
```

Tip: Ensure that the number of allocatable scratch volumes is equal to the number of available scratch library volumes in the assigned tape library.

7. Issue these **SELECT** commands to determine if there are any tape drives or paths that are offline:
   a. Check to ensure that the drives are online:

```
select drive_name,online from drives
 where online<>'YES'
```

   Example output:

```
DRIVE_NAME                      ONLINE
------------------------------  ----------------------------------------
DRIVEA                          NO
```

   b. Check to ensure that the paths to the drives are also online. A drive can be online, while the path is offline.

```
select library_name,destination_name,online
 from paths where online<>'YES' and destination_type='DRIVE'
```

   Example output:

```
    LIBRARY_NAME: TS3310
DESTINATION_NAME: DRIVEA
          ONLINE: NO
```

8. Check to see if there are any library paths that are offline with this **SELECT** command:

```
select destination_name,device,online from paths
 where online<>'YES' and destination_type='LIBRARY'
```

   Example output:

```
DESTINATION_NAME: TS3310
          DEVICE: /dev/smc0
          ONLINE: NO
```

9. If you are using the DRM, check the status and settings.
   a. Check to see which copy storage pool volumes are onsite:

```
select stgpool_name,volume_name,upd_date,voltype from drmedia
 where state in ('MOUNTABLE','NOTMOUNTABLE')
```

   Example output:

```
STGPOOL_NAME: COPYPOOL
 VOLUME_NAME: CR0000L5
    UPD_DATE: 2011-04-17 16:09:47.000000
     VOLTYPE: Copy
```

   b. Check to see if any volumes are eligible for return to onsite:

```
select stgpool_name,volume_name,upd_date,voltype from drmedia
 where state='VAULTRETRIEVE'
```

Example output:

```
STGPOOL_NAME: COPYPOOL
 VOLUME_NAME: CR0014L5
    UPD_DATE: 2011-08-22 11:12:16.000000
     VOLTYPE: Copy
```

c. Ensure that the DRM settings are valid:

```
select * from drmstatus
```

Example output:

```
   PLANPREFIX:
  INSTRPREFIX:
 PLANVPOSTFIX: @
 NONMOUNTNAME: NOTMOUNTABLE
  COURIERNAME: COURIER
    VAULTNAME: VAULT
DBBEXPIREDAYS: 60
   CHECKLABEL: Yes
  FILEPROCESS: No
  CMDFILENAME:
RPFEXPIREDAYS: 60
```

Related tasks:

"Monitoring your server processes daily" on page 768

"Monitoring your database daily" on page 769

"Monitoring disk storage pools daily" on page 772

"Monitoring scheduled operations daily"

Monitoring scheduled operations daily

Monitor scheduled operations daily to ensure that client and server scheduled operations are completing successfully, and that no problems exist.

About this task

For detailed information about the commands mentioned here, see the *Administrator's Reference*.

The following steps describe the commands that you can use to monitor scheduled operations:

Procedure

1. The most valuable command that you can use to check the status of your scheduled operations is the **QUERY EVENT** command. Issue this command and look for any missed or failed scheduled operations that might indicate a problem:

   ```
   query event * * type=client
   query event * type=admin
   ```

   Tip: The **EXCEPTIONSONLY** parameter can be used to control what type of information you want displayed on scheduled or completed events.
2. Use this **SELECT** command to specifically check for missed schedules:

   ```
   select time(scheduled_start)"Start Time",node_name"Client"
    from events where status='Missed'
   ```
3. Use this **SELECT** command to specifically check for any failed schedules:

```
select time(scheduled_start)"Start Time",node_name"Client"
 from events where status='Failed'
```

4. Check the activity log for any unexpected messages that might indicate a problem. Client messages are reported to the server with a prefix of ANE. Review the messages to examine the client view of operations such as successes, failures, and skipped files.

   ```
   query actlog search=ANE????E begindate=today
   query actlog search=ANE????W begindate=today
   query actlog search=ANE????S begindate=today
   ```

5. Check the activity log for server messages, which use the ANR prefix:

   ```
   query actlog search=ANR????E begindate=today
   query actlog search=ANR????W begindate=today
   query actlog search=ANR????S begindate=today
   query actlog search=ANR9999D begindate=today
   ```

6. Check the summary table for various scheduled client operations such as BACKUP, RESTORE, ARCHIVE, and RETRIEVE.

   ```
   select * from summary where schedule_name is not null and end_time>
    (current_timestamp - interval '24' hours) and activity in
    ('BACKUP','ARCHIVE','RETRIEVE','RESTORE')
   ```

   The select command can be modified to limit the results based on schedule, entity, success, failure, and so on. Analyze the results and compare to the previous results to determine if there are any anomalies or other indicators of potential issues.

7. Check for scheduled server processes such as MIGRATION, RECLAMATION, and EXPIRATION. This **SELECT** command reports successes and failures for the operations:

   ```
   select * from summary where schedule_name is not null and end_time>
    (current_timestamp - interval '24' hours) and activity in
    ('EXPIRATION','MIGRATION','RECLAMATION','STGPOOL BACKUP',
    'FULL_DBBACKUP','INCR_DBBACKUP')
   ```

Related tasks:

"Monitoring your server processes daily" on page 768

"Monitoring your database daily" on page 769

"Monitoring disk storage pools daily" on page 772

"Monitoring sequential access storage pools daily" on page 773

Chapter 23. Basic monitoring methods

You can monitor operations using a variety of methods, such as using queries to display information about settings and status, gain insight into database operations, or display messages in the activity log.

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.

About this task

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 784.

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.

About this task

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       Days      Days  Locked?
                      Domain      Since     Since
                      Name         Last  Password
                                 Access       Set
----------  --------  ---------  ------  --------  -------
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:

© Copyright IBM Corp. 1993, 2013

```
                    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.

About this task

You can use the QUERY SESSION command to request information about client sessions. Figure 91 shows a sample client session report.

```
  Sess Comm.  Sess     Wait   Bytes   Bytes Sess  Platform Client Name
Number Method State    Time    Sent   Recvd Type
------ ------ ------ ------ ------- ------- ----- -------- --------------------
     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 91. 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.

About this task

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 92 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 Process Description      Status
  Number
-------- ----------------------- ---------------------------------------------
       2 DELETE FILESPACE         Deleting filespace DRIVE_D for node CLIENT1:
                                   172 files deleted.
```

Figure 92. 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.

About this task

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.

About this task

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 the QUERY OPTION command in the *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.

About this task

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)

QUERY VOLHISTORY
Displays sequential volume history information that has been collected by the server.

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.

About this task

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://pic.dhe.ibm.com/infocenter/db2luw/v10r5. 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.

About this task

For SELECT statement syntax and guidelines, refer to DB2 documentation: http://pic.dhe.ibm.com/infocenter/db2luw/v10r5. 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.

About this task

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.

About this task

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: 4445.000000
      Bytes sent/second: 564321.9302768451
  Bytes received/second: 0.0026748857944

               Session: 26
                Client: MILTON
                 State: Run
          Elapsed Time: 373.000000
      Bytes sent/second: 1638.5284210992221
  Bytes received/second: 675821.6888561849
```

Tip: When issuing **SELECT * FROM DB** statements, the output column PHYSICAL_VOLUMES is included for legacy compatibility purposes only. 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 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.

About this task

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 "Tivoli Storage Manager server scripts" on page 616 for details.

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.

About this task

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 tabname='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
            MEDIAW: 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.

About this task

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 Tivoli Storage Manager activity log

The activity log contains messages that are 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.

About this task

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

You can also query the activity log for client session information. For example, issue the following command to search the activity log for any messages that were issued in relation to session 4:

```
query actlog search="(SESSION:4)"
```

This command displays the output of the query:

```
    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 dismounted
   from drive OUTFILE4 (FILE) in library OUTFILE.  (SESSION:4)
09/23/2003 10:26:40  ANR0403I Session 4 ended for node FRED (WinNT).
   (SESSION: 4)
```

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.

About this task

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
```

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.

About this task

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 30 days. 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.

About this task

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."

Chapter 24. Monitoring 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.

About this task

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 `dsmaccnt.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 `dsmaccnt.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
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

© Copyright IBM Corp. 1993, 2013

Field	Contents
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
```

Chapter 25. Reporting and monitoring with Tivoli Monitoring for Tivoli Storage Manager

Tivoli Monitoring for Tivoli Storage Manager is not delivered with Tivoli Storage Manager Version 7.1.

You can use Tivoli Monitoring for Tivoli Storage Manager Version 6.3.4 with any Tivoli Storage Manager Version 5.5 or later server. For more information about installing and configuring Tivoli Monitoring for Tivoli Storage Manager, see Technote 1649883 at: http://www.ibm.com/support/docview.wss?uid=swg21649883.

© Copyright IBM Corp. 1993, 2013

Chapter 26. 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*.

About this task

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 799.

File and user exits
See "Logging events to a file exit and a user exit" on page 800.

Tivoli event console
See "Logging events to the Tivoli Enterprise Console" on page 801.

Event server receiver (Enterprise Event Logging)
Routes the events to an event server. See "Enterprise event logging: logging events to another server" on page 811.

Simple Network Management Protocol (SNMP)
See "Logging events to an SNMP manager" on page 805.

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 93 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.

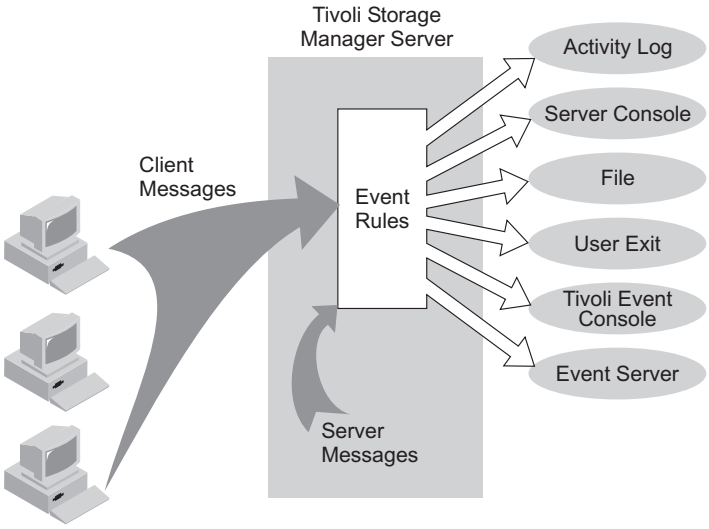


Figure 93. Event logging overview

© Copyright IBM Corp. 1993, 2013

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 799.

Enabling and disabling events

You can enable and disable events using the **ENABLE EVENTS** and **DISABLE EVENTS** commands.

About this task

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.

About this task

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.

About this task

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 791 and “Setting a size limit for the activity log” on page 791). 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 Tivoli Storage Manager activity log” on page 789.

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.

About this task

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 816.

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.

Procedure

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 798 for more information.

3. 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 799 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.

About this task

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.

About this task

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
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.

About this task

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.

About this task

To set up Tivoli as a receiver for event logging, complete the following procedure:

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 IBMTSMSERVER_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 798 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 799 for more information.

Logging events to an SNMP manager

IBM Tivoli Storage Manager supports the simple network management protocol (SNMP) together with event logging.

About this task

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 "Tivoli Storage Manager server scripts" on page 616 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:

Procedure

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.

Results

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.

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 94 illustrates a typical Tivoli Storage Manager implementation:

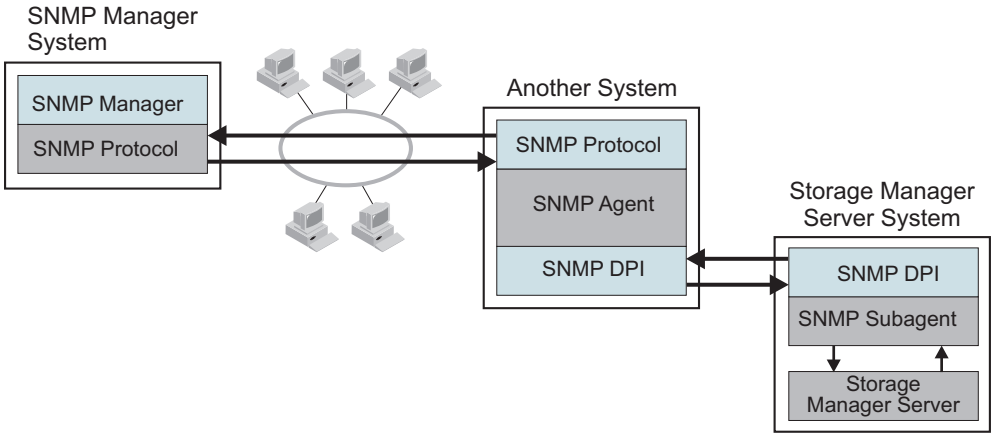


Figure 94. Tivoli Storage Manager SNMP Implementation

Figure 95 on page 809 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.

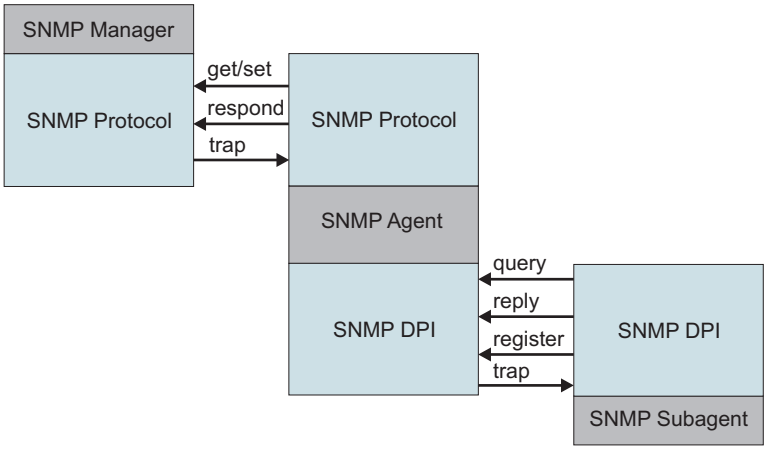


Figure 95. Manager-Agent-Subagent communication

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 *adsmserv.mib*. The file name is located in the directory in which the server is installed.
3. Merge the contents of the *adsmserv.mib* file into the */etc/mib.defs* file.

Configuring IBM Tivoli Storage Manager SNMP

You can configure SNMP by completing the following procedure.

About this task

The IBM Tivoli Storage Manager SNMP set up procedure is illustrated by Figure 96:

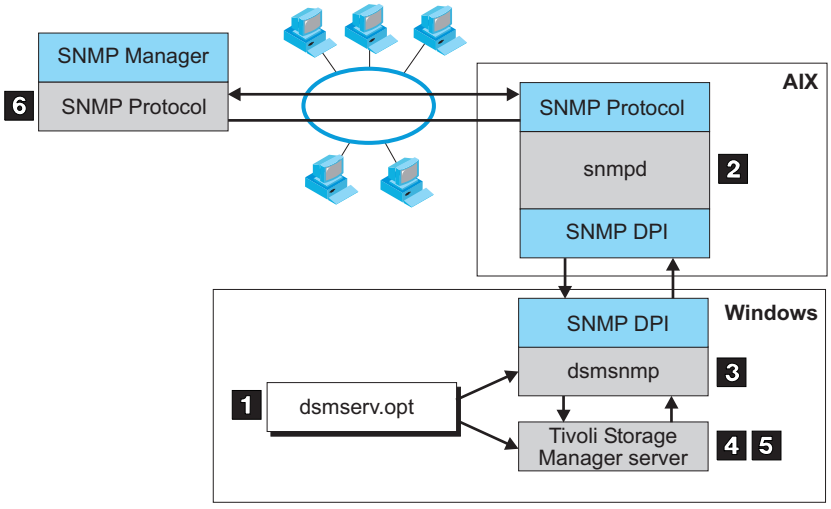


Figure 96. IBM Tivoli Storage Manager SNMP Set Up

To set up Tivoli Storage Manager monitoring through SNMP, do the following:

Procedure

1. Modify the server options file to specify the SNMP communication method. Figure 97 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 97. 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
```

6. 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 *adsmserv.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 *adsmserv.mib* file is copied to the *\netview_path\SNMP_MIB* directory and then loaded through the following command:

```
[C:\] loadmib -load adsmserv.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.

About this task

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 98 shows the relationship of a sending Tivoli Storage Manager server and a Tivoli Storage Manager event server.

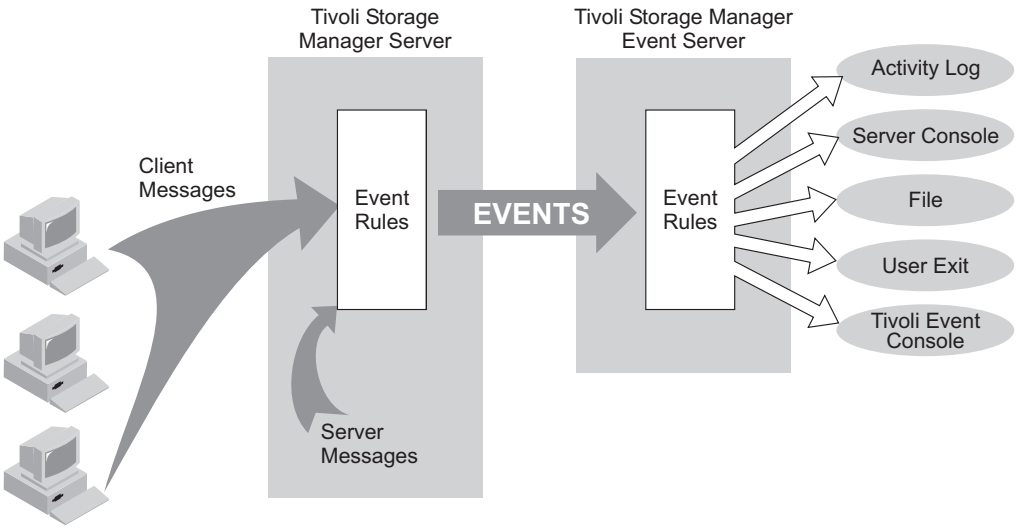


Figure 98. Server-to-server event logging

The following scenario is a simple example of how enterprise event logging can work.

The administrator at each sending server does the following:

1. 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 682.

   ```
   define server server_b password=cholla hladdress=9.115.3.45 lladdress=1505
   ```

2. Identifies the server just defined as the event server:

   ```
   define eventserver server_b
   ```

3. 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=*
   ```

4. 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.

About this task

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/IBMadsm-s/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 Chapter 26, "Logging IBM Tivoli Storage Manager events to receivers," on page 797 and *Administrator's Reference*.

Sample user exit declarations

userExitSample.h contains declarations for a user-exit program.

The environment is:

Oracle Solaris

Figure 99. 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 elEventRecvData.eventType) *
 *******************************************/

#define TSM_SERVER_EVENT       0x03  /* Server Events */
#define TSM_CLIENT_EVENT       0x05  /* Client Events */

/*************************************************
 * Application Types (in elEventRecvData.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 elEventRecvData.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 100. Sample user exit program

```
/*****************************************************************************
 * Name:            userExitSample.c
 * Description:     Example user-exit program invoked by the server
 * Environment:     Solaris
 *****************************************************************************/

#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 65. 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 (YYYYMMDDDHHmmSS)
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
0358-0387[1]	Schedule name
0389-0418[1]	Domain name

Table 65. Readable text file exit (FILETEXTEXIT) format (continued)

Column	Description
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.

Part 6. Protecting the server

Disasters, by their very nature, cannot be predicted, in either their intensity, timing, or long-term effects. The ability to recover from a disaster, if one occurs, is essential. To protect your system infrastructure and data and to recover from a disaster, use the tools and procedures that Tivoli Storage Manager provides.

© Copyright IBM Corp. 1993, 2013

Chapter 27. Managing Tivoli Storage Manager security

Administrators have specific activities to manage IBM Tivoli Storage Manager security.

The security of your data is the most important aspect of managing data. You can control access to the server and client nodes, encrypt data transmission, and protect administrator and node passwords through authentication processes. The two methods of authentication are LOCAL and LDAP. The LOCAL password authentication takes place on the Tivoli Storage Manager server, and those passwords are not case-sensitive.

LDAP password authentication takes place on the LDAP directory server, and the passwords are case-sensitive. When using LDAP authentication, the password is sent to the server by the client. By default, Secure Sockets Layer (SSL) is required when LDAP authentication is used, to avoid exposing the password. SSL is used when authenticating the server to the client and secures all communication between the client and server. You can choose not to use SSL with LDAP authentication if other security measures are in place to protect the password. One example of an alternative security measure is a virtual private network (VPN) connection.

Related concepts:

"Managing Tivoli Storage Manager administrator IDs" on page 834

"Managing passwords and logon procedures" on page 843

"Securing the server console" on page 833

"Securing sensitive client data" on page 533

Related reference:

"Managing access to the server and clients" on page 834

"Administrative authority and privilege classes" on page 835

Securing communications

You can add more protection for your data and passwords by using Secure Sockets Layer (SSL).

SSL is the standard technology for creating encrypted sessions between servers and clients. SSL provides a secure channel for servers and clients to communicate over open communication paths. With SSL, the identity of the server is verified through the use of digital certificates.

To ensure better system performance, use SSL only for sessions when it is needed. Consider adding additional processor resources on the Tivoli Storage Manager server to manage the increased requirements.

If you authenticate passwords with an LDAP directory server, Transport Layer Security (TLS) protects passwords between the Tivoli Storage Manager server and the LDAP server. TLS, a form of SSL, is required for all LDAP password communications. See "Configuring SSL or TLS for LDAP directory servers" on page 829.

© Copyright IBM Corp. 1993, 2013

Secure Sockets Layer and Transport Layer Security communication

You can use the Secure Sockets Layer (SSL) or the Transport Layer Security (TLS) protocol to provide transport layer security for a secure connection between Tivoli Storage Manager servers, clients, and storage agents. If you send data between the server, client, and storage agent, use SSL or TLS to encrypt the data.

Tip: Any Tivoli Storage Manager documentation that indicates "SSL" or to "select SSL" applies to TLS.

SSL is provided by the Global Security Kit (GSKit) that is installed with the Tivoli Storage Manager server that the server, client, and storage agent use. The Operations Center and Reporting agent do not use GSKit.

Each Tivoli Storage Manager server, client, or storage agent that enables SSL must use a trusted self-signed certificate or obtain a unique certificate that is signed by a certificate authority (CA). You can use your own certificates or purchase certificates from a CA. Either certificate can be installed and added to the key database on the Tivoli Storage Manager server, client, or storage agent. The certificate is verified by the SSL client or server that requests or initiates the SSL communication.

SSL is set up independently on the Tivoli Storage Manager server, client, and storage agent.

The Tivoli Storage Manager server, client, or storage agent can serve as SSL clients during communication. An SSL client is the component that initiates communication and verifies the certificate for an SSL server. For example, if a Tivoli Storage Manager client initiates the SSL communication with a Tivoli Storage Manager server, the Tivoli Storage Manager client is the SSL client and the server is the SSL server.

For more information about the components that can be an SSL client or SSL server, see Table 66.

Table 66. SSL clients and servers in a Tivoli Storage Manager environment

SSL client	SSL server	Description
Client	Server	The Tivoli Storage Manager client initiates a communication request with the Tivoli Storage Manager server. The client verifies the certificate. The server provides the certificate.
Server (such as a source server)	Server (such as a target server)	The Tivoli Storage Manager source server initiates a communication request with the Tivoli Storage Manager target server. The source server acts as an SSL client and verifies the certificate that the target server provides. This type of communication is common during replication processing.

Table 66. SSL clients and servers in a Tivoli Storage Manager environment (continued)

SSL client	SSL server	Description
Client through a storage agent	Server	The client uses the Tivoli Storage Manager storage agent to verify the certificate that the Tivoli Storage Manager server provides. When the storage agent communicates with the server by using the SSL communication protocol, the storage agent acts as an SSL client and verifies the certificate that the server provides. The storage agent can be the SSL client and the SSL provider at the same time.
Server	LDAP server	The Tivoli Storage Manager server initiates a communication request with the LDAP server. The Tivoli Storage Manager server acts as the SSL client and verifies the certificate that the LDAP server provides.
Operations Center	Server	The Operations Center initiates a communication request with the Tivoli Storage Manager server. The Operations Center acts as the SSL client and verifies the certificate that the Tivoli Storage Manager server provides.
Reporting	Server	The Reporting agent initiates a communication request with the Tivoli Storage Manager server. The Reporting feature acts as the SSL client and verifies the certificate that the Tivoli Storage Manager server provides.

Setting up SSL

You can set up SSL or TLS on the Tivoli Storage Manager server, backup-archive client, and storage agent to ensure that your data is encrypted during communication. You can use an SSL certificate to verify an SSL communication request between the server, client, and storage agent.

Procedure

To configure Tivoli Storage Manager servers and clients for SSL or TLS, complete the following steps:

1. Specify the TCP/IP port on which the server waits for client communications that are enabled for SSL or TLS. You can use the `SSLTCPADMINPORT` option or `SSLTCPPORT` option, or both, to specify TLS port numbers. The options are stored in the `dsmserv.opt` file.
2. Create the key database file if it does not exist. Complete the following steps to create the key database file for the server, client, and storage agent:
   - For Tivoli Storage Manager V6.3.3 and later servers, the `cert256.arm` file and other files that are related to SSL or TLS are created when the server is first started. Tivoli Storage Manager automatically creates the server key database file, `cert.kdb`. If a password exists for the server database, it is reused for the `cert.kdb` key database. After you create the database, the key database access password is generated and stored.
   - To create the key database file for the storage agent, issue the **DSMSTA SETSTORAGESERVER** command and specify the **SSL=YES** and **STAKEYDBPW=password** parameters.
   - To create the key database file, `dsmcert.kdb`, for the client, issue the following command in the `bin` directory on the client:

```
gsk8capicmd_64 -keydb -create -populate
-db dsmcert.kdb -pw password -stash
```

3. Use one of the following certificates for SSL or TLS communication:

   Self-signed certificate
   You must import a `.arm` file for the server, backup-archive client, and storage agent according to the default label that is used for the server self-signed certificate. The following table shows you which file to import:

Table 67. Determining the `.arm` file to use

Default label in the key database	Import this file for clients	Import this file for server-server communication	Import this file for storage agent-server communication
"TSM Server SelfSigned Key"	`cert.arm`	`cert256.arm`	`cert256.arm`
"TSM Server SelfSigned SHA Key"	`cert256.arm`	`cert256.arm`	`cert256.arm`

   Important: To use TLS 1.2, the default label must be "TSM Server SelfSigned SHA key". You must specify the **SSLTLS12 YES** server option in the server options file and the storage agent options file, if necessary.

   CA-signed certificate
   You must obtain a unique certificate that is signed by a CA or use a trusted self-signed certificate for each server that enables SSL or TLS. Backup-archive clients use the `cert.kdb` or `cert256.arm` files to import the self-signed certificates, which the server automatically generates.

4. Manually transfer the appropriate Tivoli Storage Manager server `.arm` file to the client computers. If you transfer the `cert256.arm` file, you must first change the default certificate in the `cert.kdb` file to the "TSM Server SelfSigned SHA Key" label. To change the default certificate, issue the following command from the server instance directory:

```
gsk8capicmd_64 -cert -setdefault -db cert.kdb
-stashed -label "TSM Server SelfSigned SHA Key"
```

5. Using a backup-archive client user ID, specify the `ssl` **yes** and `tcpport` options in the client options file:
   - `dsm.sys`

   The server is normally set up for SSL or TLS connections on a different port. If you use an SSL or TLS connection, two ports are open on the server. One port accepts regular non-SSL or non-TLS client connections and the other port accepts SSL or TLS connections only.

6. If you want to use a certificate that is issued by a certificate authority (CA), you do not need to complete steps 4 and 5. Install the CA root certificate on all clients. A set of default root certificates are preinstalled if you specified the `-populate` parameter in the command when you created the key database file.

Related reference:

"Adding a certificate to the key database" on page 826

Specifying communication ports

The Tivoli Storage Manager server can be configured to listen on four TCP/IP ports: two for regular protocols and two for the Transport Layer Security (TLS) protocols.

If you authenticate passwords with an LDAP directory server, the connection between the LDAP directory server and the Tivoli Storage Manager server must be protected. The connection between a Tivoli Storage Manager server and an LDAP directory server defaults to port 389. You do not have to use this port number and can define the port by setting the **LDAPURL** option.

For IPv4 or IPv6, the **COMMMETHOD** server option must specify either TCPIP or V6TCPIP. The server options for TLS communications are **SSLTCPPORT** and **SSLTCPADMINPORT**. The server can listen on separate ports for the following communications:

- Backup-archive clients that use the regular protocol
- Administrator IDs that use the regular protocol
- Backup-archive clients that use the TLS protocol
- Administrator IDs that use the TLS protocol

Use the **TCPADMINPORT** and **SSLTCPADMINPORT** options to separate administrative client traffic from regular client traffic on **TCPPORT** and **SSLTCPPORT** options. If the **TCPADMINPORT** and **SSLTCPADMINPORT** options are not used, administrative traffic flows on client ports as well.

You can use the following components with TLS:

- Command-line client
- Administrative command-line client
- Backup-archive client graphical user interface (GUI)
- Client API

If the **ADMINONCLIENTPORT** option is set to NO, TLS administrative client sessions require that you specify the **SSLTCPADMINPORT** option with a port number other than the one specified by the **SSLTCPPORT** option. The **SSLTCPPORT** and **SSLTCPADMINPORT** options do not affect the **TCPPORT** or **TCPADMINPORT** options and their interaction with the **ADMINONCLIENTPORT** option.

The backup-archive client user decides which protocol to use and which port to specify in the `dsmserv.opt` file for the **SSLTCPADMINPORT** option. If the backup-archive client requires TLS authentication but the server is not in TLS mode, the session fails.

Related concepts:

"Managing passwords and logon procedures" on page 843

Related tasks:

"Configuring Tivoli Directory Server for TLS on the iKeyman GUI" on page 830

"Configuring Tivoli Directory Server for TLS on the command line" on page 831

Related reference:

"Configuring Windows Active Directory for TLS/SSL" on page 832

Adding a certificate to the key database

To use Transport Layer Security (TLS), the certificate must be installed on the server, and for some root certificates, they must be installed on the client. Each server that enables TLS must obtain a unique certificate signed by a certificate authority (CA) or use a trusted 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 include the **-stashpw** parameter on a GSKit **gsk8capicmd_64** command, the password that you define is saved for later use. The server and the storage agent create the key databases by using the stash facility.

The key database is created when you start the Tivoli Storage Manager server. 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 start.

For example, to add a certificate to the key database for the client, issue the following command:

```
gsk8capicmd_64 -cert -add -label "TSM061" -format ascii
-file cert256.arm -db dsmcert.kdb -pw password
```

The `cert256.arm` file is generated by the V6.3 server for distribution to the V6.3 or later backup-archive clients, and for use in server-server communication and storage agent-server communication. The `cert.arm` file might also be generated by the V6.3 server, but is not designed for passwords that authenticate with an LDAP server.

If a certificate expires, the certificate is rejected when you attempt SSL communication. You must obtain a new certificate and add it to the key database for the server and the storage agent. The self-signed certificates for the server are created with an expiration time of 10 years. After 10 years, the certificates must be re-created and redistributed. You can re-create a self-signed certificate by deleting the `cert.arm` and `cert256.arm` files and by deleting the self-signed certificates from the key database. When you restart the server or the storage agent, new certificates and `.arm` files are created.

The first time that you enter a password you can use the `-stashpw` *password* parameter. The password is then saved in the `*.sth` file and retrieved when you use the **-stashed** parameter in later commands. You can use the `-stashpw` *password* parameter with the **CREATE** and **POPULATE** parameters.

To show the certificates that are available to use for SSL or TLS communication, issue the following command from the client directory:

```
gsk8capicmd_64 -cert -list -db dsmcert.kdb -stashed
```

Related concepts:

"Managing passwords and logon procedures" on page 843

Adding a CA-signed SSL certificate:

You can use a Transport Layer Security (TLS) 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.

About this task

The Global Security Kit (GSKit) is included in the Tivoli Storage Manager server installation. The backup-archive client and server communicate with SSL through services provided by GSKit.

Procedure

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. To receive the signed certificate and make it the default for communicating with clients, issue the following command:

   ```
   gsk8capicmd_64 -cert -receive -db cert.kdb
   -pw password -stash -file cert_signed.arm -default_cert yes
   ```

   The server key database file name is `cert.kdb`.

   Important: If your client operating system is 32-bit, replace `gsk8capicmd_64` with `gsk8capicmd` in all GSKit commands.
3. Restart the server.
4. Transfer the root certificate (`ca.arm`) to the client directory.
5. To add the root certificate to the client key database, issue the `gsk8capicmd_64 -cert -add` command. For example:

   ```
   gsk8capicmd_64 -cert -add -db dsmcert.kdb
   -pw password -label "my CA"
   -file ca.arm -format ascii
   ```

   Tip: For this example, the client key database name is `dsmcert.kdb`.
6. To verify that the client can successfully connect, issue the `dsmc query session` command.

Troubleshooting the certificate key database

Backup copies of the `cert.kdb` file ensure that Transport Layer Security (TLS) starts when you restore the Tivoli Storage Manager server. If you have a backup copy, you can restore the file and restart the server.

Procedure

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 backup-archive clients that are using TLS. Reinstall any third-party certificates on the backup-archive client. If you are using an LDAP directory server to authenticate passwords, add the root certificate that was used to sign the LDAP server's certificate. If the root certificate is already a default trusted certificate, you do not have to add it again.

Configuring a storage agent and server to use SSL

You can set up a storage agent and the Tivoli Storage Manager server to use the SSL communication method. SSL is set up independently on both the storage agent and the Tivoli Storage Manager server.

Procedure

To set up the storage agent to use SSL communication with the Tivoli Storage Manager server and client, complete the following steps:

1. On the storage agent, issue the **DSMSTA SETSTORAGESERVER** command to initialize the storage agent and add communication information to the device configuration file and the storage agent options file `dsmsta.opt`:

   Hint: The following command is entered on one line, but is displayed here on multiple lines to make it easier to read.

   ```
   dsmsta setstorageserver myname=sta
   mypa=sta_password
   myhla=ip_address
   servername=server_name
   serverpa=server_password
   hla=ip_address
   lla=ssl_port
   STAKEYDBPW=password
   ssl=yes
   ```

   Requirement:

   - When you set the **SSL=YES** and **STAKEYDBPW=password** parameters, a key database file is set up in the storage agent options file, `dsmsta.opt`. All passwords are obfuscated in `dsmsta.opt`.
   - To enable SSL communication, ensure that the Tivoli Storage Manager **LLA** parameter specifies the server **SSLTCPADMIN** port. If the **SSLTCPADMIN** is not specified, use the **SSLTCPPORT** port instead. Set the **SSL** parameter to YES.
2. Import the Tivoli Storage Manager server certificate, `cert256.arm`, to the key database file for the storage agent. Ensure that the required SSL certificates are in the key database file that belongs to each storage agent that uses SSL communication. To import the SSL certificate, switch to the storage agent directory and issue the following command:

   ```
   gskcapicmd_64 -cert -add -label server_example_name
   -db cert.kdb -stashed -file cert256.arm -format ascii
   ```
3. Specify the `SSLTCPPORT` and the `SSLTCPADMINPORT` options in the `dsmsta.opt` options file.
4. Create the key database certificate and default certificates by starting the storage agent.

   Tip: To provide the new password to the storage agent, you must change the key database password and then issue the **DSMSTA SETSTORAGESERVER** command.

a. Open a command window and change the password by issuing the following command:

   ```
   gsk8capicmd_64 -keydb -changepw -db cert.kdb -pw oldpw -newpw newpw
   ```

b. Rerun the **DSMSTA SETSTORAGESERVER** command and specify the **STAKEYDBPW=newpassword** parameter.

5. On the Tivoli Storage Manager server, issue the following command:

   ```
   define server sta
   hla=ip_address
   lla=ssl_port
   serverpa=password
   ssl=yes
   ```

6. Stop the storage agent.
7. Stop the Tivoli Storage Manager server.
8. Import the `cert256.arm` certificate from the storage agent to the key database file for the Tivoli Storage Manager server. Ensure that the required SSL certificates are in the key database file that belongs to each server that uses SSL communication before you restart the server. To import the SSL certificate from the storage agent, issue the following command:

   ```
   gskcapicmd_64 -cert -add -label server_example_name
   -db cert.kdb -stashed -file cert256.arm -format ascii
   ```

9. Stop and restart the Tivoli Storage Manager server.
10. Restart the storage agent.

Results

When the Tivoli Storage Manager server and storage agent initiate communication, SSL certificate information is displayed to indicate that SSL is in use.

Related reference:

"Adding a certificate to the key database" on page 826

Configuring SSL or TLS for LDAP directory servers

Before you can authenticate passwords by using an LDAP directory server, you must configure Secure Sockets Layer (SSL) or Transport Layer Security (TLS) to use the directory server. You must use a trusted certificate that is verified during SSL or TLS communication.

TLS is provided by the Global Security Kit (GSKit) that is installed with the Tivoli Storage Manager server.

If you use an LDAP directory server to authenticate passwords, you can use the SSL or TLS protocol to secure connections between the Tivoli Storage Manager server and the LDAP server. When you configure the Tivoli Storage Manager server to work with an LDAP directory server, you must consider the directory server to use before you configure SSL or TLS.

The LDAP directory server must supply a trusted certificate to the Tivoli Storage Manager server. If the Tivoli Storage Manager server determines that the certificate is valid, an SSL or TLS connection is established. If not, the connection fails. The root certificate that is used with the LDAP Directory server certificate must be added to the key database file for the Tivoli Storage Manager server. If the root certificate is not added, the LDAP Directory server certificate cannot be used.

The directory servers that are available are IBM Tivoli Directory Server V6.2 or 6.3 or Windows Active Directory 2003 or 2008. You can configure Tivoli Directory

Server with the graphical user interface or with the command line interface.

Configuring Tivoli Directory Server for TLS on the iKeyman GUI:

Configuring IBM Tivoli Directory Server is one of the preliminary tasks you must do before you can authenticate passwords with an LDAP directory server. The Tivoli Directory Server can use a self-signed certificate to secure the communication between server and backup-archive client, and the LDAP directory server.

About this task

You can use the iKeyman graphical user interface (GUI) to set up Tivoli Directory Server. If the Tivoli Storage Manager server already has a trusted certificate from your LDAP server, you do not have to complete the steps that are documented here. If the LDAP directory server already has a signed certificate, you do not have to complete these steps.

The X Window System client must be installed on the operating system where Tivoli Directory Server is installed. Ensure that the X Window System server is running on the local system. You must also set the DISPLAY environment variable.

Procedure

To configure Tivoli Directory Server for Transport Layer Security (TLS) by using the iKeyman GUI, complete the following steps:

1. Install and configure Java Runtime Environment 1.4.1 or later before you install Tivoli Directory Server.
2. Ensure that you have GSKit version 8 installed. See Instructions to install, uninstall, and upgrade GSKit 8 (http://www.ibm.com/support/docview.wss?uid=swg21577384).
3. Set the JAVA_HOME environmental variable to point to the Java subdirectory of the Tivoli Directory Server installation directory. The installation directory is `/opt/IBM/ldap/V`*x.x*, where *x.x* is the version number.
4. Ensure that the `local_policy.jar` and `US_export_policy.jar` files are in the `$JAVA_HOME/jre/lib/security` directory.
5. Verify whether the Content Management system (CMS) provider is registered. Open the `$JAVA_HOME/jre/lib/security/java.security` file to see whether the entry exists. If the entry does not exist, open the `java.security` file and add the following entry:

   ```
   security.provider.X=com.ibm.security.cmskeystore.CMSProvider
   ```

   where X is the next number in the order. For example:

   ```
   security.provider.1=com.ibm.jsse.IBMJSSEProvider
   security.provider.2=com.ibm.egypto.provider.IBMJCE
   security.provider.3=com.ibm.security.cmskeystore.CMSProvider
   ```

6. Create the key database file in the Tivoli Directory Server installation directory as Tivoli Directory Server instance user. Complete the following steps:
   a. Open a command prompt and enter `ikeyman` to start the Java utility.
   b. Click **Key Database File** > **New** for a new key database file or click **Open** if the key database file exists.
   c. Specify the key database type (CMS) and the key database file name, with its location. For example, *server_file*`.kdb`. Click **OK**.
   d. Supply the password for the key database file when you are prompted.

e. Select the **Stash a password to a file** check box and click **OK**.

7. Click **Create** > **New Self-Signed Certificate**.
8. Extract the certificate.
   a. On the server where Tivoli Directory Server is, open the *server_file*.kdb file.
   b. Click **Personal Certificates** > **Extract Certificate**.
   c. Enter the file name and location.
   d. Save the certificate to a text-based (ASCII) file.
9. Click **Server administration** > **Manage security properties** > **Key database**. Specify the key database path and file name. If you defined a password stash file, it is assumed to have the same file specification, with an extension of .sth. See step 6e.
10. Specify the key password. If you are not using a stashed password, specify the password for the key database file.
11. Specify the key label, which indicates the part of the key database that is to be used.
12. Copy the certificate file (ldapcert.arm) to the Tivoli Storage Manager server key database.

Configuring Tivoli Directory Server for TLS on the command line:

You must configure IBM Tivoli Directory Server before you can authenticate passwords with an LDAP directory server. The Tivoli Directory Server can use a self-signed certificate to secure the communication between server and backup-archive client, and the LDAP directory server.

About this task

If the Tivoli Storage Manager server already has a trusted certificate from your LDAP server, you do not have to complete the steps that are documented here. If the LDAP directory server already has a signed certificate, you do not have to complete these steps.

Procedure

To configure Tivoli Directory Server for Transport Layer Security (TLS), complete the following steps:

1. Using the Tivoli Directory Server instance user name, create the key database by issuing the following command:

   ```
   gsk8capicmd_64 -keydb -create -db "directory/filename.kdb"
   -pw "pa$$=w0rd" -stashpw -populate
   ```

2. Create a self-signed certificate or get one from a certificate authority (CA). To create a self-signed certificate, issue the following command:

   ```
   gsk8capicmd_64 -cert -create -db "directory/filename.kdb" -stashed -label
   "LDAP_directory_server" -dn "cn=ldapserver.company.com"
   -san_dnsname ldapserver.company.com -size 2048
   -sigalg SHA256WithRSA -expire 3650
   ```

3. Extract the certificate to a file by issuing the following command:

   ```
   gsk8capicmd_64 -cert -extract -db "directory/filename.kdb" -stashed -label
   "LDAP_directory_server" -target ldapcert.arm -format ascii
   ```

4. Copy the certificate file (ldapcert.arm) to the Tivoli Storage Manager server.

5. To add the certificate to the Tivoli Storage Manager server key database, issue the following command from the Tivoli Storage Manager server. You must issue the command from the instance user ID from the instance directory.

   ```
   gsk8capicmd_64 -cert -add -db "cert.kdb" -stashed -label
   "LDAP_directory_server" -format ascii -file ldapcert.arm
   ```

6. Configure the key database file to work with Tivoli Directory Server. To set the key database for TLS, issue the following command:

   ```
   idsldapmodify -D <adminDN> -w <adminPW> -i <filename>
   ```

   where *filename* contains the following entry:

   ```
   dn: cn=SSL,cn=Configuration
   changetype: modify
   replace: ibm-slapdSSLKeyDatabase
   ibm-slapdSSLKeyDatabase: <databasename>
   -
   replace: ibm-slapdSSLKeyDatabasePW
   ibm-slapdSSLKeyDatabasePW: <password>
   -
   replace: ibm-slapdSslKeyRingFilePW
   ibm-slapdSslKeyRingFilePW: <password>
   ```

7. Enable TLS on the LDAP directory server so that the Tivoli Storage Manager server can run the "start TLS extended" operation. This operation secures LDAP communication. Do not enable client authentication with TLS, because the Tivoli Storage Manager server does not support certificate authentication as an LDAP client.

   Tip: The Tivoli Storage Manager server authenticates with the "LDAP simple password authentication" method.

8. Restart Tivoli Directory Server and Tivoli Storage Manager server.

Configuring Windows Active Directory for TLS/SSL:

You must configure Windows Active Directory before the Tivoli Storage Manager server can authenticate passwords.

To set up the Windows Active Directory server, complete the following steps:

1. Turn off automatic root certificate updates to Windows Update if your Windows Active Directory server does not have access to the internet.
2. Synchronize the system times of the Tivoli Storage Manager server and the Windows Active Directory system. You can use a Network Time Protocol (NTP) server. For more information about synchronizing the system times, see your operating system documentation. You can also see the Microsoft website for information about synchronizing Active Directory (http://technet.microsoft.com/en-us/library/cc786897).
3. Set up Transport Layer Security (TLS) for LDAP server connections. Go to the Microsoft website (http://www.microsoft.com) and search for *LDAP* and *SSL*.
   a. Obtain a signed certificate. Active Directory requires that a signed certificate be in the Windows certificate store to enable TLS. You can obtain a signed certificate from the following sources:
      - A third-party certificate authority (CA)
      - Install the Certificate Services role on a system that is joined to the Active Directory domain and configure an enterprise root CA

      Important: Do not install the Certificate Services role on the Active Directory server. Some Active Directory Domain configurations are not

suited to accept an installed Certificate Services role. For example, when there are multiple Active Directory servers in a domain.

b. Add the signed certificate to the Windows certificate store on the Windows Active Directory server.

4. Determine whether the Tivoli Storage Manager server trusts the CA. As the instance owner from the instance directory, issue the following command:

```
gsk8capicmd_64 -cert -list -db cert.kdb -stashed
gsk8capicmd_64 -cert -details -label "My CA" -db cert.kdb -stashed
```

5. If the CA is not trusted, export the CA certificate from the certificate server. You can select either Distinguished Encoding Rules (DER) binary or Base64-encoded ASCII.

   Tip: To determine whether the file is DER binary or ASCII, open the certificate in a text editor. If you can read the characters, then the file is ASCII.
   Ensure that you have the root certificate and that the subject on the certificate matches the CA name. The "Issued by" and "Issued to/subject" for the root certificate must be the same. Export the CA certificate by using one of the following methods:

   - Export the CA certificate from the "Certificates (Local Computer)" Microsoft Management Console (MMC) snap-in.
   - Copy the certificate from `C:\Windows\system32\certsrv\CertEnroll\*.crt` into the server key database. The file is in DER binary format.
   - Download the CA certificate file from the Certificate Services web interface `http://<certificate server hostname>/certsrv/`, if it is enabled through the Certificate Enrollment Web Services.

6. Copy the certificate to the Tivoli Storage Manager server.

7. Add the certificate to the Tivoli Storage Manager server key database. Issue the following command from the Tivoli Storage Manager server as the instance user from the instance directory:

```
gsk8capicmd_64 -cert -add -db "cert.kdb" -stashed
-label "My CA" -format ascii -file myca.cer
```

Tip: The Tivoli Storage Manager server authenticates with the "LDAP simple password authentication" method.

Related tasks:

Setting up TLS

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.

Managing access to the server and clients

An administrator can control access to the server and clients through a number of methods.

Table 68 describes the typical tasks for managing access to the server and clients.

Table 68. Managing access

Task	Details
Allow a new administrator to access the server	1. "Registering administrator IDs" on page 837 2. "Granting authority to administrators" on page 838
Modify authority for registered administrators	"Managing Tivoli Storage Manager administrator IDs"
Give a user authority to access a client remotely	"Managing client access authority levels" on page 441
Give an administrator authority to create a backup set for a client node	"Generating client backup sets on the server" on page 538
Prevent administrators from accessing the server	"Locking and unlocking administrator IDs from the server" on page 842
Prevent new sessions with the server, but allow current sessions to complete	"Disabling or enabling access to the server" on page 465
Prevent clients from accessing the server	"Locking and unlocking client nodes" on page 434
Change whether passwords are required to access IBM Tivoli Storage Manager	"Disabling the default password authentication" on page 855
Change requirements for passwords	• "Modifying the default password expiration period for passwords that are managed by the Tivoli Storage Manager server" on page 850 • "Setting a limit for invalid password attempts" on page 853 • "Setting a minimum length for a password" on page 854
Prevent clients from initiating sessions within a firewall **Tip:** For information on connecting with IBM Tivoli Storage Manager across a firewall, refer to the *Installation Guide*.	"Server-initiated sessions" on page 427

Managing Tivoli Storage Manager administrator IDs

The administrator can register other administrator IDs, grant levels of authority to them, rename or remove them, and lock and unlock them from the server.

A privilege class is a level of authority granted to an administrator. The privilege class determines the administrative tasks that the administrator can complete.

Task	Required Privilege Class
Registering an administrator	System
Granting administrative authority	System

Task	Required Privilege Class
Modifying 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

Related reference:

"Administrative authority and privilege classes"

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 101 on page 836 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.

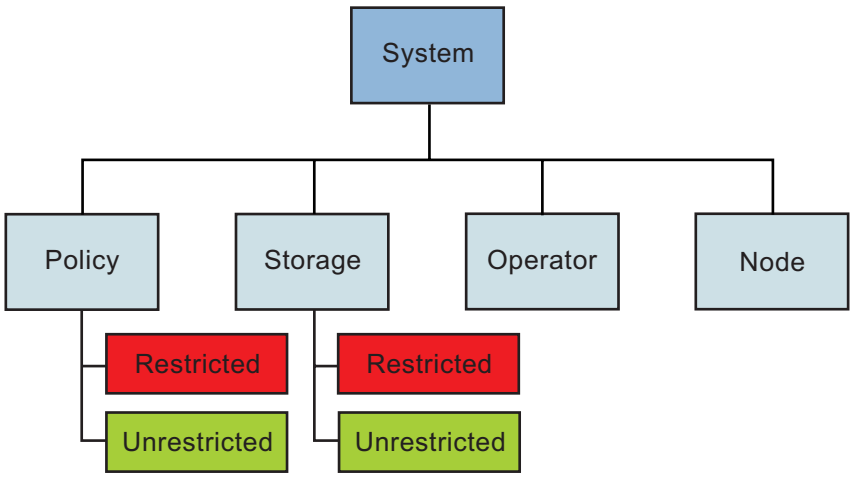


Figure 101. Administrative privilege classes

Table 69 summarizes the privilege classes, and gives examples of how to set privilege classes.

Table 69. Authority and privilege classes

Privilege Class	Capabilities
System `grant authority rocko classes=system`	Perform any administrative task with the server. • 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. • 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. • 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. • Manage IBM Tivoli Storage Manager devices • Manage IBM Tivoli Storage Manager storage

Table 69. 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. • 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 439

"Managing Tivoli Storage Manager administrator IDs" on page 834

Registering administrator IDs

An administrator can register other administrator IDs and set their authentication method by using the **REGISTER ADMIN** command.

About this task

If you want to register an administrator ID and have that ID authenticate with an LDAP directory server, ensure that all LDAP configuration steps are completed. When you use **AUTHENTICATION=LDAP** in a **REGISTER ADMIN** command, the *LDAPURL* setting must be specified in the `dsmserv.opt` file and you must define the **LDAPUSER** and **LDAPPASSWORD** options. If you do not specify an authentication method, the default (LOCAL) is used.

Examples: Registering an administrator

Example 1: Register an administrator with a user ID of DAVEHIL, the password *birds*, and a password expiration period of 120 days.

The authentication method defaults to LOCAL.

```
register admin davehil birds passexp=120 contact='backup team'
```

Example 2: Register an administrator with a user ID of PAULJONES, the password *Ne^Pa$$w0rd*, and the authentication method of LDAP.

```
register admin pauljones Ne^Pa$$w0rd authentication=ldap
```

The expiration period, length of password, and other characteristics of LDAP-authenticated passwords can be configured on the LDAP server. But most IBM Tivoli Storage Manager password-related commands also work for passwords that authenticate to an LDAP server. For more information about password policies, see "Setting the policy for an LDAP-authenticated password" on page 846.

Querying the server for information about administrators

Any administrator can query the server to show administrator information. You can restrict the query to all administrators authorized with a specific privilege class.

Procedure

To query the system for a detailed report on administrator ID DAVEHIL, issue the following example **QUERY ADMIN** command:

```
query admin davehil format=detailed
```

The results are shown in the following 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)
            Authentication: Local
```

Figure 102. A detailed administrator report

Example: Query administrator IDs that authenticate with an LDAP directory server

```
query admin authentication=ldap
```

Only administrator IDs that authenticate to the LDAP directory server are listed in the report.

Granting authority to administrators

You can grant authority by issuing the **GRANT AUTHORITY** command.

Procedure

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.

About this task

For example, JONES has restricted policy privilege for policy domain ENGPOLDOM.

Procedure

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.

About this task

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.

Procedure

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.

Procedure

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
```

Updating information about other administrators

An administrator can reset another administrator's password by issuing the **UPDATE ADMIN** command. The administrator must have system privileges.

About this task

Tip: If you authenticate a password with an LDAP directory server, the letters and characters that comprise the password are case-sensitive.

Procedure

For example, administrator DAVEHIL changes their 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 ID

You can rename an administrator ID if it needs to be identified by a new ID. You can also assign an existing administrator ID to another person by issuing the **RENAME** command. You cannot rename an administrator ID to one that exists on the system.

About this task

For example, if administrator HOLLAND leaves your organization, you can assign administrative privilege classes to another user by completing the following steps:

Procedure

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"
```

Remember: If you rename an administrator ID that authenticates with an LDAP directory server, administrators on other servers that share namespace are not renamed. You must issue a **RENAME** command for each administrator ID. When you rename an administrator ID with **SYNCLDAPDELETE=YES**, the original name is removed from the LDAP directory server and replaced with the new name. Administrators, with the original name, on other servers can no longer authenticate with the LDAP directory server. Those administrators must register with the LDAP directory server or get renamed to the new name.

Results

Note: The administrator SERVER_CONSOLE cannot be renamed.

Related concepts:

"Securing the server console" on page 833

Removing administrator IDs

You can remove administrator IDs from the server so that they no longer have access to administrative functions. You must have system privileges to remove an administrator.

Procedure

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.

Results

Remember: You can remove an administrator ID that authenticates to the IBM Tivoli Storage Manager server, and remove the corresponding administrator namespace on the LDAP directory server. If other Tivoli Storage Manager servers are not using entries on the LDAP directory server, you do not have to keep corresponding LDAP entries on the LDAP server.

Example: Remove administrator ID `pauljones` from the Tivoli Storage Manager server and the LDAP directory server

Administrator ID `pauljones` authenticates with an LDAP directory server and the person who owns that ID is leaving the company. You must find and remove all occurrences of the administrator ID, `pauljones`, on the server.

```
remove admin pauljones syncldapdelete=yes
```

In this example, the **SYNCLDAPDELETE** parameter deletes administrator pauljones from the LDAP server. If administrator pauljones is on other Tivoli Storage Manager servers, you need to run this same command on each one. Do not delete entries on the LDAP directory server if they are still needed by another Tivoli Storage Manager server. Issue SYNCLDAPDELETE=NO if other Tivoli Storage Manager servers use the administrator namespace.

Related concepts:

"Securing the server console" on page 833

Locking and unlocking administrator IDs from the server

You can issue the **LOCK ADMIN** command to lock out other administrator IDs and temporarily prevent them from accessing IBM Tivoli Storage Manager.

About this task

You can also lock or unlock administrator IDs according to the form of authentication that they use. When you specify AUTHENTICATION=LOCAL in the command, all administrator IDs that authenticate with the Tivoli Storage Manager server are affected. When you specify AUTHENTICATION=LDAP in the command, all administrator IDs that authenticate with an LDAP directory server are affected.

For our example, administrator ID MARYSMITH, takes a leave of absence from your business.

Procedure

1. Lock MARYSMITH out by entering the following example command:

   ```
   lock admin marysmith
   ```

2. When MARYSMITH returns, any system administrator can unlock the administrator ID by issuing the following command:

   ```
   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.

Example: Lock all administrator IDs that authenticate locally

```
lock admin * authentication=local
```

Related concepts:

"Securing the server console" on page 833

Restricting a non-root user ID from performing backups as root

You can prohibit client non-root user IDs from backing up data as "root" and limit backups to only the root ID.

To protect data, you can limit backups to just the root user ID when you specify BACKUPINITiation=root with the **REGISTER NODE** or **UPDATE NODE** commands.

Managing passwords and logon procedures

IBM Tivoli Storage Manager requires the server to identify authorized administrator IDs and nodes by using a password. You can authenticate administrator and node passwords with a Lightweight Directory Access Protocol (LDAP) directory server.

Restriction: Backup-archive clients must be at V6.4 or later to authenticate passwords with an LDAP directory server. Storage agents authenticating node IDs with an LDAP directory server must use a secure connection, such as Transport Layer Security (TLS) or a virtual private network.

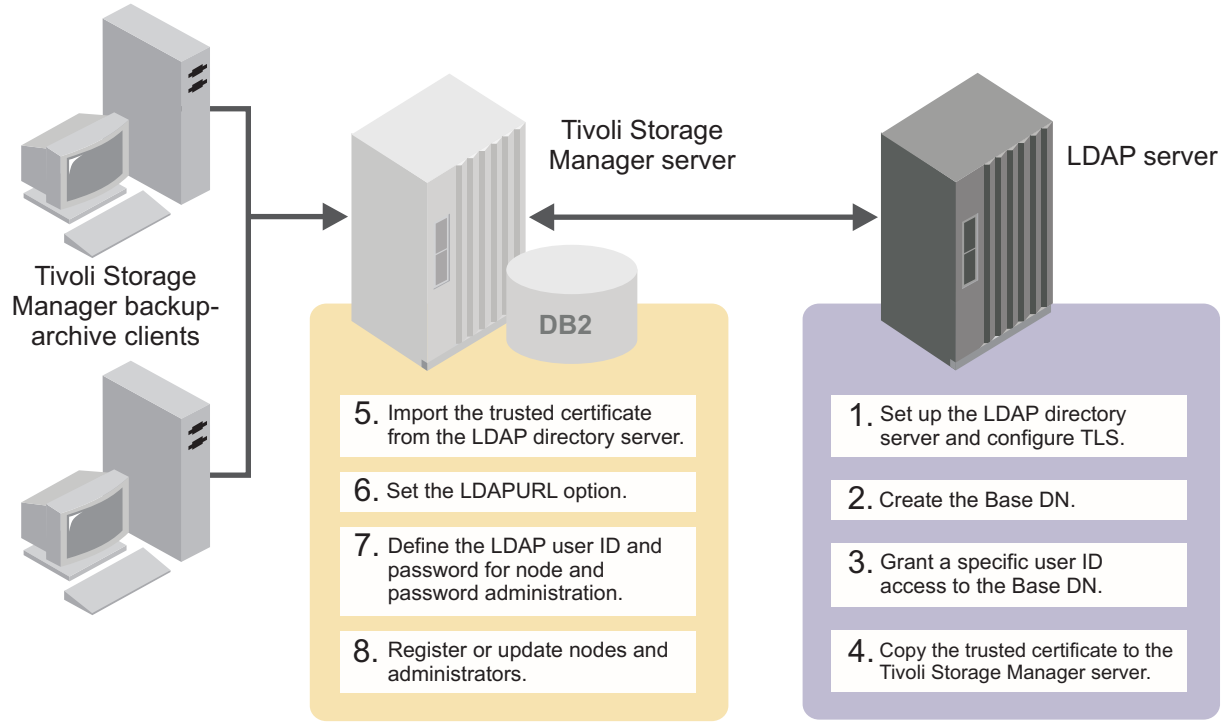


Figure 103. Configuring the server to authenticate passwords with an LDAP directory server

The first step in authenticating passwords with an LDAP directory server is to complete the configuration tasks on the Tivoli Storage Manager server and the LDAP directory server. The following table shows you which steps are accomplished on the two servers:

Table 70. The steps that are required to authenticate passwords with an LDAP directory server, and where the steps are configured

Steps to authenticate passwords with an LDAP directory server	**Where to complete the steps**
1. Set up an LDAP directory server	LDAP server
2. Create the Base DN (distinguished name)	LDAP server
3. Grant access to the Base DN to a specific user ID.	LDAP server
4. Copy the trusted certificate from the LDAP directory server to the Tivoli Storage Manager server	LDAP server

Table 70. The steps that are required to authenticate passwords with an LDAP directory server, and where the steps are configured (continued)

Steps to authenticate passwords with an LDAP directory server	Where to complete the steps
5. Import the trusted certificate from the LDAP directory server to the Tivoli Storage Manager server. If you already have a certificate on the LDAP directory server, you do not have to generate a new certificate. You can use the existing certificate to secure communication between the LDAP directory server and the Tivoli Storage Manager server.	Tivoli Storage Manager server
6. From the Tivoli Storage Manager server, update the `dsmserv.opt` file with the **LDAPURL** option. See the *Administrator's Reference* for more information about the **LDAPURL** option.	Tivoli Storage Manager server
7. Define the user ID that administers node and administrator passwords with the LDAP directory server	Tivoli Storage Manager server
8. Define the password for the user ID that administers node and administrator passwords. See the *Administrator's Reference* for more information about the **SET LDAPPASSWORD** command.	Tivoli Storage Manager server
9. Update or register node and administrator IDs to authenticate with an LDAP directory server. See the *Administrator's Reference* for more information about commands.	Tivoli Storage Manager server

The LDAP directory server interprets letters differently from the Tivoli Storage Manager server. The LDAP directory server distinguishes the case that is used, either uppercase or lowercase. For example, the LDAP directory server can distinguish between *secretword* and *SeCretwOrd*. The Tivoli Storage Manager server interprets all letters for LOCAL passwords as uppercase.

The following terms are used when describing the LDAP directory server environment:

Distinguished name (DN)

A unique name in an LDAP directory. The DN consists of the following information. The information must be ordered in this way.

- The relative distinguished name (RDN)
- The organizational unit (ou)
- The organization (o)
- The country (c)

For example:

```
uid=jackspratt,ou=marketing,o=corp.com,c=us
uid=cbukowski,ou=manufacturing,o=corp.com,c=us
uid=abbynormal,ou=sales,o=corp.com,c=us
```

In this example, the value of the RDN on the first line is of an administrator whose user ID is jackspratt. The organizational unit (marketing), organization (corp.com), and country (us) comprise the DN.

Bind To validate that a certificate is trusted between an LDAP server and another server.

Bind DN
The distinguished name that is used to authenticate with the LDAP server. (This is also the DN of the user ID that is defined in the Tivoli Storage Manager **SET LDAPUSER** command.) For example, if our **SET LDAPUSER** command is:

```
set ldapuser "uid=jackspratt,ou=media,cn=security"
```

then `uid=jackspratt,ou=media,cn=security` is the bind DN for the LDAP directory server.

Bind DN password
The password that is associated with the bind DN.

Configuring a directory server for password authentication

Before you can authenticate passwords with the LDAP server, you must configure the LDAP server to communicate with the Tivoli Storage Manager server.

Before you begin

You must know the user ID that was specified in the **SET LDAPUSER** command. For information about the Tivoli Directory access control lists, go to the Tivoli Directory server information center (http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDS.doc/welcome.htm).

Note: Windows Active Directory users who change passwords when the "Enforce password history" policy is enabled can authenticate with the previous password for one hour. For more information, see the Microsoft site (http://support.microsoft.com/?id=906305).

Procedure

Complete the following steps to set up the LDAP directory server so that it can authenticate passwords:

1. Ensure that you have a directory server installed on the LDAP server. Use one of the following directory servers:
   - IBM Tivoli Directory Server V6.2 or 6.3
   - Windows Active Directory version 2003 or 2008

   Requirement: If you use Tivoli Directory Server V6.2, you must update Global Security Kit (GSKit) to V7.0.4.33 or later. For more information, see SSL errors after upgrading to ITDS 6.3 client (http://www.ibm.com/support/docview.wss?uid=swg21469388).
2. Create the base distinguished name (Base DN) on the LDAP directory server for the Tivoli Storage Manager namespace. The Base DN is the part of the LDAP directory structure from which Tivoli Storage Manager operates, specified in the **LDAPURL** option. For example, `ou=armonk,cn=tsmdata` can be a Base DN. See your LDAP documentation for how to create a Base DN.
3. Edit the access controls on the LDAP directory server and grant access to the Base DN to the user ID, which is specified in the **SET LDAPUSER** command. This ID cannot be a part of the Base DN. You can grant access to the Base DN to

more than one user ID. However, the security of the LDAP server can be easily compromised if you have too many user IDs with full permission over the Base DN.

4. Set up the directory server. See "Configuring SSL or TLS for LDAP directory servers" on page 829.

What to do next

To verify that the LDAP directory server is properly set up, complete the following steps on the Tivoli Storage Manager server:

1. Test the forward- and reverse-DNS lookup of the LDAP directory server.
2. Test the network connection with the LDAP directory server.
3. Use an LDAP utility test to connect to the LDAP server and search without Secure Sockets Layer (SSL)/Transport Layer Security (TLS).
4. Use an LDAP utility test to connect to the LDAP server and search with SSL/TLS.

Related tasks:

"Configuring Tivoli Directory Server for TLS on the iKeyman GUI" on page 830

"Configuring Tivoli Directory Server for TLS on the command line" on page 831

Related reference:

"Configuring Windows Active Directory for TLS/SSL" on page 832

Setting the policy for an LDAP-authenticated password

Passwords that you authenticate with a Lightweight Directory Access Protocol (LDAP) server can be managed by the Tivoli Storage Manager server and the LDAP directory server.

You establish policies for passwords that will be authenticated by each server.

Restriction: You can issue Tivoli Storage Manager server commands to manage your password policies. If you set a password policy on both the LDAP server and Tivoli Storage Manager server, the settings might conflict. The result might be that you are not able to access a node or log on with an administrator ID. For information on the maximum invalid attempts policy, see the table in "Setting a limit for invalid password attempts" on page 853.

In addition to setting a policy for case sensitivity, you can configure the LDAP-authenticated password policy to set the following options:

Password history
The password history is the number of times that you must define a new password before you can reuse a password.

Minimum age
The minimum age is the length of time before you can change the password.

Maximum age
The maximum age is the length of time before you must change the password.

A combination of characters
You can determine the number of special characters, numbers, and alphabetical characters for your passwords. For example, some products set up a password policy to enforce the following rules:

- The password cannot contain the user account name or parts of the user full name that exceed three consecutive characters
- The password must be at least eight characters in length
- The password must contain characters from two of the following four categories:
  - English uppercase characters (A through Z)
  - English lowercase characters (a through z)
  - Base 10 digits (0 through 9)
  - Non-alphabetic characters (for example, !, $, #, %)

Use any of the following characters when you create a password:

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

Important: If you issue a **REGISTER NODE** or **REGISTER ADMIN** command, place special characters within quotation marks. For example:

```
register admin JohnDoe "pa$$w0rd<new>" authentication=ldap
```

The LDAP server that you use determines the complexity that you can have for passwords outside of Tivoli Storage Manager.

Configuring the Tivoli Storage Manager server to authenticate passwords with an LDAP directory server

You can authenticate passwords with an LDAP directory server only after configuring the Tivoli Storage Manager server.

Procedure

Complete the following steps on the Tivoli Storage Manager server to authenticate passwords with an LDAP directory server:

1. Import the key database file from the LDAP directory server. You can use any method to copy the file from the LDAP directory server to the Tivoli Storage Manager server.
2. Open the `dsmserv.opt` file and specify the LDAP directory server with the **LDAPURL** option. Specify the LDAP directory server URL and the base distinguished name (DN) on the **LDAPURL** option. For example:

   ```
   LDAPURL    ldap://server.dallas.gov/cn=project_x
   ```

   The default port is 389. If you want to use a different port number, specify it as part of the **LDAPURL** option. For example, to specify a port of 222:

   ```
   LDAPURL    ldap://server.dallas.gov:222/cn=project_x
   ```
3. Restart the Tivoli Storage Manager server.
4. Issue the **SET LDAPUSER** command to define the ID of the user who can administer Tivoli Storage Manager operations on the LDAP directory server. This user ID must have full administrative authority over the Base DN and be able to add, delete, and modify all Base DN entries. For example:

   ```
   set ldapuser "cn=apastolico,ou=manufacturing,o=dhs,c=us"
   ```

   See the *Administrator's Reference* for more information about the **SET LDAPUSER** command.

5. Issue the **SET LDAPPASSWORD** command to define the password for the user ID that is defined in the **LDAPUSER** option. For example:

```
set ldappassword "boX=T^p$"
```

Registering nodes and administrator IDs to authenticate passwords with an LDAP directory server

After you configure the LDAP directory server and the Tivoli Storage Manager server, you can register nodes and administrator IDs to manage the Tivoli Storage Manager server.

About this task

If the user ID and password are verified to be correct, communication lines are opened and the node or administrator ID can run Tivoli Storage Manager applications.

Procedure

Complete the following step to register either a node or an administrator ID to authenticate with an LDAP directory server:

1. Issue the **REGISTER NODE** or the **REGISTER ADMIN** command from the command-line interface and specify an authentication method. The authentication methods that are available are LDAP or LOCAL.

   LDAP The administrator ID or node authenticates their password with an LDAP directory server.

   LOCAL
   The administrator ID or node authenticates their password with the Tivoli Storage Manager server.

   For example:

   ```
   register admin admin1 "c0m=p1e#Pa$$w0rd?s" authentication=ldap
   register node node1 "n0de^Passw0rd%s" authentication=ldap
   ```

   After you issue the commands, the passwords for administrator ID *admin1* and the node ID *node1* can be authenticated with an LDAP directory server.

   Tip: A node and its password or an administrator ID and its password each occupy one inetOrgPerson object on the LDAP directory server. For information about inetOrgPerson objects, see Definition of the inetOrgPerson LDAP Object Class (http://www.ietf.org/rfc/rfc2798.txt).
   To know which authentication method is in use, issue the **QUERY NODE FORMAT=DETAILED** or **QUERY ADMIN FORMAT=DETAILED** command.
2. Optional: To register all new nodes and administrator IDs with a default authentication method, issue the **SET DEFAULTAUTHENTICATION** command. Any **REGISTER NODE** or **REGISTER ADMIN** commands that are issued after you issue the **SET DEFAULTAUTHENTICATION** command create nodes and administrators with the default authentication method. You can set the authentication methods to LDAP or LOCAL.

   For information about the **SET DEFAULTAUTHENTICATION** command, see the *Administrator's Reference*.

Updating nodes and administrator IDs to authenticate passwords with a directory server

Node and administrator ID passwords can be authenticated with an LDAP directory server or with Tivoli Storage Manager. You can use either authentication method.

About this task

When you authenticate nodes and administrator IDs with an LDAP directory server, you ensure more protection for your passwords. Communication lines between the LDAP directory server and Tivoli Storage Manager are protected with Transport Layer Security (TLS).

You can change a password authentication method after you configure the LDAP directory server and the Tivoli Storage Manager server. However, you cannot update the authentication method for your own user ID unless you have system authority. If necessary, another administrator must change the authentication method.

Procedure

To change a node or administrator password authentication method, complete the following steps:

1. From the Tivoli Storage Manager server, issue the **UPDATE NODE** or **UPDATE ADMIN** command. You do not have to include a password in the command. If you do not include a password, you are asked for a new password at the next logon. You must then enter a password with a combination of uppercase, lowercase, and numbers or valid characters. For example:

   ```
   update admin harry authentication=ldap
   ```

   The following example **UPDATE NODE** command has a password that is made up of characters that are supported by the Tivoli Storage Manager server:

   ```
   update node node1 n0de^87^n0de authentication=ldap
   ```

   Tip: A shared LDAP server might have a password that is on the LDAP directory server. In that case, the user is not prompted to enter a new password.
2. Optional: Issue the **QUERY NODE FORMAT=DETAILED** or the **QUERY ADMIN FORMAT=DETAILED** command to view the results. If you must change the authentication method for several nodes or administrator IDs, you can use a wildcard character (*). For example,

   ```
   update node * authentication=ldap
   ```

   In the preceding example, the authentication method for all nodes is changed to “LDAP pending.”

What to do next

All nodes and administrator IDs require new passwords after you run the UPDATE command. Before the node and administrative IDs are given a password, they are in the LDAP pending state. The node and administrator IDs are updated to use LDAP authentication, but you must first give them a password.

Determining which nodes and administrator IDs are configured to authenticate with an LDAP server

You might have a mixture of nodes or administrator IDs that can and cannot authenticate with an LDAP directory server. After you update nodes and administrator IDs to authenticate with an LDAP directory server, you can see which nodes or administrator IDs are using them.

Tivoli Storage Manager backup-archive clients must be at version 6.4 or later to authenticate passwords with an LDAP directory server. If all of the backup-archive clients are not upgraded to version 6.4 or later, you cannot authenticate all nodes with an LDAP directory server. To determine which nodes or administrator IDs are configured to authenticate with an LDAP directory server, you can use the **QUERY NODE** or **QUERY ADMIN** command.

Example: Determine which nodes are configured to authenticate with an LDAP server

Find the nodes that are authenticated with the LDAP directory server:

```
query node authentication=ldap
```

Example: Determine which administrator IDs are configured to authenticate with the Tivoli Storage Manager server

Find the administrator IDs that do not authenticate their passwords with an LDAP directory server:

```
query admin authentication=local
```

Example: Determine the password authentication method for one node

You can query individual nodes or administrator IDs to determine whether they authenticate with an LDAP directory server. To determine the password authentication method for node tivnode_12 issue the following command:

```
query node tivnode_12 format=detailed
```

Modifying the default password expiration period for passwords that are managed by the Tivoli Storage Manager server

By default, the Tivoli Storage Manager server sets a password expiration of 90 days. The expiration period begins when an administrator ID or client node is first registered to the server. If a user's password is not changed within this period, the user must change the password the next time they access the server.

About this task

Issue the **SET PASSEXP** command to set the password expiration period for selected administrator IDs or client nodes. You must specify the administrator ID or node name with the **ADMIN** or **NODE** parameter in the **SET PASSEXP** command. If you set the expiration period only for selected users, the expiration period can be 0 - 9999 days. A value of 0 means that user's password never expires.

Restriction: The **SET PASSEXP** command does not affect administrator IDs and nodes if their passwords are authenticated with an LDAP directory server.

The password expiration period for passwords that are authenticated with an LDAP directory server must be defined on the LDAP server.

Procedure

Issue the following command to set the expiration period of client node node_tsm12 to 120 days:

```
set passexp 120 node=node_tsm12
```

After you explicitly set password expiration for an administrator ID, 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.

Related concepts:

"Setting the policy for an LDAP-authenticated password" on page 846

Scenarios for authenticating passwords

The following example scenarios are for authenticating passwords with an LDAP directory server.

Scenario 1: Register one node to authenticate its password with an LDAP directory server

The Tivoli Storage Manager server administrator has a new node that must authenticate its password with an LDAP directory server. The first action is to create the "cn=tsmdata" entry and Base DN on the LDAP directory server. The server administrator can then set up the **LDAPURL** option that is based on the Base DN. Here is an example entry for the **LDAPURL** option:

dsmserv.opt

```
LDAPURL ldaps://mongo.storage.tucson.ibm.com:389/cn=tsmdata
```

After you set the **LDAPURL** option, restart the server. Complete the following steps to configure the server:

1. Issue the `query option ldapurl` command to validate that you entered all of the values correctly.
2. Issue the `set ldapuser uid=tsmserver,ou=Users,cn=aixdata` command to configure the **LDAPUSER**.
3. Issue the `SET LDAPPASSWORD adsm4Data` command to define the password.
4. For this scenario, the node that must be added is NODE1. Issue the following command:

   ```
   register node c0mplexPassw0rd NODE1 authentication=ldap
   ```

   command.

Scenario 2: Update a node to authenticate with an LDAP directory server

A single node (UPDNODE1) that currently authenticates with the Tivoli Storage Manager server is now required to authenticate with an LDAP directory server. For UPDNODE1, use the **AUTHENTICATION** parameter in the **UPDATE NODE** command. For example:

```
update node updnode1 newC0mplexPW$ authentication=ldap
```

If you do not want to specify a password now, you can run the command without it. At the next logon, either you enter a new password or the password is generated automatically. If the `passwordaccess=generate` option is set, a new password is generated for you.

Scenario 3: Update all nodes to authenticate their password with an LDAP directory server

If you want to update all your nodes to authenticate with an LDAP directory server, you can use a wildcard. Issue the following command to have all the nodes authenticate with an LDAP directory server:

```
update node * authentication=ldap
```

Scenario 4: Determine the authentication method of a node

If you have nodes that authenticate with the Tivoli Storage Manager server and nodes that authenticate with an LDAP directory server, you can determine where nodes are authenticating. Issue the following command to determine which nodes authenticate with an LDAP directory server:

```
query node authentication=ldap
```

Issue the following command to determine which nodes authenticate with the Tivoli Storage Manager server:

```
query node authentication=local
```

Scenario 5: Locking nodes that authenticate with the Tivoli Storage Manager server

You can issue a **LOCK NODE** command to lock all nodes that authenticate with the Tivoli Storage Manager server. These nodes might be rarely used, and you might not know by which password authentication method they are supposed to be managed. When you lock the nodes, the node owners must consult with you. At that point, you can find out whether they want to use the LDAP directory server or stay with the Tivoli Storage Manager server. You can issue the **LOCK NODE** or **UNLOCK NODE** commands with a wildcard to lock or unlock all nodes in that group. To lock all nodes that authenticate with the Tivoli Storage Manager server, issue the following command:

```
lock node * authentication=local
```

Scenario 6: Setting up a default authentication method

After you configure everything, you can design it so that every new node and administrator authenticate with an LDAP directory server. After you issue the **SET DEFAULTAUTH** command, you do not have to designate the authentication method for any **REGISTER NODE** or **REGISTER ADMIN** commands. Issue the following command to set the default authentication method to LDAP:

```
set defaultauth=ldap
```

Any **REGISTER NODE** or **REGISTER ADMIN** command that is issued after this **SET DEFAULTAUTH** command inherits the authentication method (LDAP). If you want to register a node that authenticates with the Tivoli Storage Manager server, include `AUTHENTICATION=LOCAL` in the **REGISTER NODE** command.

Setting a limit for invalid password attempts

By default, Tivoli Storage Manager does not check the number of times a user attempts to logon 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.

About this task

If you authenticate passwords with an LDAP directory server, determine on which server the invalid attempts are to be configured. Invalid password attempts can be configured on an LDAP directory server, outside of the Tivoli Storage Manager server. But the consequence of setting the number of invalid attempts on the LDAP directory server might pose some problems. For example, when the **REGISTER NODE** command is issued, the default behavior is to name the node administrator the same name as the node. The LDAP server does not recognize the difference between the node "NODE_Q" and the administrator "NODE_Q". The node and the administrator can authenticate to the LDAP server if they have the same password. If the node and administrator have different passwords, the authentication fails for either the node or administrator. If the node or the administrator fail to logon consistently, their IDs are locked. You can avoid this situation by issuing the **REGISTER NODE** command with **USERID=***userid* or **USERID=NONE**.

Consider the maximum invalid attempts policy:

Table 71. The consequences of setting the maximum invalid attempts on both the Tivoli Storage Manager server and the LDAP server for a node and admin that share a name

Authentication method	Tivoli Storage Manager password setting (A maximum of 3 invalid attempts)	LDAP directory server setting (A maximum of 3 invalid attempts)
With an LDAP directory server (LDAP)	The password provided during authentication is attempted for both the node and administrator that share a name. The policy does not affect the node or administrators that are authenticating to the Tivoli Storage Manager server, even when their passwords are different. The Tivoli Storage Manager server recognizes that the node and administrator share a name and authenticates both simultaneously. If the passwords are not the same and the correct node password is provided on 3 attempts, the administrator's incorrect password is ignored. No invalid login attempts are recorded.	The password provided during authentication is attempted for both the node and administrator that share a name. If the node and administrator have different passwords, the authentication on the LDAP server fails for either the node or administrator. If the correct node password is provided on 3 consecutive attempts, the administrator authentication fails on those same 3 attempts. The administrator ID gets locked out after the third attempt.
With the Tivoli Storage Manager server (LOCAL)	Has 3 attempts to get the password right	Not applicable.

On the Tivoli Storage Manager server, issue the **SET INVALIDPWLIMIT** command to limit the invalid password attempts for the Tivoli Storage Manager namespace.

Procedure

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 lower the limit, some clients might fail verification during the next logon attempt.

After a client node is locked, only an administrator with storage authority can unlock the node.

An administrator can also force a client to change their password on the next logon by specifying the **FORCEPWRESET=YES** parameter on the **UPDATE NODE** or **UPDATE ADMIN** command. For more information, see the *Administrator's Reference*.

Related tasks:

"Locking and unlocking client nodes" on page 434

"Locking and unlocking administrator IDs from the server" on page 842

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 Tivoli Storage Manager passwords.

About this task

This feature affects all node and administrator passwords, whether the password authenticates with the Tivoli Storage Manager server or the LDAP directory server.

You can configure the parameters of your LDAP-authenticated passwords on the LDAP directory server. The settings that you configure might conflict with the Tivoli Storage Manager server settings.

Procedure

To set the minimum password length to eight characters, issue the following example command:

```
set minpwlength 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 to "on." With password authentication set to on, all users must enter a password when accessing the server.

About this task

You can only disable password authentication for passwords that authenticate with the Tivoli Storage Manager server (LOCAL).

Procedure

To allow administrators and client nodes to access the Tivoli Storage Manager server without entering a password, issue the following command:

```
set authentication off
```

Remember: Setting password authentication to "off" reduces data security.

Chapter 28. Protecting and recovering the server infrastructure and client data

The Tivoli Storage Manager infrastructure consists of the database and the setup files that are required to recover the database and client data. The setup files include, for example, the active log and the archive log. Client data includes data that is backed up, archived, and migrated to primary storage pools.

About this task

Database backups, infrastructure setup files, and copies of client data can be stored offsite, as shown in Figure 104.

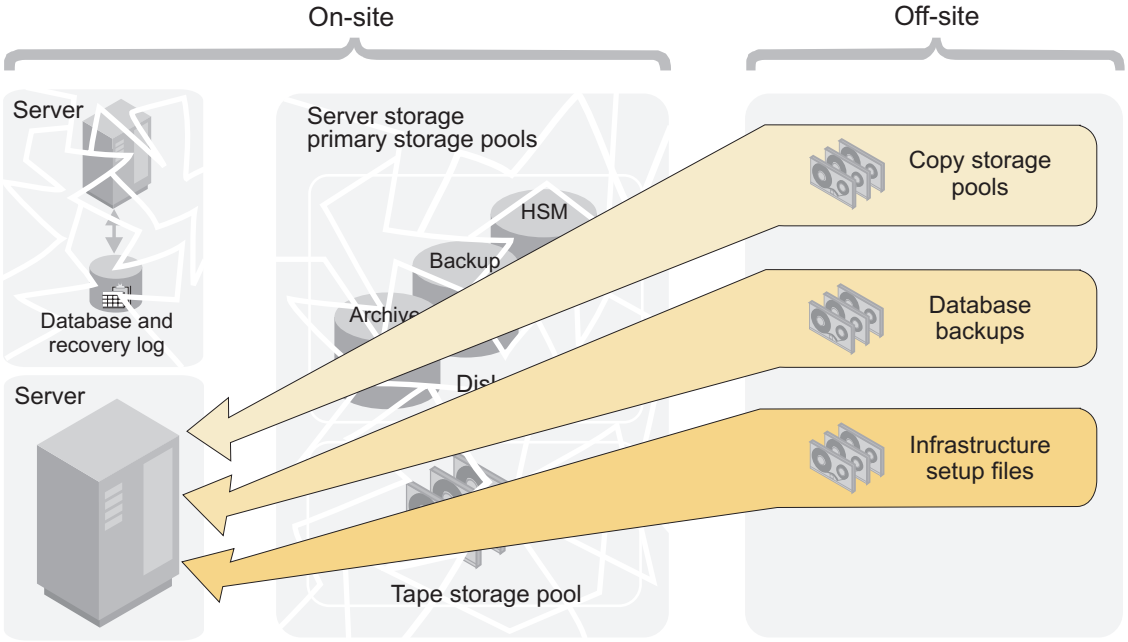


Figure 104. Recovery from a disaster

DRM: The disaster recovery manager (DRM) can automate some disaster recovery tasks. A note like this one identifies those tasks.

Related tasks:

"Storage pool hierarchies" on page 256

Related information:

Configuring clustered environments

© Copyright IBM Corp. 1993, 2013

Protecting the database and infrastructure setup files

To restore a damaged or lost database you must have a database backup. You must also have copies of the files that are required to recover the database and client data. Database backup media and setup files can be stored offsite for protection.

Restriction: Database mirroring is not possible with Tivoli Storage Manager. However, you can mirror the database by using operating system or file system capabilities. You can also use device redundancy such as RAID capabilities in the storage that is used for the server database.

DRM: To store database backup media and setup files offsite, you can use disaster recovery manager.

Related tasks:

Chapter 30, "Disaster recovery manager," on page 979

Backing up the server database

When the Tivoli Storage Manager server is first started, a full backup starts automatically. After the initial full backup, the database is backed up automatically by the database manager. To ensure that the database can be restored to an acceptable point-in-time, you can schedule daily backups or backup the database manually.

About this task

Automatic backups by the database manager are based on the following values that are set by Tivoli Storage Manager:

- The active log space that was used since the last backup, which triggers a full database backup
- The active log utilization ratio, which triggers an incremental database backup

You can back up the database to tape, FILE, or to remote virtual volumes.

To set up regular database backups, complete the following tasks:

- "Step 1: Defining device classes for database backups" on page 859
- "Step 2: Specifying virtual address space for database manager processes" on page 859
- "Configuring concurrent multistreaming" on page 860
- "Step 4: Running database backups" on page 863

Related concepts:

"Active log" on page 637

"Database: Overview" on page 634

Related tasks:

"Using virtual volumes to store data on another server" on page 722

"Increasing the size of the active log" on page 668

Step 1: Defining device classes for database backups

You can use existing device classes for database backups or you can define new ones. You can also specify device classes for incremental backups that are different from the device classes for full backups. For example, you might want to write full backups to tape and incremental backups to disk.

About this task

Reserve the device class that you want to use for backups so that the server does not attempt to back up the database if a device is not available. If a database backup shares a device class with a lower priority operation, such as reclamation, and all the devices are in use, the lower priority operation is automatically canceled. The canceled operation frees a device for the database backup.

Restriction: Tivoli Storage Manager does not support database backup (loading and unloading) to a Centera device.

Procedure

To specify the device class to be used for database backups, issue the **SET DBRECOVERY** command. For example, to specify a device class named DBBACK, issue the following command:

```
set dbrecovery dbback
```

Tips:

- When you issue the **SET DBRECOVERY** command, you can also specify the number of number of concurrent data streams to use for the backup. Use the **NUMSTREAMS** parameter.
- To change the device class, reissue the **SET DBRECOVERY** command.
- 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.
- Device class definitions are saved in the device configuration files.

Related concepts:

"Configuring concurrent multistreaming" on page 860

Related tasks:

"Protecting the device configuration file" on page 867

Step 2: Specifying virtual address space for database manager processes

Specify the percentage of the virtual address space that is dedicated to database manager processes.

About this task

By default, the percentage of the virtual address space that is dedicated to all database manager processes is set to 70 - 80 percent of system random-access memory.

Procedure

To change this setting, specify the DBMEMPERCENT server option. Ensure that the value that you specify provides adequate memory for applications other than the

Tivoli Storage Manager are running on the system.

Configuring concurrent multistreaming

Multiple, concurrent data streams reduce the time required to back up or restore the database. You can specify the number of data streams that the IBM Tivoli Storage Manager server uses for backup and restore operations.

For example, if you assign four drives to database backup processing, Tivoli Storage Manager attempts to write to all four drives concurrently. For restore operations, the server uses the information that is in the volume history file to determine the number of data streams that were used during the backup operation. The server attempts to use the same number of data streams during the restore operation. For example, if the backup operation used four data streams, the server attempts the restore operation using four data streams.

The following table summarizes server actions for different drive-availability scenarios. An *available drive* is a drive that is online and that is ready for use by the server.

Operation	If the number of available drives exceeds the specified number of streams, the server uses	If the number of available drives equals the specified number of streams, the server uses	If the number of available drives is less than the specified number of streams, the server uses
Backup	The number of drives that is equal to the specified number of streams.	The number of drives that is equal to the specified number of streams.	All available drives.
Restore	The number of drives that is equal to the number of streams that were used in the backup operation. A restore process never uses more drives than the number of streams that were used to back up the database.	The number of drives that is equal to the number of streams that were used in the backup operation.	All available drives. At least one drive is required for restore processing

Suppose that you specify four data streams for database backup operations. To indicate the maximum number of volumes that can be simultaneously mounted, you specify 4 as the value of the **MOUNTLIMIT** parameter in the device class definition. If only three drives are available at the time of the backup operation, the operation runs using three drives. A message is issued that indicates that fewer drives are being used for the backup operation than the number requested. If all four drives for the device class are online, but one drive is in use by another operation, the backup operation has a higher priority and preempts use of the drive. If you specify four data streams, but the value of the **MOUNTLIMIT** parameter is 2, only two streams are used.

Important: Although multiple, concurrent data streams can reduce the time that is required for a backup operation, the amount of time that you can save depends on the size of the database. In general, the benefit of using multiple, concurrent data streams for database backup and restore operations is limited if the database is less than 100 GB.

Another potential disadvantage is that more volumes are required for multistream processing than for single-stream processing. For example, if the backup of an 850 GB database requires a single linear tape open (LTO) volume, switching to four data streams requires four volumes. Furthermore, those volumes might be partially

filled, especially if you use high-capacity volumes and device compression. For example, if the backup of an 850 GB database consumes only 30% of a tape volume after compression, then four-stream processing will result in even larger amounts of wasted space.

Planning for concurrent multistreaming during database backup and restore operations:

The decision to use multiple, concurrent data streams for database backup and restore operations depends on the size of the database, the cost of media, and performance impacts.

Before you begin

When deciding whether to use data streaming, consider the following issues to determine whether the benefits of concurrent data streaming are sufficient. If the disadvantages of multiple, concurrent data streaming exceed the benefits, continue to use single-stream processing.

- What is the size of your database? In general, the amount of time that you save by using multiple, concurrent data streams decreases as the size of the database decreases because of the extra time caused by additional tape mounts. If your database is less than 100 GB, the amount of time that you save might be relatively small.

  In many environments with databases larger than 100 GB, two database-backup streams can provide superior performance. However, depending on your environment, additional streams might not provide enough I/O throughput relative to the size of your database, the devices that you use, and the I/O capability of your environment. Consider using three or four database-backup streams only for environments in which the following conditions apply:

  – The Tivoli Storage Manager database is located on very high-performing disk subsystems.
  – The database is spread across several different RAID arrays that use multiple database directories.
- How many drives are available for the device class to be used for database backup?
- Will server operations other than database backup operations compete for drives?
- If drives are preempted by a database backup operation, what will be the effect on server operations?
- What is the cost of the tape volumes that you use for database backup operations? For example, suppose that the backup of an 850 GB database requires a single high-capacity LTO volume. If you specify four streams, the same backup operation requires four volumes.

Specifying multiple data streams for database backup operations:

You can specify multiple data streams for automatic or manual database-backup operations. For database restore operations, the server attempts to use the same number of data streams that you specified for the backup operation.

Procedure

To specify multiple data streams, complete one of the following steps:

- For automatic database-backup operations, issue the **SET DBRECOVERY** command and specify a value for the **NUMSTREAMS** parameter. For example, if you have a device class DBBACK, issue the following command to specify two data streams:

  ```
  set dbrecovery dbback numstreams=2
  ```

- For manual database-backup operations, issue the **BACKUP DB** command and specify a value for the **NUMSTREAMS** parameter. The value of the **NUMSTREAMS** parameter that you specify with the **BACKUP DB** command overrides the value for the **NUMSTREAMS** parameter that you specify with the **SET DBRECOVERY** command.

  For example, if you have a device class DBBACK, issue the following command to specify three data streams:

  ```
  backup db dbback numstreams=3
  ```

Example

Tips:

- To change the number of data streams for automatic database backup operations, reissue the **SET DBRECOVERY** command and specify a different value for the **NUMSTREAMS** parameter. For example, reissue the **SET DBRECOVERY** command if you add additional drives to the target library or if drives are not available because of maintenance or device failure. The new value specified by the **NUMSTREAMS** parameter is used for the next backup operation.
- To display the number of data streams that are to be used for a database backup operation, issue the **QUERY DB** command.
- During a database backup operation, the number of sessions that is displayed in the output of the **QUERY SESSION** command or the **SELECT** command is equal to or less than the number of specified data streams. For example, if you specified four data streams, but only three drives are online, 3 sessions are displayed in the output. If you issue the **QUERY DRIVE** command, the number of drives in use is also 3.
- If you reduce the number of data streams after a database backup operation, this information will not be available to the server when the database is restored. To specify fewer data streams for the restore operation, take one or both of the following actions in the device configuration file:
  - Reduce the number of online and usable drive definitions by removing **DEFINE DRIVE** commands.
  - Update the value of the **MOUNTLIMIT** parameter of the **DEFINE DEVCLASS** command.
  - During database backup operations, stop other Tivoli Storage Manager database activities. Other database activities compete for database I/O and affect throughput during database backup operations that use multiple streams.

Step 4: Running database backups

Database backups can be full, incremental, or snapshot. An incremental backup includes all the changes since the last full backup. You can schedule the backups to occur automatically, or you can back up the database manually. A snapshot database backup is a full database backup that does not interrupt the current full and incremental backup series.

About this task

Ensure that you can recover the database to its most current state or to a specific point-in-time by making both full and incremental database backups:

- To restore the database to its most current state, you need the last full backup, the last incremental backup after that full backup, and the active and archive log files.

  Restriction: You cannot use snapshot database backups to restore a database to its most current state.
- To restore a database to a point-in-time, you need the last full backup before the point-in-time and the last incremental backup after the last full backup before the point-in-time.

  You can also use snapshot database backups to restore a database to a point-in-time.

Tivoli Storage Manager can make full and incremental database backups to tape while the server is running and available to clients. However, when deciding what 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 because you must load only one set of volumes to restore the entire database.
- Full backups are required for the first backup and after extending the database size.
- Only full backups prune archive log space in the archive log directory. If the available active and archive log space gets low, full database backups occur automatically. To help prevent space problems, schedule regular full backups frequently.

Tips: To protect the database, use the following guidelines:

- Store the database backup media an off-site location for protection.
- 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 issue the **MOVE DATA** command while you back up the database.

Backing up the database manually:

To back up the database manually, issue the **BACKUP DB** command from an administrative client or from the server console.

About this task

For a full database backup, specify **TYPE=FULL**. For an incremental database backup, specify **TYPE=INCREMENTAL**. For example, to run a full database backup using a device class LTOTAPE, three volumes, and three concurrent data streams, issue the following command:

```
backup db devclass=ltotape type=full volumenames=vol1,vol2,vol3
   numstreams=3
```

Scheduling database backups:

Database backups require devices, media, and time. Consider scheduling backups at specific times of the day and after major storage operations.

About this task

Consider scheduling database backups after the following types of operations:

- Regular client backup or archive operations
- Storage pool migration and reclamation
- Storage pool backups
- **MOVE DATA** or **DELETE VOLUME** command operations

For example, you might back up your storage pools daily and then immediately back up the database.

Procedure

To schedule database backups, use the **DEFINE SCHEDULE** command. For a full database backup, specify `TYPE=FULL`. For an incremental database backup, specify `TYPE=INCREMENTAL`. 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
```

Running snapshot database backups:

A snapshot database backup is a full database backup that does not interrupt the full and incremental backup series. Consider using snapshot database backups in addition to full and incremental backups.

Procedure

To make a snapshot database backup, issue the **BACKUP DB** command. For example, to make a snapshot database backup to the TAPECLASS device class, enter the following command:

```
backup db type=dbsnapshot devclass=tapeclass
```

New volume history entries are created for the snapshot database volumes.

Restriction: To prevent the accidental loss of what might only way to recover the server, you cannot delete the most current snapshot database using the **DELETE VOLHISTORY** command.

Related concepts:

"Volume history file and volume reuse" on page 79

Related tasks:

"Protecting the volume history file" on page 866

Protecting infrastructure setup files

Infrastructure setup files are prerequisites for recovering the Tivoli Storage Manager database and client data. In most cases, these files cannot be recreated, so you must ensure that copies are up-to-date and easily accessible.

Protecting the active, archive, and archive failover logs

The active log, which records transactions that are in progress on the server, is necessary to restore the database to its most current state. The archive log contains copies of closed log files that were in the active log. The archive log is not needed for normal processing, but it is typically needed for recovery of the database.

About this task

For protection against database and log media failures, place the active log and the archive log in different file systems. In addition, mirror both logs. Mirroring simultaneously writes data to two independent disks. For example, suppose that a sudden power outage causes a partial page write. The active log is corrupted and is not 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.

To protect the active log, the archive log, and the archive failover log, take the following steps:

Procedure

- To specify the active log mirror, use the **MIRRORLOGDIRECTORY** parameter on the **DSMSERV FORMAT** command. Mirror the active log in a file system that exists on a different disk drive than the primary active log.

  Tip: To specify the active log mirror, you can also use the `MIRRORLOGDIRECTORY` server option.
- You cannot mirror the archive log through Tivoli Storage Manager. However, you can mirror the archive log by using operating system or hardware mirroring, such as that provided by RAID 5. Mirror the archive log directory using file system or disk-drive subsystem-level facilities. Ensure that the mirrored log is in a file system on a different physical hardware.
- The archive failover log, also called a secondary archive log, is the directory that the server uses to store archive log files if the archive log directory is full. Specifying an archive failover log directory can prevent problems that occur if the archive log runs out of space. Use of an archive failover log is optional, but it provides an extra level of protection.

  Store the archive log and the archive failover log on different physical drives. To mirror the archive failover log, use operating system or hardware mirroring.

What to do next

Tips:

- Consider mirroring the active log and the archive log if retention protection is enabled. If a database restore is needed, you can restore the database to the current point in time with no data loss.
- You can dynamically start or stop mirroring while Tivoli Storage Manager is running.
- Despite its benefits, mirroring does not protect against a disaster or a hardware failure that affects multiple drives or causes the loss of the entire system. In

addition, mirroring doubles the amount of disk space that is required for logs. Mirroring also results in decreased performance.

Related concepts:

"Active log" on page 637

"Archive log" on page 638

"Archive failover log" on page 639

Protecting the volume history file

To restore the database, the server needs the information that is in a volume history file. You can specify duplicate volume history files. When the server updates volume information in the database, it also updates each file.

About this task

The following volume history is stored in the Tivoli Storage Manager database and updated in the volume history files:

- Sequential-access storage-pool volumes that were added, reused through reclamation or move data operations, or deleted during delete volume or reclamation operations
- Full and incremental database-backup volumes
- Export volumes for administrator, node, policy, and server data
- Snapshot database-backup volumes
- Backup set volumes

Procedure

To specify the file path and name for a volume history file, use the `VOLUMEHISTORY` server option. To specify more than one path and name, use multiple `VOLUMEHISTORY` entries. Tivoli Storage Manager stores duplicate volume histories in all the files that are specified with `VOLUMEHISTORY` options. To find the required volume-history information during a database restore operation, the server tries to open volume history files in the order in which the `VOLUMEHISTORY` entries occur in the server options file. If the server cannot read a file, the server tries to open the next volume history file.

Ensure that volume history is protected by taking one or more 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.

Tip: To manually update the volume history file, you can use the **BACKUP VOLHISTORY** command. Ensure that updates are complete by following these guidelines:

- If you must halt the server, wait a few minutes after issuing the **BACKUP VOLHISTORY** command.
- Specify multiple `VOLUMEHISTORY` options in the server options file.
- Review the volume history files to verify that the files were updated.

What to do next

DRM: DRM saves a copy of the volume history file in its disaster recovery plan file.

Related tasks:

"Deleting information about volume history" on page 606

Protecting the device configuration file

The device configuration file contains information that is required to read backup data and restore the database. You can specify duplicate device configuration files. When the server updates device configuration information in the database, it also updates each file. A device configuration file cannot be recreated.

About this task

The following device configuration information is stored in the Tivoli Storage Manager database and updated in the device configuration files:

- Devices class definitions
- Library definitions
- Drive definitions
- Path definitions
- Server definitions
- The database manager backup node ID

The device information must match the devices configured on the system where the restore operation can be performed. You might have to edit those commands in an existing file so that they match.

Procedure

To specify the file path and name for a device configuration file, use the `DEVCONFIG` server option. To specify more than one path and name, use multiple `DEVCONIG` entries. Tivoli Storage Manager stores duplicate device configuration information in all the files that are specified with `DEVCONFIG` options.
To find the required device-configuration information during a database restore operation, the server tries to open device configuration files in the order in which the `DEVCONFIG` entries occur in the server options file. If the server cannot read a file, the server tries to open the next device configuration file.
To ensure the availability of device configuration information, take one or more of the following steps:

- Store at least one copy of the device configuration 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 volume history file.
- Store a remote copy of the file, for example, on an NFS-mounted file system.

Tips:

- To manually update the device configuration file, use the **`BACKUP DEVCONFIG`** command. Ensure that updates are complete by following these guidelines:
  - If you must halt the server, wait a few minutes after issuing the **`BACKUP DEVCONIG`** command.
  - Specify multiple `DEVCONFIG` options in the server options file.

– Review the device configuration files to verify that the files were updated.
– If you are using automated tape libraries, volume location information is 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.

What to do next

DRM: DRM automatically saves a copy of the device configuration file in its disaster recovery plan file.

Related tasks:

"Updating the device configuration file" on page 894

Protecting the server options file

To restore the database, you need a copy of the server options file. The server options file includes the file paths of the active log, the archive log, the active log mirror, and the archive failover log. This information is required to restore the database.

About this task

To ensure the availability of server options file, take one or more of the following steps:

- Store at least one copy of the server options 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 automatically saves a copy of the server options file in its disaster recovery plan file.

Protecting information about the database and recovery logs

To restore the database, you need detailed information about the database and recovery log. The *recovery log* includes the active log, the active log mirror, the archive log, and the archive failover log. The recovery log contains records of changes to the database.

About this task

You can determine the following information from the recovery log:

- The directory where the recovery log is located
- The amount of disk space required

If you lose the recovery log, you lose the changes that were made since the last database backup.

Procedure

Complete both of the following steps:

- To obtain detailed information about the database, issue the following commands:

```
query db format=detailed
query dbspace
```

- To obtain information about the recovery log, issue the following command:

```
query log format=detailed
```

What to do next

DRM: DRM helps you save database and recovery log information.

Protecting the Secure Sockets Layer digital certificate file

As part of the process of setting up IBM Tivoli Storage Manager to use Secure Sockets Layer (SSL) for client-server authentication, a digital certificate file, `cert.kdb`, is created.

The `cert.kdb` 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 Global Security Kit (GSKit) requires a separate file in a certain format. The `cert256.arm` file is generated by the V6.3 server for distribution to the V6.3 clients.

Keep backup copies of the `cert.kdb` and `cert256.arm` file in a secure location. If both of the original files and any copies are lost or corrupted, you can generate a new 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 operation. Ensure that you back up the server database frequently to prevent data loss.

For more information about encryption keys, see *IBM Tivoli Storage Manager Using the Application Programming Interface*.

Related tasks:

"Troubleshooting the certificate key database" on page 827

Protecting the disaster recovery plan

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.

You can use server-to-server communications to store copies of the recovery plan on a remote target server, in addition to traditional disk-based files. Storing recovery plan files on a target server provides the following advantages:

- A central repository for recovery plan files
- Automatic expiration of plan files
- Query capabilities for displaying information about plan files and their contents

- Fast retrieval of a recovery plan file if a disaster occurs

You can also store the recovery plan locally, on CD, or in print.

DRM: DRM can query the server and generate a detailed recovery plan for your installation.

Related tasks:

"Storing the disaster recovery plan locally" on page 991

"Storing the disaster recovery plan on a target server" on page 991

Related reference:

"The disaster recovery plan file" on page 1018

Protecting client data

If storage pools and volumes are not backed up, client data can be permanently lost.

Protecting the data that is in primary storage pools

To protect client data, back up your primary storage pools to copy storage pools. For fast client restores, copy active client backup data that is in primary storage pools to active-data pools.

About this task

A typical Tivoli Storage Manager configuration includes a primary disk pool and primary tape pool for data backup. Copy storage pools contain active and inactive versions of data that is backed up from primary storage pools. Figure 105 shows a configuration with an onsite FILE-type active-data pool and an offsite copy storage pool.

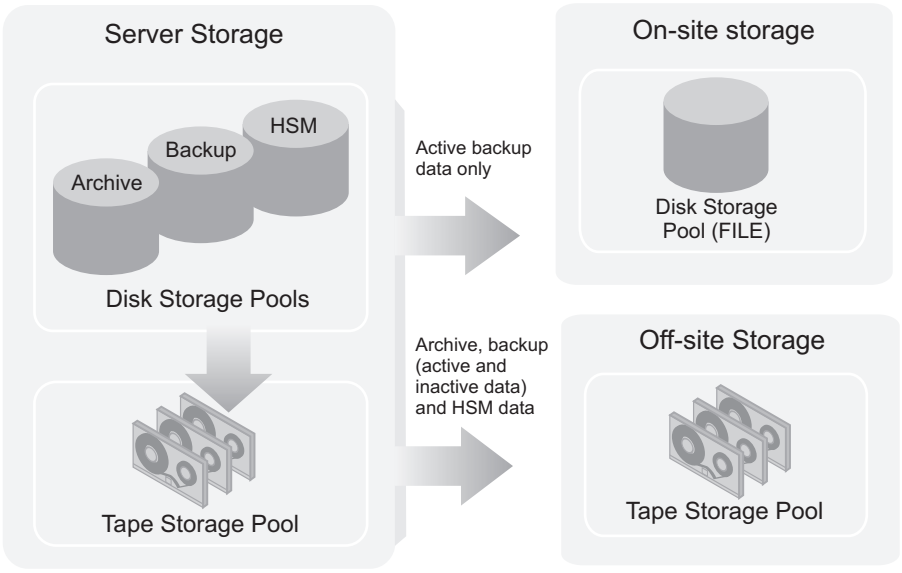


Figure 105. Copy storage pools and active-data pools

Related concepts:

"Active-data pools" on page 237

"Copy storage pools" on page 237
"Primary storage pools" on page 236

Related tasks:

"Storage pool hierarchies" on page 256

Backing up primary storage pools

To protect client data, back up your primary storage pools to copy storage pools. Copy active client backup data in primary storage pools to active-data pools for fast client restores.

About this task

Tip: 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.

Each of the commands in the following examples 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.

Procedure

- To back up data in a primary storage pool to a copy storage pool, use the **BACKUP STGPOOL** command. For example, to back up a primary storage pool named ARCHIVEPOOL to a copy storage pool named DISASTER-RECOVERY, issue the following command:

  ```
  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 exist in the copy storage pool. The data format of the copy storage pool and the primary storage pool can be NATIVE, NONBLOCK, or the NDMP formats NETAPPDUMP, CELERRADUMP, or NDMPDUMP. The server copies data from the primary storage pool only to a copy storage pool that has the same format.

  Restriction: 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.

  Tip: To further minimize the potential 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 on-site. Ensure that you mark the volumes as OFFSITE before you back up the database. To avoid marking volumes as offsite or physically move volumes:
  - Specify a device class of SERVER in your database backup.
  - Back up a primary storage pool to a copy storage pool or associated with a device class of SERVER.

- To copy active data, use the **COPY ACTIVEDATA** command. For example, to copy active data from a primary storage pool named BACKUPPOOL to an active-data pool named CLIENT-RESTORE, 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 exist in the active-data pool.

Results

Because backups and active-data copies are made incrementally, you can cancel the processes. If you reissue the **BACKUP STGPOOL** or **COPY ACTIVEDATA** command, the backup or active-data copy continues from the point at which the process was canceled.

Restrictions:

- If a backup is to be made to a copy storage pool and the file 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 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.

Related concepts:

"Active-data pools" on page 237

"Copy storage pools" on page 237

"Primary storage pools" on page 236

"Securing sensitive client data" on page 533

Related tasks:

"Backing up the data in a storage hierarchy" on page 261

Chapter 18, "Automating server operations," on page 609

Example: Scheduling a backup with one copy storage pool:

Create a schedule for backing up two primary storage pools to the same copy storage pool.

About this task

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.

Procedure

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.

Results

Tips:

- To minimize tape mounts, you can take one or both of the following steps:
  - Back up the disk storage pool first, then the tape storage pool.
  - If you schedule storage pool backups and migrations and have enough disk storage, back up or copy as many files as possible from the disk storage pool to copy storage pools and active-data pools. After the backup and copy operations are complete, migrate the files from the disk storage pools to primary tape storage pools.
- if you have active-data pools, you can schedule the **COPY ACTIVEDATA** command to copy the active data that is in primary storage pools to the active-data pools.

Related concepts:

"Active-data pools" on page 237

"Copy storage pools" on page 237

"Primary storage pools" on page 236

"Securing sensitive client data" on page 533

Related tasks:

"Backing up the data in a storage hierarchy" on page 261

Chapter 18, "Automating server operations," on page 609

Protecting data in a Centera storage pool:

Performing a storage pool backup for data stored in a Centera storage pool is not supported. To ensure the safety of the data, therefore, consider using the replication feature of the Centera storage device.

About this task

With this feature, you can copy data to a replication Centera storage device at a different location. If the data in the primary Centera storage pool become unavailable, you can access the replication Centera storage device by specifying its IP address using the **HLADDRESS** parameter on the **UPDATE DEVCLASS** command for the device class pointed to by the Centera storage pool. After the primary Centera storage device is re-established, you can issue the **UPDATE DEVCLASS** command again and change the value of the **HLADDRESS** parameter to point back to the primary Centera storage device. You must restart the server each time you update the **HLADDRESS** parameter on the **UPDATE DEVCLASS** command.

Related concepts:

"Files on sequential volumes (CENTERA)" on page 46

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 simultaneously stored into each copy storage pool specified for the primary storage pool.

About this task

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.

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.

Related tasks:

"Writing data simultaneously to primary, copy, and active-data pools" on page 327

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 are expired, deleted, or moved from the volume.

About this task

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 the time that is 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 erased from the volumes: The database references to these files are removed. Thus the file data might still exist on sequential volumes if the volumes are not immediately reused.

A disaster might force you to restore the database using a database backup that is not the most recent backup. In this case, some files might not be recoverable because the server cannot find them on current volumes. However, the files might exist on volumes that are in pending state.

Procedure

You might 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 before file expiration.
2. Use a primary, copy-storage, or active-data pool volume that is not rewritten and that contains the expired file at the time of database backup.

Results

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, delay the reuse of volumes for as long as you keep your oldest database backup.

Related tasks:

"Scenario: Protecting the database and storage pools" on page 886

Related reference:

"Running expiration processing to delete expired files" on page 506

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.

About this task

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 might 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.

Audit a volume when the following conditions are true:

- The volume is damaged.
- The volume was accessed for a long time, for example, six months
- A read or write error occurs while accessing the volume
- The database was 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.

Note: If Tivoli Storage Manager detects a damaged file on a Centera volume, then a command is sent to Centera to delete the file. If Centera is unable to delete the file because the retention period for the file is not expired, then the volume that contains the file is not be deleted.

To display the results of a volume audit after it completes, use the **QUERY ACTLOG** command.

Related tasks:

"Requesting information from the activity log" on page 790

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** parameter on 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 a 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 a 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 a volume in an 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 were 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.

About this task

Data validation is helpful if you introduce new hardware devices. The validation assures that the data is not corrupted 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** parameter 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)

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.

Related reference:

"Errors in an audit of active-data storage pool volumes"

"Errors in an audit of copy storage pool volumes" on page 877

"Errors in an audit of a primary storage pool volume" on page 877

Choosing when to enable data validation:

Data validation is available for nodes and storage pools. The forms of validation are independent of each other.

About this task

Figure 106 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 106 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.

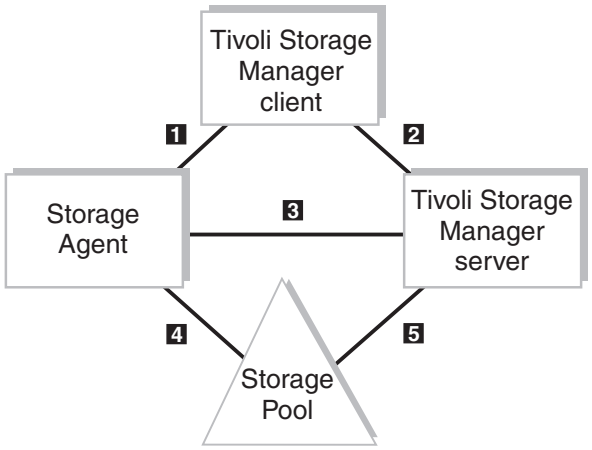


Figure 106. Data transfer eligible for data validation

Table 72 provides information that relates to Figure 106. This information explains the type of data being transferred and the appropriate command to issue.

Table 72. Setting data validation

Numbers in Figure 106	Where to Set Data Validation	Type of Data Transferred	Command	Command Parameter Setting
1	Node definition	File Data and Metadata	See *Note*	See *Note*
2	Node definition	File Data and Metadata	**REGISTER NODE UPDATE NODE**	VALIDATEPROTOCOL=ALL or VALIDATEPROTOCOL=DATAONLY
3	Server definition (storage agent only)	Metadata	**DEFINE SERVER UPDATE SERVER**	VALIDATEPROTOCOL=ALL

Table 72. Setting data validation (continued)

Numbers in Figure 106 on page 879	Where to Set Data Validation	Type of Data Transferred	Command	Command Parameter Setting
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 107 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 when the storage pool **CRCDATA** setting is 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.

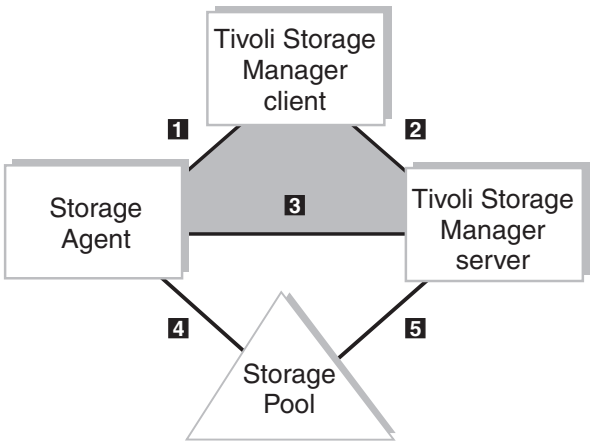


Figure 107. Protocol data validation versus storage pool data validation

If the network is unstable, you might decide to enable only 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 might have more critical data than others and might 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.

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 server-to-server exchange, you can use virtual volumes. 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 might decide to enable only 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 might decide that data validation for storage pools is no longer required after the devices prove to be stable.

Related tasks:

"Using virtual volumes to store data on another server" on page 722

"Auditing storage pool volumes" on page 875

Related reference:

"Validating a node's data during a client session" on page 530

Performance considerations for data validation:

Data validation affects performance because the server requires additional processor power 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:

Use the **AUDIT VOLUME** command to specify an audit for data written to volumes within a range of days, or to run an audit for a storage pool.

About this task

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 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.

About this task

To display the results of a volume audit after it completes, you can issue the **QUERY ACTLOG** command.

Procedure

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 Process Description      Status
  Number
-------- ----------------------- --------------------------------------------
       4 Audit Volume            Storage Pool BACKUPPOOL, Volume
         (Inspect Only)           /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 is stored. For example, Figure 108 on page 883 shows five volumes defined to ENGBACK2. In this example, File A spans VOL1 and VOL2, and File D spans VOL2, VOL3, VOL4, and VOL5.

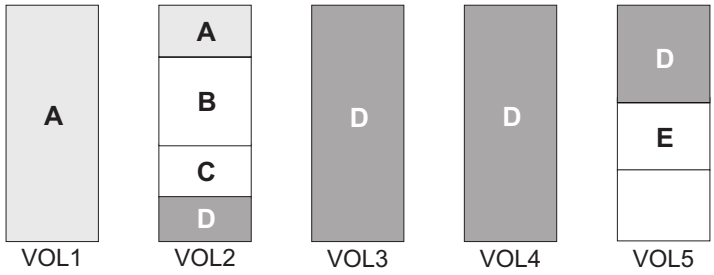


Figure 108. Tape volumes with files a, b, c, d, and e

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 demount 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.

About this task

This option is useful when the volume you want to audit contains part of a file, the rest of which is stored on a different, damaged volume. For example, to audit only volume VOL5 in the example in Figure 108 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.

About this task

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 might include disk volumes.

Procedure

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.

About this task

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.

About this task

For example, if your critical users store data in storage pool STPOOL3 and you want all volumes in the storage pool audited every two 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, you can take steps to correct them.

About this task

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 on-site 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.

Procedure

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 on-site copy storage pool or an active-data pool volume, it is used to provide client access to the file.
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.
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** parameter displays damaged files on a specific volume.

Related tasks:

"Data validation during audit volume processing" on page 878

"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.

About this task

This section explains how to restore damaged files based on the scenario in "Example: Scheduling a backup with one copy storage pool" on page 873.

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
```

Scenario: Protecting the database and storage pools

To demonstrate how you might protect the database and storage pools, suppose that your company takes steps to protect against the permanent loss of data that is stored in the database and storage pools. You can modify this procedure to meet your needs.

About this task

This scenario assumes a storage hierarchy that consists of the following storage pools:

- Default random-access storage pools named BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL
- A tape storage pool named TAPEPOOL

To provide extra levels of protection for client data, the scenario also specifies an offsite copy storage pool and an onsite active-data pool.

The standard procedures for the company 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

Procedure

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, can 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 ACTIVEDATA** 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 exist in the copy pool or active-data pool are backed up.

      Consider turning migration off during the rest of the day. You can add a schedule to migrate from disk to tape. 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. To change the access mode, issue 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 by using the **BACKUP DB** command. For example, issue the following command:

      ```
      backup db type=incremental devclass=tapeclass scratch=yes
      ```

      Restriction: Do not run the **MOVE DRMEDIA** and **BACKUP STGPOOL** or **BACKUP DB** commands concurrently. Ensure that the storage pool backup processes are complete before you issue the **MOVE DRMEDIA** command.

5. Perform the following operations nightly after the scheduled operations completes:
   a. Back up the volume history and device configuration files. If they change, 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 must be returned onsite. For example, issue the following command:

```
query volume stgpool=disaster-recovery access=offsite status=empty
```

These volumes, which became empty through expiration, reclamation, and file space deletion, waited the delay time that is 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 that are identified in step 5c onsite and update their access to read/write.

Recovering the database and client data

If a disaster occurs, recovering the database and client data are among the first priorities. Depending on your disaster recovery strategy, the recovery site can be your primary data center or a remote data center. Use the disaster recovery plan to organize recovery activities.

Related tasks:

Chapter 30, “Disaster recovery manager,” on page 979

Restoring the database

To restore the database, you must have the database backup volumes. The database backup volumes can be full, incremental, or snapshot.

About this task

For database restore operations, the Tivoli Storage Manager server reads the information that is in the volume history file to determine the number of data streams to read. The server attempts to match the number of streams that were used during the backup operation. For example, if the backup operation used four streams, the Tivoli Storage Manager server attempts the restore operation using four streams.

If you reduce the number of data streams after a database backup operation, this information will not be available to the server when the database is restored. To specify fewer data streams for the restore operation, take one or both of the following actions in the device configuration file:

- Reduce the number of online and usable drive definitions by removing **DEFINE DRIVE** commands.
- Update the value of the **MOUNTLIMIT** parameter of the **DEFINE DEVCLASS** command.

Related concepts:

“Configuring concurrent multistreaming” on page 860

Types of server database restores

Point-in-time restores are typically used for situations such as disaster recovery or to remove the effects of errors that can cause inconsistencies in the database. If you want to recover the database to the time when the database was lost, recover the database to its current state.

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

Database 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 and archive logs from archive log directory.

Restoring a server database to a point in time:

To restore a database to point in time, you need the latest full backup before the point in time. You also need the latest incremental backup after that last full backup. You can also use snapshot database backups to restore a database to a specific point in time.

Before you begin

Before restoring the database, have available the following infrastructure setup files:

- Server options file
- Volume history file:

  Copy the volume history file pointed to by the server options file. The backup copy must a different name. If the restore fails and you must try it again, you might need the backup copy of the volume history file. After the database is restored, any volume history information pointed to by the server options is lost. This information is required to identify the volumes to be audited.

  If your old volume history file shows that any of the copy storage pool volumes that are required to restore your storage pools were reused (STGREUSE) or deleted (STGDELETE), you might 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.
- Device configuration file:

  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.
- Detailed query output about the database and recovery log

If files were migrated, reclaimed, or moved after a backup, the files might be lost and the space occupied by those files might be 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.

Procedure

To restore the database to a point-in-time, complete the following steps:

1. 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
   ```

2. Use the **DSMSERV 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 completes 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. The first volume contains 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 specified date.

3. From the old volume history information that was generated by the **QUERY VOLHISTORY** command, obtain 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 remaining steps in this procedure. It might also be necessary to update the device configurations in the restored database.
4. Issue **AUDIT VOLUME** command and specify the **FIX=YES** parameter to audit all disk volumes, all reused volumes, and all deleted volumes.

   The audit volume process identifies files recorded in the database that can no longer be found on a volume. If a copy of the file is in a copy storage pool or an active-data pool, the file on the audited volume is marked as damaged. Otherwise, the file is deleted from the database and is lost.
5. If the audit detects any damaged files, issue the **RESTORE STGPOOL** command to restore those files after you audit the volumes in the storage pool.
6. 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.
7. Redefine any storage pool volumes that were added since the database backup.

What to do next

After a restore, the volume inventories for Tivoli Storage Manager and for your tape management system might 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 might also be inconsistent. Issue the **AUDIT LIBRARY** command to synchronize these inventories.

Related tasks:

"Updating the device configuration file" on page 894

"Restoring to a point-in-time in a shared library environment" on page 903

"Delaying reuse of volumes for recovery purposes" on page 874

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.

About this task

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.

Attention: Recovering the database to its most current state is not possible if the active or archive logs are lost.

Procedure

To restore a database to its most current state, issue the **DSMSERV RESTORE DB** command. For example:

```
dsmserv restore db
```

If the original database and recovery log directories are available, use the **DSMSERV RESTORE DB** utility to restore the database. However, if the database and recovery log directories are lost, recreate them first, and then issue the **DSMSERV RESTORE DB** utility.

Example: Restoring a library manager database

In this example, a library manager's corrupted database is restored. You can modify the procedure to meet your needs.

About this task

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.

Procedure

Complete the following steps to restore the corrupted database:

1. Copy the volume history file to a temporary location and rename the file. After the database is restored, any volume history information that is pointed to by the server options is lost. You 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 directories exist. If the original database or recovery log directories were lost, recreate them using the operating system **mkdir** command.

   Note: The directories must have the same name as the original directories.
5. Use the **DSMSERV RESTORE DB** utility to restore the database to the current time.
6. Start the Tivoli Storage Manager server instance.
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 any volumes that cannot be located as destroyed, and recover those volumes from copy storage pool backups. Recovery from active-data pool volumes is not suggested 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.

Results

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 installation date and time for the database. For servers that are defined for server-to-server communications, issue an **UPDATE SERVER** command with **FORCESYNC=YES**.

Related tasks:

"Restoring to a point-in-time a library manager server" on page 903

Example: Restoring a library client database

In this example, a library client's corrupted database might be restored. You can modify this procedure to meet your needs.

About this task

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 usage are known as library clients. Each library client maintains a database of volume usage and volume history. If the database of the library client becomes corrupted, it might be restored by following these steps:

Procedure

1. Copy the volume history file to a temporary location and rename the file. After the database is restored, any volume history information that is pointed to by the server options is lost. You 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 directories exist. If the original database or recovery log directories were lost, recreate them using the operating system **mkdir** command.

   Note: The directories must have the same name as the original directories.
5. Use the **DSMSERV RESTORE DB** utility to restore the database to the current time.
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 any volumes that cannot be located as destroyed, 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.

Results

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 installation date and time for the database. For servers that are defined for server-to-server communications, issue an **UPDATE SERVER** command with **FORCESYNC=YES**.

Related tasks:

"Restoring to a point-in-time a library client server" on page 904

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.

About this task

If this occurs, 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 is set to 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.

Related tasks:

"Recovering with different hardware at the recovery site" on page 1010

"Automated SCSI library at the original and recovery sites" on page 1010

Related reference:

Automated SCSI library at the original site and a manual scsi library at the recovery site

Restoring storage pools and storage pool volumes

When you restore a storage pool, the server determines which files are in that storage pool. Using file copies from a copy storage pool or an active-data pool, server restores the files that were in the storage pool 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.

About this task

The **RESTORE STGPOOL** command restores specified primary storage pools that have files with the following problems:

- The primary copy of the file had read errors during a previous operation. Files with read errors are marked as damaged.
- The primary copy of the file on a volume that has an access mode of DESTROYED..
- The primary file is 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.

Restrictions:

- Cached copies of files in a disk storage pool are never restored. References to any cached files were identified with read errors or cached files that are stored on a *destroyed* volume are removed from the database during restore processing.
- 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. Do not consider active-data pools for recovery of a primary pool unless the loss of inactive data is acceptable.
- You cannot restore a storage pool defined with a CENTERA device class.
- 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 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 becomes 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.

After the files are restored, the old references to these files in the primary storage pool are deleted from the database. Tivoli Storage Manager 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 were restored to other locations, the destroyed volume is automatically deleted from the database.

Tivoli Storage Manager uses database information to determine which files to restore for a 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 (DISCARDDATA=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.

Procedure

To restore a storage pool, use the **RESTORE STGPOOL** command. To identify volumes that contain damaged primary files, use the PREVIEW=YES parameter. 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.

What to do next

DRM: DRM can help you track your on-site 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.

Related tasks:

"Fixing damaged files" on page 884

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.

About this task

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.

Procedure

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 completing 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.

Related tasks:

"Keeping client files together using collocation" on page 353

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.

About this task

The restoration might 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 STGPOOL** command, but files were backed up to a different copy storage pool or

active-data pool. If you suspect this problem, use the **RESTORE STGPOOL** command again without specifying a copy storage pool or active-data pool from which to restore files. You can specify the **PREVIEW** parameter on the second **RESTORE STGPOOL** 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** and with the **DISCARDDATA** parameter to YES
  - **AUDIT VOLUME** with **FIX** parameter set to 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

Restore processing copies files from a copy storage pool or an active-data pool onto new primary storage pool volumes.

About this task

Tivoli Storage Manager uses database information to determine which files to restore for a volume. 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 (**DELETE VOLUME** command with the **DISCARDDATA** parameter set to YES), 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.

After files are restored, the server deletes database references to files on the original primary storage pool volumes. 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. A primary storage pool volume becomes 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.

Procedure

To recreate files for one or more volumes that were lost or damaged, use the **RESTORE VOLUME** command. 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.

What to do next

Attention:

- Cached copies of files in a disk storage pool are never restored. References to any cached files that are on a volume that is being restored are removed from the database during restore processing.
- You can also recreate active versions of client backup files in storage pool volumes by using duplicate copies in active-data pools. However, do not consider active-data pools for recovery of a volume unless the loss of inactive data is acceptable. If the server determines that an inactive file must be replaced but cannot find it in the active-data pool, restoring from an active-data pool might cause some or all inactive files to be deleted from the database.
- You cannot restore volumes in a storage pool defined with a CENTERA device class.

Scenario: Recovering a lost or damaged storage pool volume:

In this scenario, an operator inadvertently destroys a tape volume (DSM087) belonging to the TAPEPOOL storage pool. A Tivoli Storage Manager administrator recovers the data stored on the destroyed volume by using the offsite copy storage pool.

Procedure

The Tivoli Storage Manager administrator completes the following steps:

1. Determine the copy pool volumes that contain the backup copies of the files that were stored on the volume that was destroyed. Issue the following command:

   ```
   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 on-site 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 on-site.
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 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 on-site, 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.

Fixing an incomplete volume restoration:

When the restoration of a volume might be incomplete, you can get more information about the remaining files on volumes for which restoration was incomplete.

About this task

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 VOLUME** command, but files were backed up to a different copy storage pool or a different active-data pool. If you suspect this problem, use the **RESTORE VOLUME** command again without specifying a copy storage pool or active-data pool from which to restore files. You can specify the **PREVIEW** parameter on the second **RESTORE VOLUME** 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** with the **DISCARDDATA** parameter set to YES
  - **AUDIT VOLUME** with the **FIX** parameter set to 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.

Marking volumes as destroyed:

The *destroyed* volume access mode designates primary volumes for which files are to be restored.

About this task

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 many primary storage pool volumes after a major disaster.

Procedure

To mark a volume as destroyed, complete the following steps:

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 identify 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.

Restore and retrieve files by priority:

When you restore a file that has multiple copies that are stored on different sequential storage pools, the Tivoli Storage Manager server selects the storage pool volume to restore the file from by priority. The server also restores or retrieves files from the client by priority. The priority is based on the status of the volume.

If you did not specify preemption settings for the server, and a mounted volume is in use by another process, restore operations is pending until the other processes complete. The restore operation is pending even when an unmounted volume is available. In Tivoli Storage Manager V7.1, the server prioritizes idle volumes before unmounted volumes to enable a restore operation to complete faster.

The server selects a storage pool to restore or retrieve a file from in the following order:

1. Mounted file volume
2. Unmounted file volume
3. Unmounted VTL volume
4. Idle (mounted, but not in use) volume
5. Unmounted, tape library volume
6. Unmounted, stand-alone tape volume
7. In use, VTL or tape volume
8. Unavailable volume
9. Offsite volume
10. Irretrievable file volume

For example, there are three copies of a file stored in three storage pool volumes that are called V1, V2, and V3. The storage pool volumes have the following status:

- V1 is idle and is stored in a storage tape pool called P1
- V2 is in use and is stored in a storage tape pool called P2
- V3 is unmounted and is stored in a storage tape pool called P3

The file is restored from the V1 volume in the P1 storage pool because the storage pool volume has a higher priority.

Preventing duplication of restored files

When Tivoli Storage Manager restores data, it might cause some duplication of restored files. Duplication 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.

About this task

If duplication occurs, 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. Alternatively, 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 occurs only 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 do not designate 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 are restored a second time, and duplicate copies are generated when volume DupB2 is processed.

Restoring and recovering an LDAP server

If you use an LDAP directory server to authenticate passwords, you might need to restore its contents at some time.

There are ways to avoid locking your ID and not being able to logon to the server or rendering data unavailable.

- Give system privilege class to the console administrator ID.
- Make sure that at least one administrator with system privilege class can access the server with LOCAL authentication.
- Do not back up the LDAP directory server to the IBM Tivoli Storage Manager server. An administrator who backs up the Windows Active Directory or the IBM Tivoli Directory Server to the Tivoli Storage Manager server might render them unusable. The Tivoli Storage Manager server requires an external directory for the initial administrator authentication. Backing up the directory server to the Tivoli Storage Manager server locks the administrator ID and renders them unable to logon to the LDAP directory server.

Recovering an LDAP server with replication or other methods:

You must configure the LDAP settings on a target server before replicating, exporting, or importing nodes and administrators onto it.

You must run the **SET LDAPUSER** and **SET LDAPPASSWORD** commands, and define the **LDAPURL** option on the target server. If it is not configured properly, you can replicate, export, import, or use Enterprise Configuration on the target server. But all nodes and administrators that are transferred from the source to the target with the LDAP server are then changed to use LDAP authentication. Nodes and administrators that changed to LDAP authentication on the target server become inaccessible.

You can configure the target server for LDAP authentication after replicating or exporting to it, but the data is unavailable until that occurs. After configuring the LDAP settings at the target server level, the node or administrator entries must be set up on the LDAP server. Either share the LDAP server between the source and the target server, or replicate the source LDAP server to the target server. All applicable nodes and administrators are transferred to the target.

If the transfer is unsuccessful, the LDAP administrator must manually add the node and administrator passwords onto the LDAP server. Or you can issue the **UPDATE NODE** or **UPDATE ADMIN** commands on the IBM Tivoli Storage Manager server.

Audit the LDAP server

If the IBM Tivoli Storage Manager gets out-of-sync with the LDAP directory server, you might notice some unexpected errors. To put the data in-sync, issue the **AUDIT LDAPDIRECTORY** command.

After you issue the **AUDIT LDAPDIRECTORY FIX=YES** command, the following events occur:

- All nodes and administrators that were removed from the LDAP directory server are listed for you.
- All nodes and administrators that are missing from the LDAP directory server are listed for you. You can correct these missing entries by issuing the **UPDATE NODE** or **UPDATE ADMIN** command.

- A summary is provided to you on how many entries were examined on the LDAP directory server.
- Error messages are issued to you if the Tivoli Storage Manager server cannot connect to the LDAP directory server. The configuration is controlled by the **LDAPURL**option settings.

If you want to see what is out-of-sync without removing any nodes or administrators, issue the **AUDIT LDAPDIRECTORY FIX=NO** command. After issuing this command, you receive reports on the discrepancies between the Tivoli Storage Manager server database and the LDAP directory server.

If multiple Tivoli Storage Manager servers share an LDAP directory server, avoid issuing the **AUDIT LDAPDIRECTORY FIX=YES** command.

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.

Restoring to a point-in-time a library manager server

A point-in-time restore of a library manager server can create inconsistencies between the volume inventories of the library manager and library client servers. Steps must be taken to prevent this problem.

About this task

The restore operation removes all library client server transactions that occurred after the point in time from the volume inventory of the library manager server. However, the volume inventory of the library client server still contains those transactions. New transactions can then be written to these volumes, resulting in a loss of client data.

Procedure

Complete the following steps 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 re-enter those volume transactions that were removed by the restore on the library manager server. 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 can cause volumes to be removed from the volume inventory of a library client server and later overwritten.

About this task

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, complete the following steps:

Procedure

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, review the volume history and issue the **UPDATE VOLUME** command as needed.

Example: Recovering to a point-in-time

An administrator restores a library manager server and a library manager client to the point-in-time of the last backup.

About this task

The processor on which Tivoli Storage Manager is located, the database, and all on-site 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:

Procedure

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 on-site from the offsite location.

   Note: Do not change the access mode of these volumes until after you complete step 7 on page 905.
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.
5. Restore the database from the latest backup level by issuing the **DSMSERV RESTORE DB** utility.
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 on-site at the time of the disaster. Any volumes that were on-site would were 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 DISCARDDATA 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
   ```

   Clients can now 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. With the new volumes, clients can also back up, archive, or migrate files to the server. If you use only scratch volumes in the storage pool, you are not required to complete this step.
10. Restore files in the primary storage pool from the copies 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 are restored for a destroyed volume, that volume is automatically deleted from the database.
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.

Related tasks:

"Updating the device configuration file" on page 894

"Recovering the database and client data" on page 888

"Fixing an incomplete storage pool restoration" on page 896

Chapter 29. Replicating client node data

Node replication is the process of incrementally copying, or *replicating*, data that belongs to backup-archive client nodes. Data is replicated from one IBM Tivoli Storage Manager server to another Tivoli Storage Manager server.

The server from which client node data is replicated is called a *source replication server*. The server to which client node data is replicated is called a *target replication server*. A server can function as the source server of replicated data for some client nodes and as the target server of replicated data for other client nodes.

The purpose of replication is to maintain the same level of files on the source and the target replication servers. As part of replication processing, client node data that was deleted from the source replication server is also deleted from the target replication server. When client node data is replicated, only the data that is not on the target replication server is copied.

If a disaster occurs and the source replication server is temporarily unavailable, client nodes can recover their data from the target replication server. If the source replication server cannot be recovered, you can convert client nodes to store data on the target replication server. When there is an outage, the source replication server can automatically fail over to a target replication server for data recovery.

You can replicate the following types of client node data:

- Active and inactive backup data together, or only active backup data
- Archive data
- Data that was migrated to a source replication server by Tivoli Storage Manager for Space Management clients

Use node replication for data recovery at a disaster recovery site and to maintain the same level of files on the source and target replication servers. Node replication is used for the following objectives:

- Controlling network throughput by scheduling node replication at specific times
- Recovering data from large-scale site loss

To ensure high server availability, you can use node replication with clustering. In a clustered environment, a client is less likely to fail over to another server. If you replicate data from several source replication servers to one target replication server, there is a high dependency on the target replication server. A clustered environment eases the dependency on the target replication server.

The following table specifies replication compatibility for Tivoli Storage Manager source and target replication servers.

Table 73. Replication compatibility of Tivoli Storage Manager servers

Source replication server version	Compatibility with target replication server version
V6.3.0 - V6.3.2	V6.3.0 - V6.3.2
V6.3.3	V6.3.3 or later
V6.3.4 or later	V6.3.4 or later

© Copyright IBM Corp. 1993, 2013

Table 73. Replication compatibility of Tivoli Storage Manager servers (continued)

Source replication server version	Compatibility with target replication server version
V7.1	V7.1

You can replicate data for client nodes that are V6.3 or earlier. You can also replicate data that was stored on a Tivoli Storage Manager V6.2 or earlier server before you upgraded it to V6.3. You can also replicate data that was stored on a Tivoli Storage Manager V6.2 or earlier server before you upgraded it to V6.3.

For information about implementing a replication configuration, see "Setting up the default replication configuration" on page 934.

Related concepts:

Managing passwords and logon procedures

Automatic failover for data recovery overview

Automatic failover for data recovery occurs if the source replication server is unavailable because of a disaster or a system outage.

During normal operations, when the Tivoli Storage Manager V7.1 client logs in to a source replication server, it receives connection information for the target failover server. The client node stores the failover connection information in the client options file. During client restore operations, the Tivoli Storage Manager server automatically changes clients to the target replication server and back again. Only one failover server can exist per node at any time. The server information is stored in the options file. The failover server can be modified only if the default replication server is modified and another replication is completed for the node.

To use automatic failover for replicated client nodes, the source replication server, the target replication server, and the client must be at the v7.1 level or later. If any of the servers are at an earlier level, automatic failover is disabled and you must rely on manual failover.

If the client cannot connect to the source replication server, it uses the failover connection information and attempts to log on to the target failover server. The client logs on to the target replication server and is allowed only to recover data. The client cannot store data during failover processing.

When a new client operation is started, the client attempts to connect to the source replication server. The client resumes operations on the source server if the source replication server is available.

When you configure a node for replication by using the **REGISTER NODE** command or the **UPDATE NODE** command, the failover server is not specified.

Related tasks:

"Restoring, retrieving, and recalling data manually from a target replication server" on page 976

Source and target node-replication servers

A *source replication server* is a Tivoli Storage Manager server that has client node data. You replicate the client node data to another server, called the *target replication server*. The client node data on the target replication server is managed by policies on the source replication server. You cannot replicate nodes from a Tivoli Storage Manager V6.3.3 or later source replication server to a target replication server that is running on an earlier version of Tivoli Storage Manager.

Replication server configurations

A source replication server can replicate client node data to only one target replication server at a time. However, a single target replication server can be the target for multiple source replication servers. A client node cannot back up, archive, or migrate data to a server where the node is a target node for replication.

The example configuration in the following figure illustrates the relationships between source and target replication servers in a network.

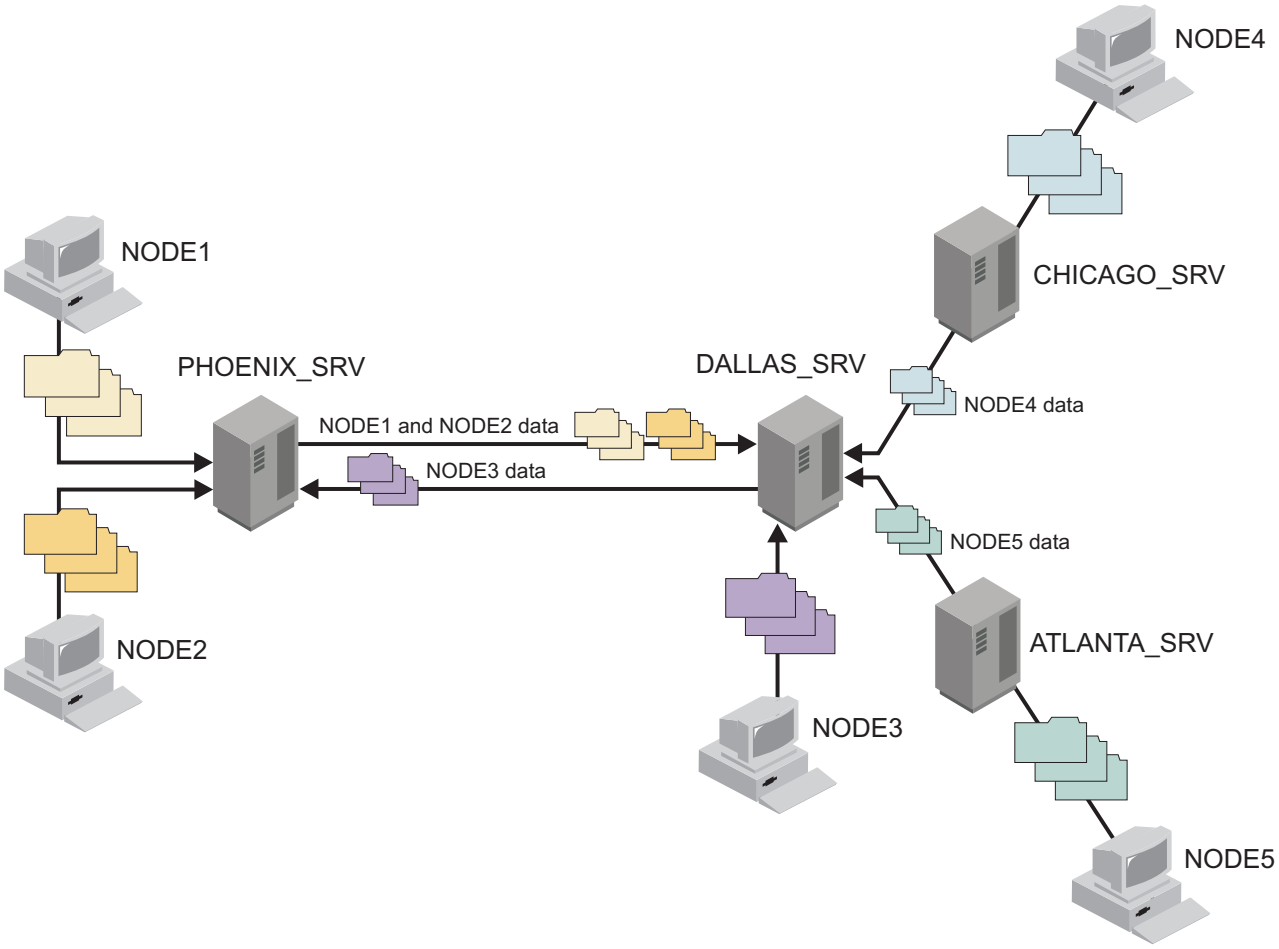


Figure 109. Networked configuration of replication servers

Figure 109 depicts the following data movements:

- PHOENIX_SRV and DALLAS_SRV replicate client node data to each other. Each server is both a source and a target for replicated client-node data.

- NODE1 and NODE2 are defined to PHOENIX_SRV. The nodes are configured for replication. During store operations, NODE1 and NODE2 send their data to PHOENIX_SRV. The data is then replicated to DALLAS_SRV.
- NODE3 is defined to DALLAS_SRV. The node is configured for replication. During store operations, NODE3 sends its data to DALLAS_SRV. The data is then replicated to PHOENIX_SRV.
- Data belonging to NODE4 and NODE5 is replicated to DALLAS_SRV.

Data that is replicated to a target server cannot be replicated again. For example, the data that belongs to NODE4 cannot be replicated to PHOENIX_SRV from DALLAS_SRV.

Policy management for node replication

The management class and storage hierarchy that are used by a client node on a target replication server depend on whether the domain for the node exists on the target server.

When a client node is registered on a target replication server, the domain for the node is sent to the target server. If the target server does not have a domain with the same name, the node on the target server is placed in the standard domain on the target server and bound to the default management class.

To maintain the same number of file versions on the source and the target replication servers, the source replication server manages file expiration and deletion. If a file on the source replication server is marked for deletion, but not yet deleted by the expiration processing, the target replication server deletes the file during the next replication process. Expiration processing on the target replication server is disabled for replicated data. The file on the target replication server is deleted by the source replication server after the file is expired and deleted on the source.

If a client node is removed from replication on the target replication server, the policies on the target replication server are enabled. Data on the target replication server is then managed by the policies that are on the target replication server, and expiration processing can delete expired files.

Important: Policies that are defined on replication servers and that are dissimilar can cause undesirable side-effects. As newer versions of backup files are replicated, versions that exceed the value of the **VEREXISTS** parameter for the copy group are marked for immediate deletion. If the node that owns the files is configured for replication, expiration does not delete the files. However, because these files are marked for immediate deletion, they are not available for the client to restore. The files remain in the storage pool until replication deletes them based on the policy on the source replication server.

Tips:

- Policies and storage pool hierarchies on the source and target replication servers can be different. You can use deduplicated storage pools on the source replication server, on the target replication server, or both. However, to maintain synchronization of the data on source and target replication servers, configure the management classes on the source and target replication servers to manage data similarly. To coordinate policies, consider using Tivoli Storage Manager enterprise configuration.
- Ensure that sufficient space is available in the storage pool on the target replication server.

- If a replicated file on the target replication server is deleted, it is eligible for replication the next time replication runs for the client node that owns the file.

Node replication processing

Replication processing involves the interaction between replication rules, states, and modes. Data deduplication and the updating of client node attributes are also factors in node replication processing.

Replication rules

Replication rules control what data is replicated and the order in which it is replicated.

Replication rule definitions

Replication rules are assigned to data types. Data types include backup data, archive data, and space-managed data that is migrated by Tivoli Storage Manager for Space Management clients.

The Tivoli Storage Manager server has the following predefined set of replication rules. You cannot create replication rules.

ALL_DATA
Replicates backup, archive, or space-managed data. The data is replicated with a normal priority. For example, you can assign the ALL_DATA rule to backup data and archive data, and assign a different rule to space-managed data.

ACTIVE_DATA
Replicates only active backup data. The data is replicated with a normal priority. You can assign this rule only to the backup data type.

ALL_DATA_HIGH_PRIORITY
Replicates backup, archive, or space-managed data. The data is replicated with a high priority. In a replication process that includes both normal-priority and high-priority data, high-priority data is replicated first.

ACTIVE_DATA_HIGH_PRIORITY
Replicates active backup data. The data is replicated with a high priority. You can assign this rule only to the backup data type.

DEFAULT
Replicates data according to the rule that is assigned to the data type at the next higher level in the *replication-rule hierarchy*. The replication-rule hierarchy comprises file space rules, individual client-node rules, and server rules. Server rules apply collectively to all nodes that are defined to a source replication server and that are configured for replication.

Rules that are assigned to data types in file spaces take precedence over rules that are assigned to data types for individual nodes. Rules that are assigned to data types for individual nodes take precedence over server rules. For example, if the DEFAULT replication rule is assigned to back up data in a file space, the server checks the replication rule for backup data that is assigned to the client node. If the client node rule for backup data is DEFAULT, the server checks the server rule for backup data.

The DEFAULT rule is valid only for data types at the file space level and the client node level. It is not valid for data types at the server level.

NONE
Backup, archive, or space-managed data is not replicated. For example, if you do not want to replicate archive data, assign the NONE rule to the archive data type.

Tip: When you set up the default replication configuration, you do not have to assign or change replication rules. Tivoli Storage Manager automatically assigns the DEFAULT replication rule to all data types in the file spaces and in the client nodes that you configured. The system-wide replication rules are automatically set to ALL_DATA. You can change file space, client-node, and system-wide rules after you set up the default configuration.

If a file space is added to a client node that is configured for replication, the file space rules for data types are initially set to DEFAULT. If you do not change the file space rules, the client node and server rules determine whether data in the file space is replicated.

Replication rule attributes

Replication rules have attributes. You can modify the replication state of a rule.

A replication rule has the following attributes:

Rule name
Specifies the name of a replication rule, such as ALL_DATA, ALL_DATA_HIGH_PRIORITY, and so on.

Priority
Specifies the order in which data is replicated. Data can be replicated with normal priority or high priority. In a replication process that includes both normal-priority and high-priority data, high-priority data is replicated first. The replication rules ALL_DATA and ACTIVE_DATA are normal priority. The replication rules ALL_DATA_HIGH_PRIORITY and ACTIVE_DATA_HIGH_PRIORITY are high priority.

Active-data only?
Specifies that the rule replicates only active backup data. The value of this attribute is YES for the rules ACTIVE_DATA and ACTIVE_DATA_HIGH_PRIORITY.

State Specifies whether a replication rule is enabled or disabled. Before data is replicated, the state of the rule that applies to the data is checked. If the state is enabled, the data is replicated. If the state is disabled, the data is skipped during replication.

You can change the state of a replication rule.

To display the attributes of replication rules, issue the **QUERY REPLRULE** command.

Replication rule hierarchy

Replication rules are applied to data types in file spaces and in individual client nodes. Replication rules are also applied collectively at the server level to all client nodes that are defined to the server and that are configured for replication. File space rules take precedence over rules for individual nodes. Rules for individual nodes take precedence over server rules.

In a client node that is configured for replication, each file space has three replication rules. One rule applies to backup data in the file space. The other rules apply to archive data and to space-managed data. The rules for the file space exist regardless of whether the file space has backup, archive, or space-managed data. If

a file space does not contain a particular type of data, the rule for the data type in the file space is ignored during replication processing.

Similarly, each client node that is configured for replication has replication rules for backup data, archive data, and space-managed data. Client node rules apply to all the file spaces that belong to a node. Replication rules also exist at the server level that applies collectively to every client node that is configured for replication on a source replication server.

The following figure shows the replication rule hierarchy.

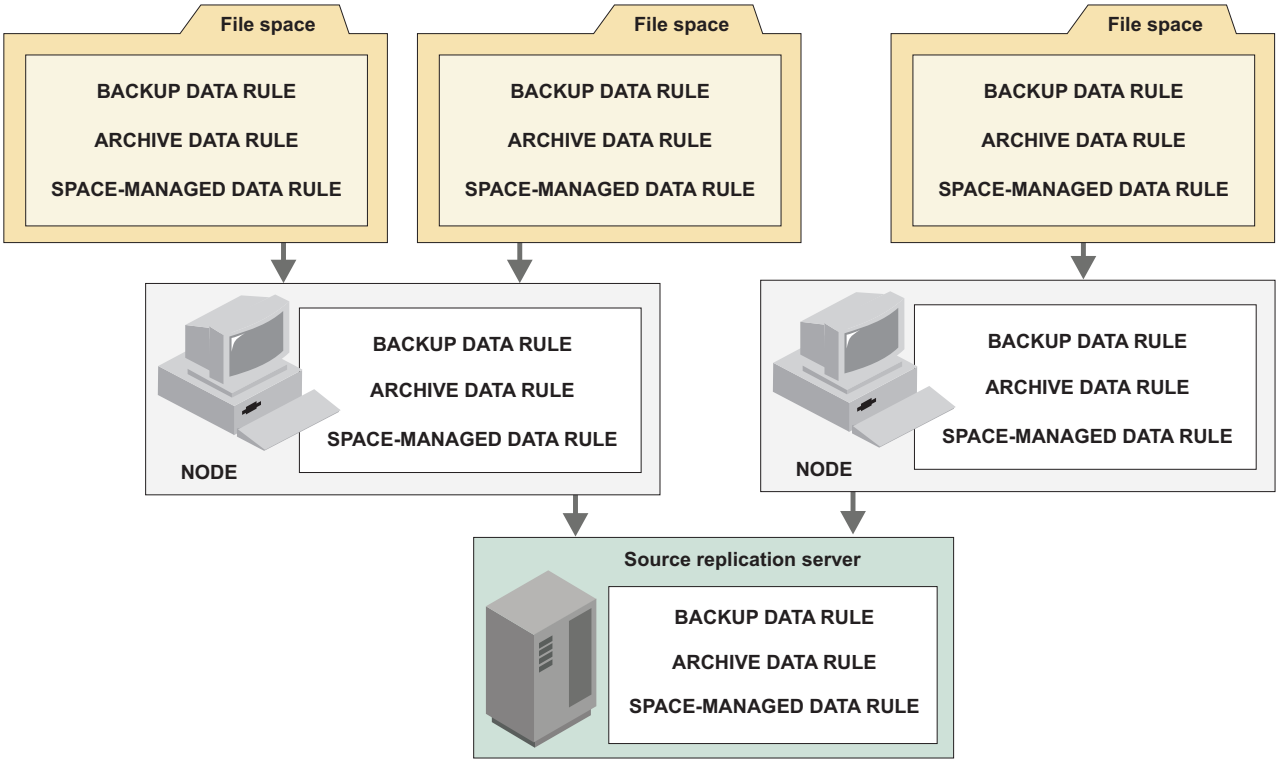


Figure 110. Replication-rule hierarchy

During replication processing, file space rules take precedence over rules for individual nodes. Rules for individual nodes take precedence over server rules. The replication rule that has precedence is called the *controlling replication rule*.

Example: Processing replication rules

In this example, replication rules are assigned to data types in file spaces, client nodes, and the server.

The following figure shows an example replication configuration that comprises two client nodes and three file spaces. This figure shows how Tivoli Storage Manager processes replication rules.

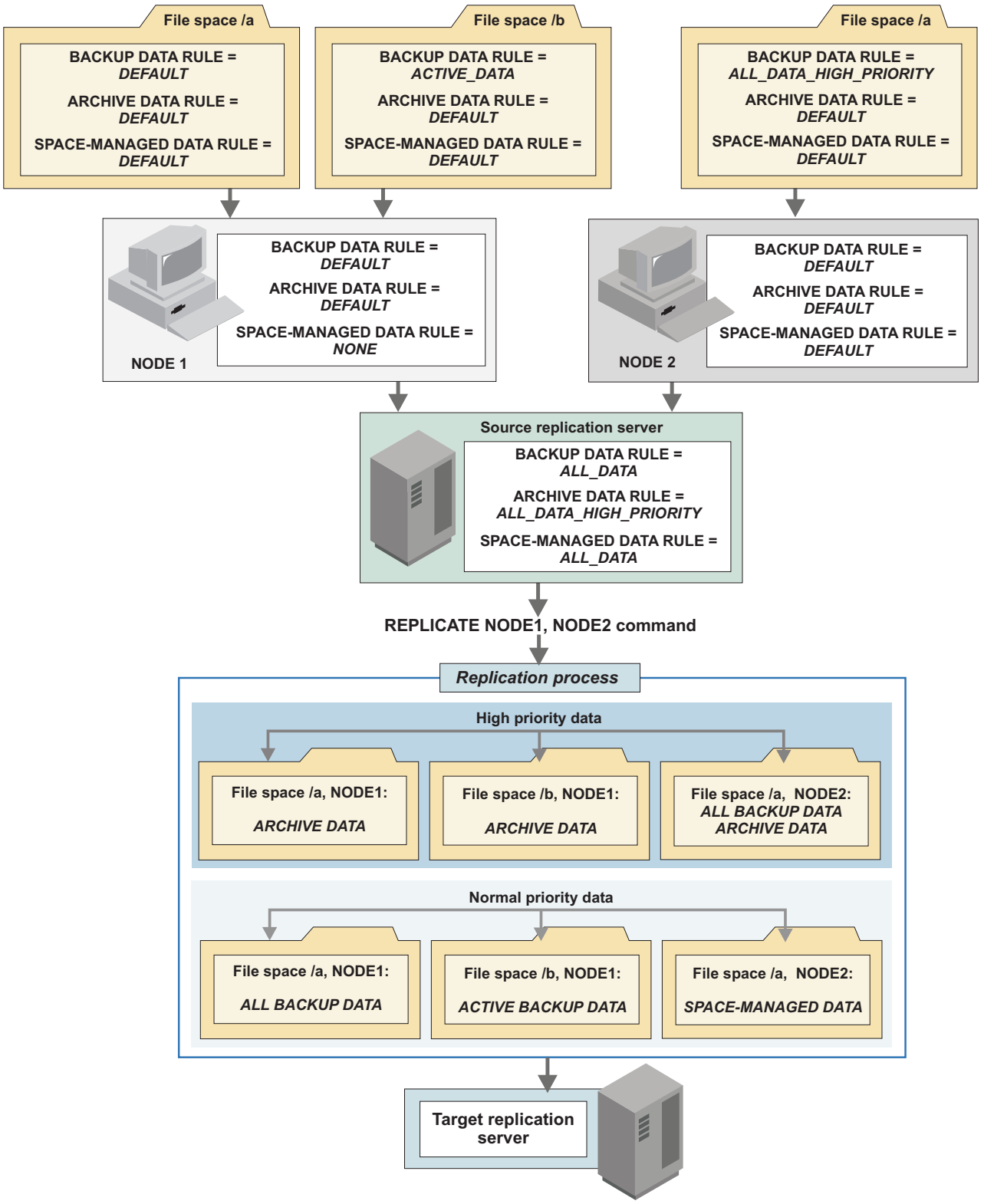


Figure 111. Replication configuration example

When the **REPLICATE NODE** command is issued, a single replication process begins. The source replication server identifies client nodes that are configured for replication and the rules that apply to the file spaces in nodes that are enabled.

For example, the file spaces in NODE1 and NODE2 contain important archive data. Replication of that data is a high priority. The source replication server checks the rules for archive data in file spaces. Because the rules are set to DEFAULT, the source replication server checks the client-node replication rules for archive data. These rules are also set to DEFAULT. As a result, the source replication server checks the server replication rule for archive data. Because it is set to ALL_DATA_HIGH_PRIORITY, archive data in file spaces that belong to NODE1 and NODE2 is replicated before normal-priority data.

The backup data in file space `/a` that belongs to NODE2 is also high priority. The file space rule for backup data, which is ALL_DATA_HIGH_PRIORITY, takes precedence over the client node rule of DEFAULT and the server rule of ALL_DATA.

Tips:

- Figure 111 on page 914 shows one possible configuration to achieve the specified results. In general, multiple configurations can exist that accomplish the same purpose.

  For example, to replicate archive data first, you can assign the ALL_DATA_HIGH_PRIORITY replication rule to the archive data type in each file space that belongs to NODE1 and NODE2.
- Figure 111 on page 914 shows one replication process. To replicate certain client nodes ahead of other client nodes, you can issue multiple **REPLICATE NODE** commands in sequence, either manually or in a maintenance script. Each command can specify a different client node or different file spaces in an individual client node. For example, suppose NODE1 contains a large amount of data and you want to conserve bandwidth. To replicate client node data sequentially, you can specify NODE1 in a single **REPLICATE NODE** command and NODE2 in another **REPLICATE NODE** command.

Related concepts:

"Replication rule hierarchy" on page 912

"Replication rule definitions" on page 911

Replication state

Replication state indicates whether replication is enabled or disabled. When you disable replication, replication does not occur until you enable it.

Replication states apply to the following objects:

Replication rules

The state of a replication rule can be enabled or disabled. If a replication rule is disabled, data to which the rule is assigned is not replicated.

For example, suppose that you have 500 client nodes that are defined to a source replication server. The replication rule that is assigned to archive data in some file spaces is ALL_DATA_HIGH_PRIORITY. The server rule that is assigned to archive data is also ALL_DATA_HIGH_PRIORITY. If you disable the ALL_DATA_HIGH_PRIORITY rule, archive data in any file space that is assigned that rule is skipped during replication processing.

Individual client nodes

When you disable replication for a client node, backup, archive, and space-managed data in file spaces that belong to the client node is skipped during replication processing. Disabling replication for a client node can be

useful, for example, if the data that belongs to the node is corrupted or contaminated. After the problem is fixed, you can enable replication.

For replication to occur, the client node must be enabled on the source and the target replication servers.

Data types in file spaces

You can enable or disable replication of one or more data types in a file space. For example, you can disable replication of backup data in a file space.

Disabling replication of a data type in a file space can be useful if, for example, a file space has large amounts of backup data and archive data. You can enable the replication of backup data and disable replication of archive data. After the backup data is replicated, you can enable replication of archive data.

You cannot disable replication for an entire file space. You can only disable replication for an entire file space by setting the state of each data type to disabled.

During replication, the replication state of a file-space data type is checked only on the source replication server. The replication state of a file space data type on the target replication server is irrelevant.

Figure 112 on page 917 shows the interaction of replication states and replication rules. In the example, NODE1 has a single file space `/a` that contains archive data. Assume that the replication state of NODE1 on the target replication server is ENABLED and that replication processing for all nodes is enabled.

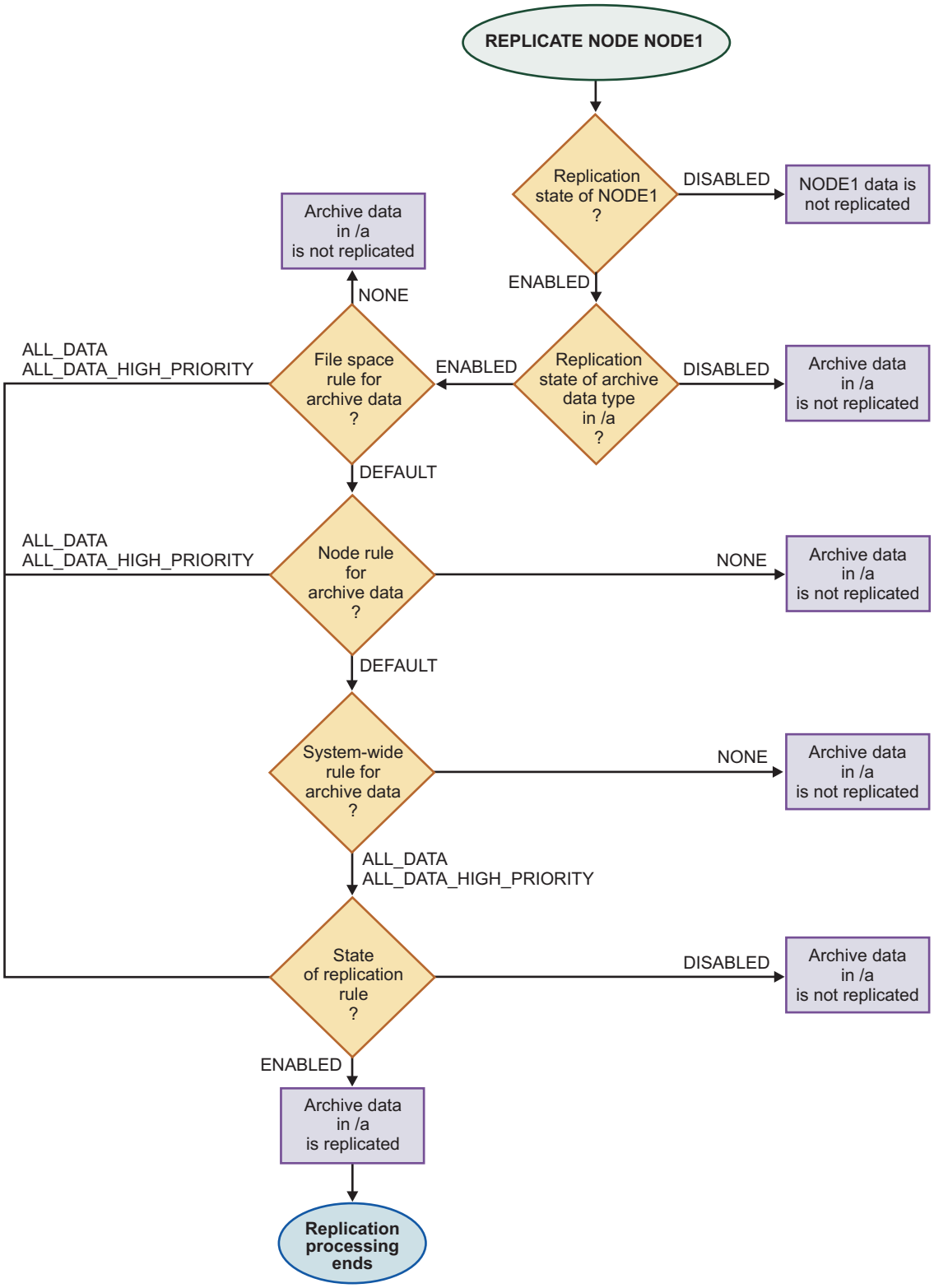


Figure 112. Replication state example

During replication processing, the following replication states are checked:

1. NODE1
2. Archive data type in file space `/a`

3. Replication rule for archive data

To determine the replication state of a file space, issue the **QUERY FILESPACE** command. To determine the replication state of a client node, issue the **QUERY NODE** command, and to determine the replication state of a rule, issue the **QUERY REPLRULE** command.

Replication mode

Replication mode is part of a client node definition and indicates whether a client node is set up to send or receive replicated data. The replication mode can also indicate whether the data that belongs to a client node is to be synchronized the first time that replication occurs. Data synchronization applies only to client nodes whose data was exported from the source replication server and imported on the target replication server.

The following modes are possible for a client node whose data is not being synchronized:

SEND Indicates that the client node is set up to send data to the target replication server. The SEND replication mode applies only to the client node definition on a source replication server.

RECEIVE
Indicates that the client node is set up to receive replicated data from the source replication server. The RECEIVE replication mode applies only to the client node definition on a target replication server.

NONE
The client node is not configured for replication. To be configured for replication, the client node must be enabled or disabled for replication.

If the data that belongs to a client node was previously exported from a source replication server and imported on a target replication server, the data must be synchronized. Synchronization is also required after a database restore to preserve the client node data that is on the target replication server. When the data that belongs to a client node is synchronized, entries in the databases of the source and target replication servers are updated.

The following special settings for replication mode are required to synchronize data.

Restriction: To synchronize data, the date of the imported data on the target replication server must be the original creation date.

SYNCSEND
Indicates that data that belongs to the client node on the source replication server is to be synchronized with the client node data on the target replication server. The SYNCSEND mode applies only to the client node definition on a source replication server.

When data synchronization is complete, the replication mode for the node on the source replication server is set to SEND.

SYNCRECEIVE
Indicates that data that belongs to the client node on the target replication server is synchronized with the client node data on the source replication server. This SYNCRECEIVE mode applies only to the client node definition on the target replication server.

When data synchronization is complete, the replication mode for the node on the target replication server is set to RECEIVE.

Replication of deduplicated data

Data deduplication is a method for eliminating redundant data that is stored in sequential-access disk (FILE) primary storage pools, copy storage pools, and active-data storage pools. Before the data is replicated, the source replication server determines whether storage pools are set up for data deduplication.

Restriction: During replication processing, the simultaneous-write function is disabled on the target replication server when you store data to a primary storage pool that is enabled for data deduplication. Data that is replicated consists of only files or extents of data that do not exist on the target replication server.

The following table shows the results when storage pools on source and target replication servers are enabled for data deduplication. The destination storage pool is specified in the backup or archive copy-group definition of the management class for each file. If the destination storage pool does not have enough space and data is migrated to the next storage pool, the entire file is sent, whether the next storage pool is set up for deduplication.

If the storage pool on the source replication server is	And the destination storage pool on the target replication server is	The result is
Enabled for data deduplication	Enabled for data deduplication	Only extents that are not stored in the destination storage pool on the target replication server are transferred.
Enabled for data deduplication	Not enabled for data deduplication	Files are reassembled by the source replication server and replicated in their entirety to the destination storage pool.
Not enabled for data deduplication	Enabled for data deduplication	The source replication server determines whether any extents were identified for files that were previously stored in deduplicated storage pools. Any files that were never in a deduplicated storage pool are replicated in their entirety. For files that had extents that were previously identified, only extents that do not exist in the destination storage pool are transferred.
Not enabled for data deduplication	Not enabled for data deduplication	Files are replicated in their entirety to the destination storage pool.

Tip: If you have a primary storage pool that is enabled for deduplication on a source replication server, you can estimate a size for a new deduplicated storage pool on the target replication server. Issue the **QUERY STGPOOL** command for the primary deduplicated storage pool on the source replication server. Obtain the

value for the amount of storage space that was saved in the storage pool as a result of server-side data deduplication. This value is represented by the field `Duplicate Data Not Stored` in the command output. Subtract this value from the estimated capacity of the storage pool.

Related tasks:

"Deduplicating data" on page 281

"Writing data simultaneously to primary, copy, and active-data pools" on page 327

Client node attributes that are updated during replication

During node replication, the source replication server updates the client node attributes that are part of the node definition that is on the target replication server.

A Tivoli Storage Manager V7.1 server can replicate any node definition that is set as a proxy agent for a replicating node. The associated replication node definition must be replicated to the target replication server for failover to occur.

The following client node attributes are updated during node replication:

- Aggregation
- Automatic file space rename
- Archive delete authority
- Backup delete authority
- Backup initiation
- Cipher strength
- Client option sets
- Compression option
- Contact
- Data-read path
- Data-write path
- Email address
- File space access rules that are created with the client **SET ACCESS** command
- High-level address
- Low-level address
- Node lock state
- Operating system
- Option set name
- Password
- Password expiration period
- Proxy node
- Role override
- Session initiation
- Transaction group maximum
- URL
- Validate protocol

Attention:

- Ensure that all servers in your replication configuration do not create client option set conflicts on target replication servers. When an option set is created, all client nodes that reference that option set use the options that are defined. Client nodes that are replicated from different source replication servers can use the same option set. Only the first option set that is replicated is defined on the target replication server.
- The node password for the client node is updated during replication on the source replication server by one server and on the target replication server by a different server. Because authentication can happen on a Lightweight Directory Access Protocol (LDAP) directory server or the Tivoli Storage Manager server, data can be lost. If dual authentication occurs, the password is not updated during replication.

The following client node attributes are not updated during node replication:

- Domain name (might not exist on target replication server)
- Node conversion type
- Target level for automatic client deployment

For details about client node attributes, go to Tivoli Documentation Central at https://www.ibm.com/developerworks/community/wikis/home/wiki/Tivoli Documentation Central, and select **Tivoli Storage Manager**. Open the information center that matches the version of Tivoli Storage Manager that you installed. You can then search for the attribute.

Node replication restrictions

Restrictions can affect planning and implementation. For example, Tivoli Storage Manager applies the replication rule for archive data to the data that was migrated by the HSM for Windows client.

The following restrictions apply to node replication:

Store operations to a target replication server

If a client node is configured for replication, you cannot back up, archive, or migrate its data to the server that is the target replication server for the replicated data that belongs to the node.

Failover to a target replication server

You can use automatic failover only with Tivoli Storage Manager V7.1 servers and clients. Only one failover server for each node can be used. The failover server is always set to the last target server to which the node was replicated. The client can recover data from the target replication server, but it cannot store data during failover processing. Issue the **QUERY SESSION** command to determine whether a client node is in failover mode.

Client node definition on the target replication server

If you plan to add a node for replication, the client node definition cannot exist on the target replication server. If the client node definition does exist on the target replication server, you must remove or rename the node before replication can occur.

However, if the data that belongs to a client node was exported from the source replication server and imported on the target replication server, you do not have to delete the client node definition on the target replication server. To replicate, the data must be synchronized between the source and target replication servers. Synchronization occurs during replication.

To synchronize data, the data must be imported with the value of the **DATES** parameter that is set to ABSOLUTE on the **IMPORT NODE** command.

Import and export operations

Data that belongs to a replicated client node cannot be imported to the target replication server for replication. However, you can export the data that belongs to the client node from the source replication server to other servers. To export, you can use media or server-to-server virtual volumes. Replication rules are not exported.

Data migrated by the HSM for Windows client

The Tivoli Storage Manager for HSM for Windows client provides hierarchical storage management (HSM) for Windows NTFS file systems. When the HSM for Windows client stores data on the Tivoli Storage Manager server, the data is stored as archive data, not as space-managed data.

During replication processing, Tivoli Storage Manager applies the replication rule for archive data to the data that was migrated by the HSM for Windows client. For example, suppose that a backup-archive client has a file space that contains two directories. The data in one directory is archived to the Tivoli Storage Manager server. The data in the other directory is migrated by the HSM for Windows client, but it is stored as archive data. Both sources of data are associated with the same file space on the server.

If you set the file space replication rule for archive data to ALL_DATA and the file-space replication rule for space-managed data to NONE, the rule for space-managed data is ignored during replication processing. All the data in the file space is replicated to the target replication server according to the rule for archive data.

Objects that cannot be replicated

The following objects cannot be replicated to a target replication server:

- Replication rules
- Server node definitions
- Network-attached storage data in nonnative storage pools
- Client schedules
- Backup sets

Tips:

- If you want to convert client nodes for store operations to a target replication server, you can manually duplicate client schedules that are on the source replication server.
- You can generate backup sets on the target replication server for a replicated client node.

Retention protection

You cannot configure servers for replication on which archive retention protection is enabled.

Replication and file groups

When you are replicating files from one server to another, it is possible that some of the files that are being replicated belong to a group of files that are managed as a single logical entity. If a replication process ends without replicating all the files in a group, client nodes are unable to restore, retrieve, or recall the file group. When replication runs again, the source replication server attempts to replicate the missing files.

Renaming a node
If a node is configured for replication, it cannot be renamed.

Backing up a single client node to two source replication servers
If you have been backing up, archiving, or migrating a client node to two different servers, do not set up replication of the node from both source replication servers to the same target replication server. Replicating from two source servers might create different versions of the same file on the target server. Replicating from two source servers can cause unpredictable results when you restore, retrieve, or recall the file.

Password propagation to the target replication server
When client node data is replicated for the first time, the source server sends the node definition, including the password, to the target replication server. During subsequent replications, if the node password is updated, the source server attempts to send the updated password to the target replication server.

Whether these attempts succeed depends on the node authentication method and on the combination of methods that are used on the source and target replication servers. A conflict occurs if a node password is authenticated on the source server by one server and on the target replication server by a different server. Because authentication can happen on an LDAP (Lightweight Directory Access Protocol) directory server or the Tivoli Storage Manager server, data can be lost. If this kind of dual authentication occurs, the password is not updated during replication.

Simultaneous-write function
During replication processing, the simultaneous-write function is disabled on the target replication server when you store data to a primary storage pool that is enabled for data deduplication. Data that is replicated consists of only files or extents of data that do not exist on the target replication server.

Related tasks:

"Writing data simultaneously to primary, copy, and active-data pools" on page 327

Task tips for node replication

Task-tip topics identify the commands that you use for tasks that are related to node replication. Task-tip topics also provide links to topics with detailed information about a task.

Change replication rules

Customize a node replication configuration by changing the rules that apply to individual file spaces, to individual client nodes, and to collectively all client nodes that are configured for replication.

If you want to change replication rules...	Use these commands...	For more information, see...
For individual file spaces. For example, change an ALL_DATA rule for backup data to ACTIVE_DATA.	`UPDATE FILESPACE`	"Changing replication rules for individual file spaces" on page 940

If you want to change replication rules...	Use these commands...	For more information, see...
For individual client nodes. For example, change the replication rule that applies to backup data in all of the file spaces that belong to an individual node.	**UPDATE NODE**	"Changing replication rules for individual client nodes" on page 942
For all client nodes. For example, change a server replication rule of ALL_DATA to NONE.	**SET BKREPLRULEDEFAULT**, **SET ARREPLRULEDEFAULT**, and **SET SPREPLRULEDEFAULT**	"Changing server replication rules" on page 944

Add and remove client nodes for replication

Add client nodes for replication processing and remove them from replication processing.

If you want to...	Use these commands...	For more information, see...
Add client nodes for replication.	**REGISTER NODE** and **UPDATE NODE**	"Adding client nodes for replication processing" on page 950
Remove client nodes from replication. Removing a node from replication converts a replicating node to a non-replicating node. A non-replicating node can back up, archive, and migrate data directly to the target replication server.	**REMOVE REPLNODE**	"Removing client nodes from replication processing" on page 951 and "Converting client nodes for store operations on a target replication server" on page 977

Manage replication servers

Manage source and target replication servers. For example, change a target replication server or disable or enable inbound or outbound sessions.

Table 74. Commands for managing replication servers

If you want to...	Use these commands...	For more information, see...
Set up Secure Sockets Layer (SSL) communications between a source and target replication server	**DEFINE SERVER** and **UPDATE SERVER**	"Configuring a server for SSL communications" on page 956
Change a target replication server	**SET REPLSERVER**	"Selecting a new target replication server" on page 953
Remove a target replication server	**SET REPLSERVER** and **REMOVE REPLSERVER**	"Removing a replication server" on page 955
Control the number of node replication sessions	**REPLICATE NODE**	"Controlling throughput for node replication" on page 964

Table 74. Commands for managing replication servers (continued)

If you want to...	Use these commands...	For more information, see...
Disable or enable inbound or outbound sessions from a source or target replication server.	**DISABLE SESSIONS** and **ENABLE SESSIONS**	"Disabling and enabling outbound or inbound sessions" on page 968
Disable or enable outbound replication processing from a source replication server	**DISABLE REPLICATION** and **ENABLE REPLICATION**	"Disabling and enabling outbound node replication processing" on page 969
Remove a replication configuration	**REMOVE REPLNODE** and **SET REPLSERVER**	"Removing a node replication configuration" on page 977
Specify, at a server level, the IP address that the client uses to communicate with the failover server, if it is different from the IP address that replication uses	**SET FAILOVERHLADDRESS**	"Configuring the target replication server" on page 954

Validate a configuration and preview results

Before you replicate client node data, you can confirm that your replication rules are set up correctly. You can also preview results. For example, you can display a list of files to be replicated.

Table 75. Commands for validating a replication configuration and previewing results

If you want to...	Follow these steps...	For more information, see...
Determine whether server, client node, and file-space replication rules are set up correctly before you replicate data. You can also verify the connection between the source and target replication servers.	**VALIDATE REPLICATION**	"Validating a replication configuration" on page 958
Preview replication results.	**REPLICATE NODE**	"Previewing node replication results" on page 958
Display information about the replicating server.	**QUERY REPLSERVER**	"Removing a replication server" on page 955

Manage replication processing

Manage replication processing. For example, start a replication process or disable a replication rule.

Table 76. Commands for managing replication processing

If you want to...	Use these commands...	For more information, see...
Replicate data. You can replicate data by individual file space, by priority, and by data type.	**REPLICATE NODE** and **DEFINE SCHEDULE**	"Replicating data by command" on page 959
Temporarily disable replication for a data type in a file space.	**UPDATE FILESPACE**	"Disabling and enabling replication of data types in a file space" on page 966

Table 76. Commands for managing replication processing (continued)

If you want to...	Use these commands...	For more information, see...
Temporarily disable replication for an individual client node.	**UPDATE NODE**	"Disabling and enabling replication for individual client nodes" on page 967
Temporarily disable replication of data that is assigned a particular replication rule.	**UPDATE REPLRULE**	"Disabling and enabling replication rules" on page 970
Temporarily disable inbound and outbound server sessions, including replication sessions for all client nodes.	**DISABLE SESSIONS** and **ENABLE SESSIONS**	"Disabling and enabling outbound or inbound sessions" on page 968
Temporarily disable outbound replication processing from a source replication server.	**DISABLE REPLICATION** and **ENABLE REPLICATION**	"Disabling and enabling outbound node replication processing" on page 969
Prevent replication of backup, archive, or space-managed data in a file space on a source replication server, and delete the data from the target replication server.	**UPDATE FILESPACE**	"Purging replicated data in a file space" on page 971
Cancel all replication processes.	**CANCEL REPLICATION**	"Canceling replication processes" on page 973

Monitor replication processing and verify results

Preview results before replicating data, monitor replication processing, and verify results after replication ends.

Table 77. Commands for monitoring replication processing

If you want to...	Use these commands...	For more information, see...
Specify the number of days to retain replication records in the Tivoli Storage Manager database.	**SET REPLRETENTION**	"Retaining replication records" on page 976
Display information about the replication settings for a file space.	**QUERY FILESPACE**	"Displaying information about node replication settings for file spaces" on page 973
Display information about the replication settings for a client node.	**QUERY NODE**	"Displaying information about node replication settings for client nodes" on page 973
Display information about replication rules.	**QUERY REPLRULE**	"Displaying information about node replication rules" on page 973
Display records of running and ended replication processes.	**QUERY REPLICATION**	"Displaying information about node replication processes" on page 973

Table 77. Commands for monitoring replication processing (continued)

If you want to...	Use these commands...	For more information, see...
Determine whether replication to the target replication server is keeping pace with the number of files that are eligible for replication on the source replication server.	**QUERY REPLNODE**	"Measuring the effectiveness of a replication configuration" on page 974
Measure the effects of data deduplication.	**QUERY REPLICATION**	"Measuring the effects of data deduplication on node replication processing" on page 975

Planning for node replication

Use a checklist of planning considerations to streamline implementation. Planning considerations include items such as the types of data that you want to replicate and the replication rules that you want to apply.

About this task

As you plan, remember that a target replication server must be accessible from a source replication server by using an IP connection. The connection must provide sufficient bandwidth to accommodate the volume of data to be replicated. If the connection is insufficient and becomes a bottleneck for replication, keeping the data on the two servers synchronized can be a problem. Keep in mind that you can use client-side data deduplication with node replication to reduce network bandwidth requirements and storage requirements.

The destination storage pool on a target replication server must have sufficient space to store replicated data.

Include the following steps in your planning process:

Procedure

- Identify the client nodes and, if necessary, the files spaces that belong to those nodes, that have data to be replicated.
- Identify the types of data that you want to replicate. With Tivoli Storage Manager, you can replicate backup data, archive data, and space-managed data. If you decide to replicate backup data, you must decide whether to replicate only active backup data or to replicate active and inactive backup data. In environments with large amounts of backup data, it might make sense to replicate only active backup data. The type of data that you replicate can vary by node or by file space.
- Determine the order in which data is to be replicated. If you want to replicate some data ahead of other data in a replication process, you can assign a high-priority replication rule to the data that you want to replicate first. A high-priority classification does not increase the speed of replication.
- Set up a default replication configuration. In the default configuration, all backup, archive, and space-managed data is replicated in all file spaces in all replication-enabled client nodes. You can also change the replication rules at the file space, client node, and server levels. In general, you can apply different replication rules at different levels and achieve the same results.

- Identify source and target replication servers. If you have multiple source replication servers, consider whether you want to replicate data to one or more target replication servers. A single target replication server can have multiple source replication servers. However, a source replication server can replicate data to only one target replication server at a time.
- Consider using the enterprise configuration functions of Tivoli Storage Manager to set up, manage, and synchronize policy configurations on the source and the target replication servers. For example, you can change a policy configuration on a source replication server and have the changes that are automatically distributed to the target replication server.
- Consider how you want to replicate data. You can replicate one or more client nodes or defined groups of client nodes in a single process. You can schedule replication processes at different times to control network throughput.
- Decide whether you want to implement replication gradually or at one time. You can start with a simple configuration that comprises a single source replication server and a single target replication server. After you experiment with that configuration, you can build a broader network of replicating servers.
- If you use an LDAP directory server to authenticate passwords, any target replication servers must be configured for LDAP passwords. Data that is replicated from a node that authenticates with an LDAP directory server is inaccessible if the target replication server is not configured. If your target replication server is not configured, replicated data from an LDAP node can still be written to the replication server. However, the target replication server must be configured to use an LDAP directory so that you can access the data.
- Ensure that you allocate extra space for the active log and the archive log during replication processing.

Determining server database requirements for node replication

Node replication requires more Tivoli Storage Manager database space to track the files that are replicated.

About this task

To determine whether the database can manage more space requirements, you must estimate how much more database space that node replication will use.

Requirement: Place the database and database logs on separate disks that have a high performance capability. Use a separate disk or mount point for the following options:

- Other applications that use the database and logs
- System tasks, such as system paging

Procedure

1. Determine number of files for each node and data type that is in use. Issue the **QUERY OCCUPANCY** command for each node and data type that you plan to replicate. For example, you can display information about the file spaces that are assigned to the node named PAYROLL by issuing the following command:

   ```
   query occupancy payroll
   ```

2. Determine how much more database space is required by using the value for the total number of files that are used by all nodes and data types. Use the following formula to calculate the amount of database space that is required:

Total_number_of_files_from_all_nodes_and_data_types * 300 (the number of additional bytes needed for each replicated file)

Important: You must increase the available database space when the additional required space approaches or exceeds the size of your database. Ensure that you examine both replication servers and their databases and increase the database size if necessary.

3. Increase the size of the database by the additional database space required and include an additional 10% of the database size.

Estimating the total amount of data to be replicated

Determine the amount of data that is replicated initially and incrementally daily. Initial replications typically take longer to complete than incremental daily replications.

Procedure

1. Estimate the initial amount of data to be replicated. Decide which nodes, file spaces, and data types must be replicated. To calculate the total amount of physical space that is occupied for each file space and data type that must be replicated, issue the **QUERY OCCUPANCY** command for each node. For example, you can display information about the file spaces assigned to the node named PAYROLL by issuing the following command:

   ```
   query occupancy payroll
   ```

   Tip: Tune the performance of replication to the data type. For example, if you do not plan to replicate a data type in a file space, exclude the number of files for that data type.

2. Determine the amount of data that is backed up daily by the client nodes. Complete the following steps to estimate the amount of data that is replicated incrementally daily:
   a. When client nodes complete a store operation, the client logs completion messages with the server. The completion messages report statistics or information for a client operation that was transferred to the server. Message ANE4961I shows the number of bytes that are transferred during a client operation.
   b. To determine the average amount of bytes that are backed up daily by a particular node, add the number of bytes that are shown in the ANE4961I messages over several days. Then calculate the average amount of bytes to be replicated for all nodes to determine how much data is replicated daily.

Estimating network bandwidth required for replication

To ensure that the initial replication is successful, determine whether you have the network bandwidth, processor availability, and the time for replication. Estimating network bandwidth is based on the amount of data that must be replicated.

Procedure

1. To calculate the required network bandwidth, determine the following information:
   - Total amount of data (TD) to be replicated, in gigabytes. To calculate the total amount of data, see "Estimating the total amount of data to be replicated."
   - Length of replication window time (RWT) in hours. The length of the replication window time is the time that you schedule during server maintenance for replication to occur.

- Data deduplication ratio (DR), if you are using the data deduplication function. Issue the **QUERY STGPOOL** command and use the FORMAT=DETAIL value of to determine the data deduplication ratio value. If you are not using the data deduplication function, use 100 as the data deduplication ratio value.

2. Calculate the bandwidth required by using the following formula:

   `(TD * (100 / DR) * 8192) / (RWT * 3600) = Required_Network_Bandwidth (Mbps/second)`

What to do next

If the value for required network bandwidth exceeds the capabilities of your network, you must adjust the values in the formula. Reduce the TD value or increase the replication time, to reduce the value for *Required_Network_Bandwidth*. If you cannot adjust the TD or the RWT time values, adjust or replace your existing network to reduce the additional workload.

Calculating the time that is required for replication

You can estimate how many hours it will take to replicate data. You must know the amount of data you want to replicate and the bytes per hour that the network can process.

Procedure

1. Determine the time that is required for replication by using the number of bytes you want to replicate and the bytes per hour value. Use the following formula to calculate how many hours it takes to complete replication:

   Total_bytes_to_be_Replicated / Bytes_Per_Hour = Hours_to_Complete_Replication.

   To calculate the *Total_bytes_to_be_Replicated* and the *Bytes_Per_Hour* values, complete the steps in "Tuning replication processing" on page 965.

2. For the initial replication, determine how many hours it takes for the replication to occur over the network during a daily window for replication by using the following calculation:

   Days_to_Complete_Replication = Hours_to_Complete_Replication / 24

   Requirement: For daily incremental replications, the *Hours_to_Complete_Replication* value must be larger than the amount of time you schedule for replication.

What to do next

When you determine how long it takes for replication to finish, you can decide which method you use to complete the initial replication. The method that you use for the initial replication is based on the data, time, and bandwidth values that you calculated.

Related tasks:

"Selecting a method for the initial replication" on page 931

Selecting a method for the initial replication

Based on the test replication results and the total amount of data that you want to replicate, determine a method to use for the initial replication.

Related tasks:

"Calculating the time that is required for replication" on page 930

"Tuning replication processing" on page 965

Method 1: Exporting and importing data by using node replication synchronization

Use this method if you have a large amount of data to replicate and you cannot wait for initial replication to complete. During replication, the data that is exported from the source replication server and imported to the target replication server is synchronized, and the data is displayed as if it is replicated.

Procedure

1. Export the data from the source replication server for the nodes that you want to replicate. For example, to export client node information and all client files for NODE1 directly to SERVERB, issue the following command:

   ```
   export node node1 filedata=all toserver=serverb
   ```

   Tip: You can also export client data directly to another server so that it can be immediately imported. For example, to export client node information and all client files for NODE1 directly to SERVERB, issue the following command:

   ```
   export node node1 filedata=all toserver=serverb
   ```

   The data is automatically imported when it is exported to another server.
2. Import the data to the target replication server. For example, to import client node information from tape volumes TAPE01, TAPE02, and TAPE03, issue the following command:

   ```
   import node volumenames=tape01,tape02,tape03 dates=absolute
   ```

   Requirement: Ensure that you use the **DATES=ABSOLUTE** parameter on the **IMPORT NODE** command when you import the data to the target replication server. The ABSOLUTE value sets the insertion dates for the objects on the target replication server to be the same as the insertion dates on the source replication server. When synchronization occurs, the servers use the insertion dates to identify files that are the same and updates them as replicated files.
3. Configure the nodes on the source and target replication servers. Issue the **UPDATE NODE** command and specify the following parameters:
   - On the source replication server: **REPLMODE=SYNCSEND**
   - On the target replication server: **REPLMODE=SYNCRECEIVE**
4. When replication is started, the nodes are configured for synchronization and no data is transferred to the server. The source and target replication servers exchange information about the data and add entries to the database so that the data is managed as it is replicated. After the data synchronization is complete, the nodes are automatically configured by the Tivoli Storage Manager server to replicate data daily. When synchronization is complete on the source replication server, the parameter is automatically set to **REPLMODE=SEND**. When synchronization is complete on the target replication server, the parameter is automatically set to **REPLMODE=RECEIVE**.

Method 2: Replicating subsets of nodes

Use this method to replicate subsets of nodes incrementally, based on specific characteristics of the data of the nodes. You can use any characteristic to create subsets of nodes. You can use this method if you have time to complete initial replication but the network cannot manage replicating all nodes at one time.

About this task

When you decide how many nodes to add to a group, consider the amount of data that is replicated daily by the nodes.

Procedure

1. Prioritize the subset of nodes that have critical data. Replicate critical data first, by issuing the **REPLICATE NODE** command.
2. Continue to replicate the high-priority nodes daily while incrementally adding the replication of other subsets of nodes that contain important, but not critical, data.
3. Repeat this process until all subsets of all nodes that must be replicated complete their initial replication.

Related concepts:

"Node replication processing" on page 911

"Replication rules" on page 911

Method 3: Replicating active data first

Use this method to set up an initial replication of only active data to avoid overloading the network or servers if you replicate both active and inactive data. Active versions of backup data are more important than the inactive versions.

Procedure

1. Configure the replication rules to replicate active data.
2. Replicate the active data. For example, issue the following **UPDATE NODE** command to replicate active data for a node named PAYROLL:

   ```
   update node payroll bkreplrule=active_data
   ```

3. After the initial replication of the active data is completed, configure the replication rules to replicate all versions of the data by issuing the following **UPDATE NODE** command:

   ```
   update node payroll bkreplrule=active_data_high_priority
   ```

Results

During the next scheduled replication, any new active versions, including all inactive versions, are replicated. The files that were active but are now inactive are not replicated again.

Method 4: Replicating all nodes

Use this method if you determine you have time for the initial replication process to complete, and you can configure the nodes and begin or schedule a replication.

Procedure

1. Configure all nodes that must be replicated by issuing the **UPDATE NODE** command and specify the **REPLSTATE=ENABLED** parameter.
2. Issue the **REPLICATE NODE** command to start replication processing.
3. Monitor the progress of the replication by issuing the **QUERY PROCESS** command. Summary information is displayed, which includes the amount of data that is replicated and the process duration.

   Remember: If you do not have time to complete replication, you can cancel it after it has started, by issuing the **CANCEL REPLICATION** command.
4. Use the summary information to determine whether the values of the controlled test match the actual replication values. You calculate the values of the controlled test in "Tuning replication processing" on page 965. For example, to display information about a replication process 23, issue the following command:

   ```
   query process 23
   ```

   The following summary information is displayed:

```
Process   Process              Status
Number    Description
--------  --------------       ---------
     23  REPLICATE NODE        Replicating Node(s) IRONMAN.
                                File spaces complete: 0.
                                File spaces examining and replicating: 1.
                                File spaces replicating: 0.
                                File spaces not started: 3.
                                Files current: 11,920.
                                Files replicated: 0 of 0.
                                Files updated: 0 of 0.
                                Files deleted: 0 of 0.
                                Bytes Replicated: 0 of 0.
                                Elapsed Time: 0 Day(s),
                                0 Hour(s), 1 Minute(s).
```

Scheduling incremental replication after the initial replication

After you complete the initial replication, schedule subsequent incremental replications frequently. Scheduling replication frequently ensures that the data on the target replication server is maintained at an acceptable recovery point that you specify in your data recovery plan. Daily incremental replications typically do not require as much time to complete as the initial replication.

About this task

If you are unable to complete the replication process in the amount of time that you scheduled, increase the number of data sessions that transfer data to the target replication server. Replication performance improves when more deduplicated data is stored on the target replication server. When more extents are stored on the target replication server, more duplicates are found for an extent.

Procedure

If you are replicating data from storage pools that are enabled for data deduplication, run processes in the following order:

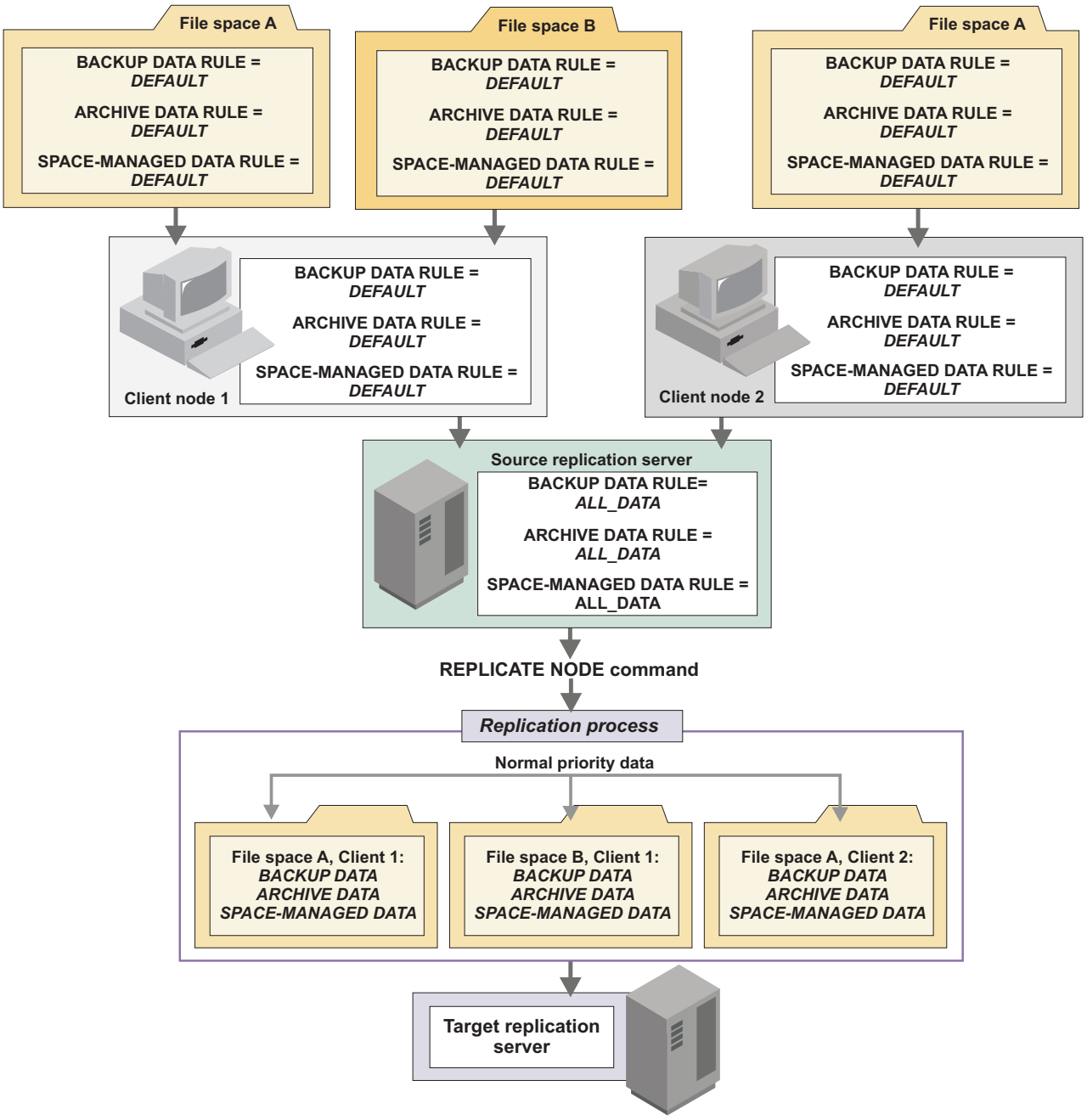
1. To identify duplicates, issue the **IDENTIFY DUPLICATES** command. Break files into extents to reduce the amount of data that is sent to the target replication server when replication occurs.
2. To start replication processing, issue the **REPLICATE NODE** command. Only file extents that do not exist on the target replication server are sent during replication, which reduces the required bandwidth and improves performance.
3. To start reclamation processing, issue the **RECLAIM STGPOOL** command. Reclamation removes and links duplicated extents.

Setting up the default replication configuration

The default configuration replicates all backup, archive, and space-managed data in all the file spaces in all the replication-enabled nodes.

About this task

The following figure shows the replication rules that are created in the default configuration. *Backup data* includes both active and inactive backup data.



To set up the default replication scenario, complete the following tasks:

"Step 1: Setting up server-to-server communications" on page 936
"Step 2: Specifying a target replication server" on page 938
"Step 3: Configuring client nodes for replication" on page 938

After you complete the default configuration, you can change the replication rules to meet your specific replication requirements.

Step 1: Setting up server-to-server communications

Define the source and the target replication servers to each other. When you define servers, you can also specify Secure Sockets Layer (SSL) communications.

About this task

Server definitions are required for the source replication server to communicate with the target replication server and for the target replication server to report status to the source replication server.

Before you begin this procedure, complete the following prerequisites:

- Ensure that the target replication server is accessible from the source replication server with an IP connection.
- Ensure that there is enough network bandwidth to accommodate the volume of data to be replicated. If bandwidth is insufficient and the flow of data is impeded, the ability to keep the two servers synchronized is affected.
- If you plan to use SSL communication, the SSL infrastructure must be in place on the source and target replication servers. Required SSL certificates must be in the key database file that belongs to each server.

Important: You can specify only one target replication server for a source replication server. However, you can specify one or more source replication servers for a single target replication server. Source and target replication servers must be V6.3 or later.

The method that you use to set up servers depends on whether the server definitions exist and on whether you are using the cross-define function to automatically define one server to another.

Procedure

To set up server-to-server communications, complete one of the following actions:

- If server definitions do not exist and you want to cross-define servers, complete the following steps:
  1. On the target replication server, issue the following commands:

     ```
     set servername target_server_name
     set serverpassword target_server_password
     set serverhladdress _target_server_ip_address
     set serverlladdress target_server_tcp_port
     set crossdefine on
     ```

     Remember: If you want an SSL connection, the value for the **SET SERVERLLADDRESS** command on the target replication server must be an SSL port. The value of the **SET SERVERNAME** command must match the server name in the server definition.
  2. On the source replication server, issue the following commands:

     ```
     set servername source_server_name
     set serverpassword source_server_password
     set serverhladdress source_server_ip_address
     set serverlladdress source_server_tcp_port
     ```

     Remember: If you want an SSL connection, the value for the **SET SERVERLLADDRESS** command on the source replication server must be an SSL port. The value of the **SET SERVERNAME** command must match the server name in the server definition.

3. On the source replication server connect to the target replication server by using the **DEFINE SERVER** command. If you want an SSL connection, specify SSL=YES. For example:

   ```
   define server target_server_name hladdress=target_sever_ip_address
   lladdress=target_server_tcp_port serverpassword=
   target_server_password crossdefine=yes ssl=yes
   ```

   A server definition is created on the source replication server, and the source replication server is connected to the target replication server. A definition for the target replication server is created that points to the source replication server.

- If server definitions do not exist and you are not using the cross-define function, complete the following steps:
  1. Issue the following commands on both the source and target replication servers:

     ```
     set servername server_name
     set serverpassword server_password
     set serverhladdress ip_address
     set serverlladdress tcp_port
     ```

     Remember: If you want an SSL connection, the value for the **SET SERVERLLADDRESS** command on the replication servers must be an SSL port. The value of the **SET SERVERNAME** command must match the server name in the server definition.
  2. Issue the **DEFINE SERVER** command on each server. Do not specify the **CROSSDEFINE** parameter. If you want an SSL connection, specify SSL=YES. For example:
     - On the source replication server:

       ```
       define server target_server_name hladdress=target_server_ip_address
       lladdress=target_server_tcp_port serverpassword=target_server_password
       ssl=yes
       ```

     - On the target replication server:

       ```
       define server source_server_name hladdress=source_server_ip_address
       lladdress=source_server_tcp_port serverpassword=source_server_password
       ssl=yes
       ```

- If definitions exist for both the source and target replication servers, issue the **UPDATE SERVER** command on each server. Do not specify the **CROSSDEFINE** parameter. You can use the **QUERY STATUS** command to determine the server names. If you want an SSL connection, specify SSL=YES. For example:
  - On the source replication server:

    ```
    update server target_server_name hla=target_server_ip_address
    lladdress=target_server_tcp_port serverpassword=target_server_password
    ssl=yes
    ```

  - On the target replication server:

    ```
    update server source_server_name hladdress=source_server_ip_address
    lladdress=source_server_tcp_port serverpassword=
    source_server_password
    ssl=yes
    ```

Step 2: Specifying a target replication server

For each source replication server, you can specify only a single target replication server. Multiple source replication servers can share the same target replication server.

Before you begin

Before you begin this procedure, issue the **PING SERVER** command. The command verifies that the definitions for the source and target replication servers are valid and that the servers are connected.

Procedure

To specify a target replication server, issue the **SET REPLSERVER** command on the source replication server. For example, to specify a server named PHOENIX_SRV as the target replication server, issue the following command:

```
set replserver phoenix_srv
```

Results

Issuing the **SET REPLSERVER** command also sets replication rules to ALL_DATA. To display replication rules, you can issue the **QUERY STATUS** command.

Related concepts:

"Replication server configurations" on page 909

Step 3: Configuring client nodes for replication

When a node is configured for replication for the first time, it cannot exist on the target replication server unless the data of the node was exported from the source replication server and imported on the target replication server.

Before you begin

Before you begin this procedure:

- Ensure that the source replication server and target replication server are set up for server-to-server communications and that they can communicate with each other.
- Ensure that a target replication server was specified.
- If the data for a client node was exported from a source replication server and imported to the target replication server, you must synchronize the data. To synchronize data, the data must be imported with the value of the **DATES** parameter on the **IMPORT NODE** command that is set to ABSOLUTE. The source replication server uses the file names and the insertion dates that files were imported to identify identical files.

Restrictions:

- If a client node definition does not exist on the target replication server, do not create it. The definition for the client node on the target replication server is created automatically when the data of the node is replicated the first time.
- If a client node definition exists on both the source and target replication servers, but the data that belongs to the client node was not exported and imported, you must rename or remove the client node on the target replication server before data can be replicated.

- If you previously removed a client node from replication on the source replication server, but not on the target replication server, you do not have to rename or remove the node on the target replication server.

About this task

To configure a client node for replication, take one of the following actions, depending on whether data of the node was exported from the source replication server and imported on the target replication server:

Procedure

- If the data of the node was not exported from the source replication server and imported on the target replication server, complete one of the following steps:
  - If the client node is not already registered on a source replication server, issue the **REGISTER NODE** command on the source replication server. Specify REPLSTATE=ENABLED or REPLSTATE=DISABLED.

    For example, to enable a new client node, NODE1, for replication, issue the following command:

    ```
    register node node1 password replstate=enabled
    ```

  - If the client node is already registered on a source replication server, issue the **UPDATE NODE** command on the source replication server. Specify REPLSTATE=ENABLED or REPLSTATE=DISABLED.

    For example, to enable an existing client node, NODE1, for replication, issue the following command:

    ```
    update node node1 replstate=enabled
    ```

- If the data of the node was exported from the source replication server and imported to the target replication server, complete the following steps:
  1. On the source replication server, issue the **UPDATE NODE** command:
     a. Specify REPLSTATE=ENABLED or REPLSTATE=DISABLED.
     b. Specify REPLMODE=SYNCSEND.
  2. On the target replication server, issue the **UPDATE NODE** command and specify REPLMODE=SYNCRECEIVE.

  Data is synchronized during replication. After the replication process is complete, the **REPLMODE** parameter in the client node definition on the source replication server is set to SEND. The **REPLMODE** parameter in the client node definition on the target replication server is set to RECEIVE, and the **REPLSTATE** parameter is set to ENABLED.

Results

If you set the replication state of the client node to DISABLED, the replication mode is set to SEND, but replication does not occur. If you set the replication state of the client node to ENABLED, the client node definition is created on the target replication server when replication occurs for the first time. In addition, the replication mode of the client node on the target replication server is set to RECEIVE, and the replication state is set to ENABLED.

If you add a file space to a client node that is configured for replication, the file space replication rules for data types are automatically set to DEFAULT. To change file space replication rules, issue the **UPDATE FILESPACE** command.

To determine the replication mode and the replication state that a client node is in, issue the **QUERY NODE** command.

What to do next

The default configuration is complete after client nodes are configured for replication. You are now ready to replicate. If you do not change the default replication rules, all backup, archive, and space-managed data in all replication-enabled client nodes is replicated.

Related concepts:

"Replication mode" on page 918

"Replication state" on page 915

Customizing a node replication configuration

With Tivoli Storage Manager, you can change the replication configuration, including the type and priority of data that is replicated.

Changing replication rules

You can change replication rules that apply to data types in individual file spaces, to individual client nodes, and collectively to all client nodes that are configured for replication.

Changing replication rules for individual file spaces

When replication processing begins, the source server first checks the replication rules that apply to data types in file spaces. File space rules take precedence over client node rules and server rules.

About this task

Rules for file spaces are either normal priority or high priority. In a replication process that includes both normal-priority and high-priority data, high-priority data is replicated first. If you issue the **REPLICATE NODE** command for two or more clients, all high priority data for all file spaces in the specified nodes is processed before normal priority data.

Before you select a rule, consider the order in which you want the data to be replicated. For example, suppose that a file space contains active backup data and archive data. Replication of the active backup data is a higher priority than the archive data. To prioritize the active backup data, specify `DATATYPE=BACKUP REPLRULE=ACTIVE_DATA_HIGH_PRIORITY`. To prioritize the archive data, issue the **UPDATE FILESPACE** command again, and specify `DATATYPE=ARCHIVE REPLRULE=ALL_DATA`.

You can specify the following rules:

ALL_DATA
: Replicates backup, archive, or space-managed data. The data is replicated with a normal priority.

ACTIVE_DATA
: Replicates only the active backup data. The data is replicated with a normal priority.

Attention:

- If you specify ACTIVE_DATA, inactive backup data in the file space is not replicated, and inactive backup data in the file space on the target replication server is deleted.
- If you specify ACTIVE_DATA, you cannot specify ARCHIVE or SPACEMANAGED as values for the parameter **DATATYPE** in the same command instance.

ALL_DATA_HIGH_PRIORITY
Replicates backup, archive, or space-managed data. The data is replicated with a high priority.

ACTIVE_DATA_HIGH_PRIORITY
Replicates active backup data. The data is replicated with a high priority.

Attention:

- If you specify ACTIVE_DATA_HIGH_PRIORITY, inactive backup data in the file space is not replicated, and inactive backup data in the file space on the target replication server is deleted.
- If you specify ACTIVE_DATA_HIGH_PRIORITY, you cannot specify ARCHIVE or SPACEMANAGED as values for the parameter **DATATYPE** in the same command instance.

DEFAULT
Replicates data according to the client node rule for the data type.

For example, suppose that you want to replicate the archive data in all the file spaces that belong to a client node. Replication of the archive data is a high priority. One method to accomplish this task is to specify `DATATYPE=ARCHIVE REPLRULE=DEFAULT` for each file space. Ensure that the client replication rule for archive data is set to ALL_DATA_HIGH_PRIORITY or to DEFAULT. If the client replication rule is DEFAULT, the server replication rule for archive data must be set to ALL_DATA_HIGH_PRIORITY.

NONE
Data is not replicated.

For example, if you do not want to replicate the space-managed data in a file space, specify NONE.

To display the replication rules for a file space, issue the **QUERY FILESPACE** command. Specify `FORMAT=DETAILED`.

To change file-space replication rules, issue one or more of the following commands on the source replication server:

Procedure

- To change a replication rule for backup data, issue the **UPDATE FILESPACE** command and specify `DATATYPE=BACKUP`. Specify the **REPLRULE** parameter. For example, to specify the ACTIVE_DATA rule for backup data in file space `/a` on NODE1, issue the following command:

  ```
  update filespace node1 /a datatype=backup replrule=active_data
  ```

- To specify a replication rule for archive data, issue the **UPDATE FILESPACE** command and specify `DATATYPE=ARCHIVE`. Specify the **REPLRULE** parameter. For example, to specify the ALL_DATA_HIGH_PRIORITY rule for archive data in file space `/a` on NODE1, issue the following command:

```
update filespace node1 /a datatype=archive replrule=all_data_high_priority
```

- To specify a replication rule for space-managed data, issue the **UPDATE FILESPACE** command and specify DATATYPE=SPACEMANAGED. Specify the **REPLRULE** parameter. For example, to specify the NONE rule for file space /a on NODE1, issue the following command:

```
update filespace node1 /a datatype=spacemanaged replrule=none
```

Example

In the following example, assume that you have two client nodes, NODE1 and NODE2. The nodes have the following file spaces:

- NODE1: /a, /b, /c
- NODE2: /a, /b, /c, /d, /e

All the file space rules are set to DEFAULT. The backup, archive, and space-managed rules for NODE1 and NODE2 are also set to DEFAULT. The server rules are set to ALL_DATA. To replicate the backup data in file space /a in NODE1 and file space /c in NODE2 first, before replicating data in the other file spaces, issue the following commands:

```
update filespace node1 /a datatype=backup replrule=all_data_high_priority
update filespace node2 /c datatype=backup replrule=all_data_high_priority
```

The data that belongs to the two nodes is replicated in the following order:

1. High Priority: Data in file space /a that belongs to NODE1 and data in file space /c in NODE2
2. Normal priority: Data in file spaces /b and /c that belongs to NODE1 and data in file spaces /a, /b, /d, and /e that belongs to NODE2

What to do next

Important: Data types in new file spaces that are added to a client node after the node is configured for replication are automatically assigned the DEFAULT replication rule.

Related concepts:

"Replication rules" on page 911

Changing replication rules for individual client nodes

Client-node replication rules apply to every file space that belongs to a client node. To replicate data according to a client node rule for a data type, the replication rule that applies to the data type in a file space must be set to DEFAULT.

About this task

Rules for client nodes are either normal priority or high priority. In a replication process that includes both normal-priority and high-priority data, high-priority data is replicated first. If you issue the **REPLICATE NODE** command for two or more clients, all high priority data for all file spaces in the specified nodes is processed before normal priority data.

Before you select a rule, consider the order in which you want the data to be replicated. For example, suppose that a client node contains active backup data and archive data. Replication of the active backup data is a higher priority than

replication of the archive data. To prioritize the active backup data, specify the ACTIVE_DATA_HIGH_PRIORITY replication rule for backup data. Specify the ALL_DATA rule for archive data.

You can specify the following rules:

ALL_DATA
Replicates backup, archive, or space-managed data. The data is replicated with a normal priority.

ACTIVE_DATA
Replicates only the active backup data. The data is replicated with a normal priority.

Attention:

- If you specify ACTIVE_DATA, inactive backup data that belongs to the client node is not replicated.
- If the replication rule for backup data in any file spaces that belong to the client node is DEFAULT, inactive backup data in those file spaces on the target replication server is deleted.

ALL_DATA_HIGH_PRIORITY
Replicates backup, archive, or space-managed data. The data is replicated with a high priority.

ACTIVE_DATA_HIGH_PRIORITY
Replicates only the active backup data in client nodes. The data is replicated with a high priority.

Attention:

- If you specify ACTIVE_DATA_HIGH_PRIORITY, inactive backup data that belongs to the client node is not replicated.
- If the replication rule for backup data in any file spaces that belong to the client node is DEFAULT, inactive backup data in those file spaces on the target replication server is deleted.

DEFAULT
Replicates data according to the server rule for the data type.

For example, suppose that you want to replicate the archive data in all client nodes that are configured for replication. Replication of the archive data is a high priority. One method to accomplish this task is to set the file-space and client-node replication rules for archive data to DEFAULT. Set the server rule for archive data to ALL_DATA_HIGH_PRIORITY.

NONE
Data is not replicated. For example, if you do not want to replicate the space-managed data in a client node, specify the NONE replication rule for space-managed data.

To display the replication rules that apply to all file spaces that belong to a node, issue the **QUERY NODE** command and specify `FORMAT=DETAILED`.

Remember: File spaces are not displayed for client nodes that are registered on the source replication server but that have not completed store operations. File spaces are created only after the client stores data to the source replication server. Replication rules for data types in file spaces are automatically assigned values of DEFAULT.

To change replication rules for a node, issue one or more of the following commands on the source replication server:

Procedure

- To change a replication rule for backup data, issue the **UPDATE NODE** command and specify the **BKREPLRULEDEFAULT** parameter. For example, to specify the ACTIVE_DATA rule for backup data in NODE1, issue the following command:

  ```
  update node node1 bkreplruledefault=active_data
  ```

- To change a replication rule for archive data, issue the **UPDATE NODE** command and specify the **ARREPLRULEDEFAULT** parameter. For example, to specify the ALL_DATA_HIGH_PRIORITY rule for archive data in NODE1, issue the following command:

  ```
  update node node1 arreplruledefault=all_data_high_priority
  ```

- To change a replication rule for space-managed data, issue the **UPDATE NODE** command and specify the **SPREPLRULEDEFAULT** parameter. For example, to specify the NONE rule for space-managed data in NODE1, issue the following command:

  ```
  update node node1 spreplruledefault=none
  ```

Related concepts:

"Replication rules" on page 911

Changing server replication rules

Server replication rules apply to all client nodes that are defined to a source replication server and that are configured for replication. To replicate data according to a server rule for a data type, the replication rule that applies to the data type in client nodes and file spaces must be set to DEFAULT.

About this task

Server rules are either normal priority or high priority. In a replication process that includes both normal-priority and high-priority data, high-priority data is replicated first. If you issue the **REPLICATE NODE** command for two or more clients, all high priority data for all file spaces in the specified nodes is processed before normal priority data.

Before you select a rule, consider the order in which you want the data to be replicated. For example, suppose that your client nodes contain active backup data and archive data. Replication of the active backup data is a high priority. To prioritize the active backup data, specify the ACTIVE_DATA_HIGH_PRIORITY replication rule. Specify the ALL_DATA rule for archive data.

You can specify the following rules:

ALL_DATA
: Replicates backup, archive, or space-managed data. The data is replicated with a normal priority.

ACTIVE_DATA
: Replicates only the active backup data. The data is replicated with a normal priority.

Attention:

- If you specify ACTIVE_DATA, inactive backup data that belongs to client nodes is not replicated.
- If the replication rules for backup data in any file spaces and any client nodes is DEFAULT, inactive backup data in those file spaces on the target replication server is deleted. For example, suppose the rules for backup data in file space /a in NODE1 and file space /c in NODE2 are DEFAULT. The rules for backup data in NODE1 and NODE2 are also DEFAULT. If you specify ACTIVE_DATA as the server rule, inactive data in file spaces /a and /c is deleted.

ALL_DATA_HIGH_PRIORITY
Replicates backup, archive, or space-managed data. The data is replicated with a high priority.

ACTIVE_DATA_HIGH_PRIORITY
Replicates only the active backup data in client nodes. The data is replicated with a high priority.

Attention:

- If you specify ACTIVE_DATA_HIGH_PRIORITY, inactive backup data that belongs to client nodes is not replicated.
- If the replication rules for backup data in any file spaces and any client nodes is DEFAULT, inactive backup data in those file spaces on the target replication server is deleted. For example, suppose the rules for backup data in file space /a in NODE1 and file space /c in NODE2 are DEFAULT. The rules for backup data in NODE1 and NODE2 are also DEFAULT. If you specify ACTIVE_DATA as the server rule, inactive data in file spaces /a and /c is deleted.

NONE
Data is not replicated. For example, if you do not want to replicate the space-managed data in your client nodes, specify NONE.

Restriction: The replication rule DEFAULT is not a valid server rule.

To change server replication rules, issue one or more of the following commands on the source replication server:

Procedure

- To change the server replication rule that applies to backup data, issue the **SET BKREPLRULEDEFAULT** command on the source replication server. For example, to specify the ACTIVE_DATA rule for backup data, issue the following command:

  ```
  set bkreplruledefault active_data
  ```

- To change the server replication rule that applies to archive data, issue the **SET ARREPLRULEDEFAULT** command on the source replication server. For example, to specify the ALL_DATA_HIGH_PRIORITY rule for archive data, issue the following command:

  ```
  set arreplruledefault all_data_high_priority
  ```

- To change the server replication rule that applies to space-managed data, issue the **SET SPREPLRULEDEFAULT** command on the source replication server. For example, to specify the NONE rule for space-managed data, issue the following command:

  ```
  set spreplruledefault none
  ```

Related concepts:

"Replication rules" on page 911

Scenario: Customizing the default replication configuration

You set up the default replication configuration for two client nodes, NODE1 and NODE2. You decide that you want to replicate only certain types of data in their file spaces and that you want to replicate some data types before other data types.

About this task

NODE1 has two file spaces, `/a` and `/b`. NODE2 has one file space, `/a`. File space and client replication rules for backup, archive, and space-managed data are set to DEFAULT. Server replication rules are set to ALL_DATA. You have the following goals:

- Replicate only the active backup data in file space `/a` that belongs to NODE1.
- Do not replicate any space-managed data in any of the file spaces that belong to NODE1.
- Replicate the archive data in all file spaces that belong to NODE1 and NODE2. Make the replication of the data a high priority.
- Replicate the active and inactive backup data in file space `/a` that belongs to NODE2. Make replication of the data a high priority.

To accomplish these goals, replication rules must be set as shown in the following figure:

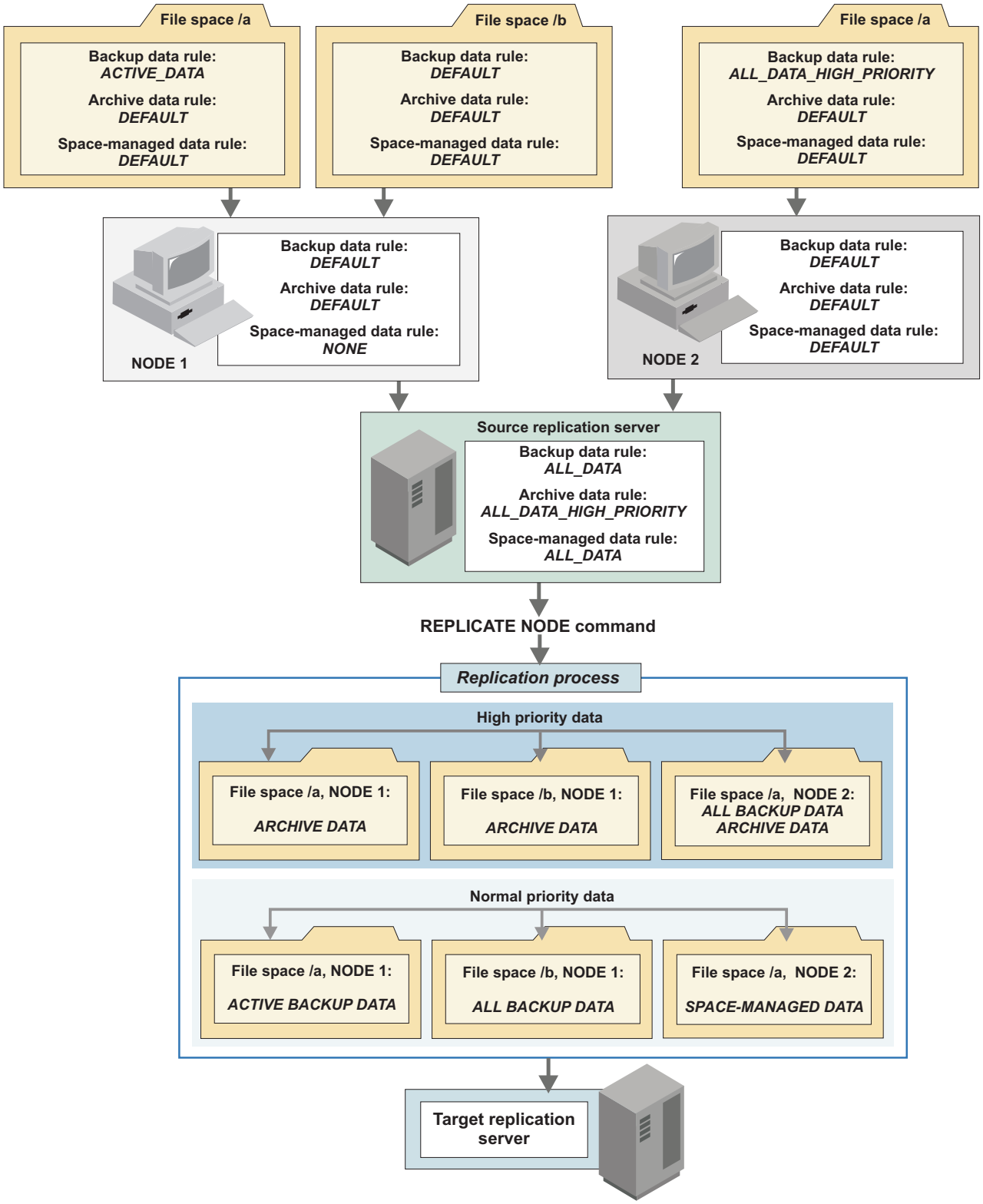


Figure 113. Customized replication configuration

To change to the configuration illustrated in the figure, complete the following steps:

Procedure

- To change the server replication rule for archive data, issue the **SET ARREPLRULEDEFAULT** command on the source replication server:

  ```
  set arreplruledefault all_data_high_priority
  ```

- To change the replication rule for space-managed data for NODE1, issue the **UPDATE NODE** command:

  ```
  update node node1 spreplruledefault=none
  ```

- To change the replication rules for backup data for file spaces that belong to NODE1 and NODE2, issue the **UPDATE FILESPACE** command:

  ```
  update filespace /a node1 fsreplrule=bkreplrule replrule=
     active_data
  update filespace /a node2 fsreplrule=bkreplrule replrule=
     all_data_high_priority
  ```

Results

When the **REPLICATE NODE** command is issued from a command line or in a maintenance script, high-priority data is replicated first. High-priority data includes archive data that belongs to NODE1 and NODE2 and backup data that belongs to NODE2. No preference is given to which node's data is replicated first.

After high-priority data is replicated, normal-priority data is replicated. Normal-priority data includes the active data in file space `/a` and the backup data in file space `/b` that belong to NODE1. The backup data and space-managed data that belongs to NODE2 is replicated along with the normal-priority data that belongs to NODE1. No preference is given to which node's data is replicated first.

Tips:

- In Figure 113 on page 947, all the data in all the files spaces of both client nodes is replicated in one process. However, if the amount of node data is large and you do not have enough bandwidth to replicate data in a single process, you can use one of the following methods:
  - Schedule or manually issue separate **REPLICATE NODE** commands at different times for NODE1 and NODE2.
  - Replicate high-priority and normal-priority data separately at different times by specifying the **PRIORITY** parameter on the **REPLICATE NODE** command.
  - Replicate different data types at different times by specifying the **DATATYPE** parameter on the **REPLICATE NODE** command.
  - Combine replication by priority and by data type by specifying both the **PRIORITY** and **DATATYPE** parameters on the **REPLICATE NODE** command.
- To verify the replication rules that apply to the file spaces in the client nodes, issue the **VALIDATE REPLICATION** command. You can also use this command to verify that the source replication server can communicate with the target replication server. To preview results, issue the **REPLICATE NODE** command and specify `PREVIEW=YES`.

Related concepts:

"Replication rules" on page 911

Scenario: Converting to node replication from import and export operations

You are exporting client node data from one or more servers and importing the data to another server for disaster recovery. You want to achieve the same results by replicating data. In this scenario, the server that was exporting data is to be the source replication server. The server where the data was imported is to be the target replication server.

Before you begin

Client node data that was exported and imported must be synchronized between the source and target replication servers. You set up client nodes to synchronize their data as part of the process of configuring nodes for replication. Data is synchronized the first time that replication occurs. To synchronize data, the data must be imported to the disaster recovery server by using ABSOLUTE as the value for the **DATES** parameter on the **IMPORT NODE** command.

Procedure

To convert from import and export operations to replication operations, complete the following steps:

1. Set up server definitions for the source and target replication servers: "Step 1: Setting up server-to-server communications" on page 936
2. Specify the server to which you are importing data as the target replication server: "Step 2: Specifying a target replication server" on page 938
3. On the source replication server, issue the **UPDATE NODE** command:
   a. Specify `REPLSTATE=ENABLED` or `REPLSTATE=DISABLED`. Disabling replication means that replication does not occur until you enable it.
   b. Specify `REPLMODE=SYNCSEND`.

   To update the definitions for all the nodes that are defined to the source replication server, issue the following command.

   ```
   update node * replstate=enabled replmode=syncsend
   ```
4. On the target replication server, issue the **UPDATE NODE** command and specify `REPLMODE=SYNCRECEIVE` and `REPLSTATE=ENABLED`:

   ```
   update node * replmode=syncreceive replstate=enabled
   ```
5. If necessary, modify the file-space, client-node, and server replication rules: "Changing replication rules" on page 940.
6. Replicate the data that belongs to your nodes: "Replicating data by command" on page 959

   After all data is synchronized, the **REPLMODE** parameter in the client node definition on the source replication server is set to SEND. The **REPLMODE** parameter in the client node definition on the target replication server is set to RECEIVE.

What to do next

Important: You cannot display information about running replication processes for client nodes that are being converted from import and export operations to replication operations. The conversion process might run for a long time, but it occurs only once for a client node that is being converted.

After you set up a basic replication configuration, you can change file space, client-node, and server replication rules. To replicate data, issue the **REPLICATE NODE** command in an administrative schedule or on a command line.

Related concepts:

"Replication mode" on page 918

"Replication state" on page 915

Adding and removing client nodes for replication

You can add or remove client nodes for replication at any time after you set up a replication configuration.

Adding client nodes for replication processing

If the data that belongs to a client node was previously exported from the source replication server and imported to the target replication server, you must set up the node to synchronize its data. You set up a client node to synchronize its data as part of the process of adding the node for replication.

Before you begin

Before you add a client node for replication, ask the following questions:

- Was the data that belongs to the client node previously exported from the server that is to be the source replication server for replicated data?
- If the data was exported, was it imported on the server that is now the target replication server for replicated data?
- When you imported the data, did you specify DATES=ABSOLUTE on the **IMPORT NODE** command?

If you answered "yes" to all of the preceding questions, you must set up to synchronize the data on the source and target replication servers. The following procedure explains how to set up synchronization when you add client nodes for replication. Synchronization occurs during replication.

Restrictions:

- If a client node definition does not exist on the target replication server, do not create it. The definition for the client node on the target replication server is created automatically when the data of the node is replicated the first time.
- If a client node definition exists on both the source and target replication servers, but the data that belongs to the client node was not exported and imported, you must rename or remove the client node on the target replication server before data can be replicated.
- If you previously removed a client node from replication on the source replication server, but not on the target replication server, you do not have to rename or remove the node on the target replication server.

Procedure

To add a node for replication, complete one of the following actions:

- If the data of the node was not exported from the source replication server and imported on the target replication server, complete one of the following steps:
  - If the client node is not already registered on a source replication server, issue the **REGISTER NODE** command on the source replication server. Specify REPLSTATE=ENABLED or REPLSTATE=DISABLED.

For example, to enable a new client node, NODE1, for replication, issue the following command:

```
register node node1 password replstate=enabled
```

– If the client node is already registered on a source replication server, issue the **UPDATE NODE** command on the source replication server. Specify REPLSTATE=ENABLED or REPLSTATE=DISABLED.

For example, to enable an existing client node, NODE1, for replication, issue the following command:

```
update node node1 replstate=enabled
```

- If you exported node data from the source replication server and imported the data to the target replication server or if you reconfigured replication for a node after you removed the node from replication on the source replication server, complete the following steps:
  1. On the source replication server, issue the **UPDATE NODE** command:
     a. Specify REPLSTATE=ENABLED or REPLSTATE=DISABLED
     b. Specify REPLMODE=SYNCSEND.
  2. On the target replication server, issue the **UPDATE NODE** command:
     a. Specify REPLSTATE=ENABLED or REPLSTATE=DISABLED.
     b. Specify REPLMODE=SYNCRECEIVE.

Results

If you set the replication state of the client node to DISABLED, the replication mode is set to SEND, but replication does not occur. If you set the replication state of the client node to ENABLED, the client node definition is created on the target replication server when replication occurs for the first time. In addition, the replication mode of the client node on the target replication server is set to RECEIVE, and the replication state is set to ENABLED.

If you add a file space to a client node that is configured for replication, the file space replication rules for data types are automatically set to DEFAULT.

What to do next

After you add client nodes for replication, ensure that they are included in any existing administrative schedules for replication. Alternatively, you can create a schedule for replication that includes the new client nodes.

Related concepts:

"Replication mode" on page 918

"Replication state" on page 915

Removing client nodes from replication processing

Removing a client node from replication changes the node to a non-replicated node. If you remove a client node from replication on a target replication server, the client node can back up, archive, or migrate data to the server.

About this task

Removing a client node from replication deletes only information about replication from the server database. Removing a node from replication does not delete the data that belongs to the node that was replicated.

Procedure

To completely remove a client node from replication, issue the **REMOVE REPLNODE** command on the source and target replication servers that have the node that is configured for replication. For example, to remove NODE1 and NODE2 from replication, issue the following command:

```
remove replnode node1,node2
```

What to do next

To verify that the node was removed, issue the **QUERY NODE** command on the source and the target replication servers. For example, to verify that NODE1 and NODE2 were removed, issue the following command:

```
query node node1,node2 format=detailed
```

If the node was removed, the fields `Replication State` and `Replication Mode` are blank. If you do not want to keep the node data that is stored on the target replication server, you can delete it using the **DELETE FILESPACE** command for the file spaces that belong to the node. If you do not want to keep the node definition on the target replication server, delete the node definition by issuing the **REMOVE NODE** command.

If you remove a client node from replication, rename the node, or delete the node data, and then remove the node. You can add the node for replication later. All the node data is replicated to the target replication server.

Resetting the replication mode

If you set the replication mode incorrectly when you configure a client node with data that was exported and imported, you must remove the client node from replication before reconfiguring the node.

About this task

For example, suppose that you updated the definition of a client node whose data you wanted to replicate. The data that belongs to the node was previously exported from the source replication server and imported to the target replication server. You specified ENABLED as the setting of the **REPLSTATE** parameter. However, you did not specify SYNCSEND as the replication mode on the source replication server. As a result, the **REPLMODE** parameter was automatically set to SEND, and data that belongs to the node was not synchronized or replicated.

Procedure

To reconfigure the client node for replication, complete the following steps:

1. Issue the **REMOVE REPLNODE** command for the client node. For example, to remove a client node, NODE1, from replication, issue the following command:

   ```
   remove replnode node1
   ```

   Issuing the **REMOVE REPLNODE** command resets the replication state and the replication mode for the client node to NONE.
2. Issue the UPDATE NODE command with the correct parameters and values. For example, to enable NODE1 for replication and synchronize the data that belongs to the node, complete the following steps:
   a. On the source replication server, issue the following command:

      ```
      update node node1 replstate=enabled replmode=syncsend
      ```

b. On the target replication server, issue the following command:

```
update node node1 replstate=enabled replmode=syncreceive
```

Results

After synchronization and replication are complete, the **REPLMODE** parameter in the client node definition on the source replication server is set to SEND. The **REPLMODE** parameter in the client node definition on the target replication server is set to RECEIVE.

Related concepts:

"Replication mode" on page 918

"Replication state" on page 915

Managing source and target replication servers

You can change or remove target replication servers. You can also protect communications during node replication operations with Secure Sockets Layer.

Adding a source replication server

To add a source replication server, you follow the same procedure that you followed when you set up the default replication configuration. As part of the procedure, you set up server-to-server communications, specify a target replication server, and configure client nodes for replication.

About this task

You can add a source replication server to an existing configuration. For example, suppose that you have a replication configuration comprising a single source replication server and a single target replication server. You can add another source server that replicates data to the existing target server.

Related concepts:

"Replication server configurations" on page 909

Selecting a new target replication server

After you set up a basic replication configuration, you can change the target replication server.

Before you begin

Before you begin this procedure:

- Ensure that the source and the target replication servers have server definitions for each other. The following properties must be set in the definitions:
  - Server name
  - Server password
  - Server high-level address
  - Server low-level address
- Delete any administrative schedules on the source replication server that specify the target replication server. Create new schedules that specify the new target replication server.

Procedure

To change a target replication server, issue the **SET REPLSERVER** command on the source replication server. Specify the name of the new target replication server. For example, to specify NEW_TGTSRV as the new target replication server, issue the following command:

```
set replserver new_tgtsrv
```

Results

If you change target replication servers during replication processing, replication to the old target replication server continues until the processing ends or is canceled. When replication occurs for the first time with the new target replication server, files are replicated again.

The following example describes what occurs when you change or add target replication servers. Suppose that DRSERVER is the target replication server for PRODSERVER. PRODSERVER has one client, NODE1.

1. Files A, B, and C that belong to NODE1 are replicated to TGTSRV.
2. You change the target replication server to NEW_TGTSRV.
3. NODE1 backs up files D, E, and F to SRCSRV.
4. Replication occurs for NODE 1. Files, A, B, and C, which were replicated to TGTSRV, are replicated to NEW_TGTSRV. New files D, E, and F are also replicated to NEW_TGTSRV.
5. You reset the target replication server to TGTSRV.
6. Replication occurs for NODE1. Files D, E, and F are replicated to TGTSRV.

Files A, B, C, D, E, and F now exist on SRCSRV, TGTSRV, and NEW_TGTSRV.

Related concepts:

"Replication server configurations" on page 909

Configuring the target replication server

You can configure the target replication server for automatic failover for data recovery. It is only necessary to configure for automatic failover if you use separate, dedicated networks for server-to-server communication and client access.

Procedure

1. Determine the default replication server by using **QUERY STATUS** command. For example, issue the following command from the target replication server to determine the default replication server:

   ```
   QUERY STATUS
   ```

2. Identify the high-level address that is used by the source replication server for replication. For example, issue the following command from the source replication server:

   ```
   QUERY SERVER
   ```

3. Identify the high-level address that the client uses to connect to the source replication server. For example, issue the following command from the target replication server:

   ```
   QUERY SERVER
   ```

4. If the source replication server and the client use a different high-level address, issue the **SET FAILOVERHLADDRESS** command on the target replication server. For example, to set the high-level address for the target replication server, issue the following command:

   ```
   SET FAILOVERHladdress high_level_address
   ```

   Restriction: You can specify only one failover server for each node target replication server. The failover server information is stored in the client options file.

Removing a replication server

If you do not want to replicate data from a source replication server to its target replication server, you can switch or remove a target replication server in a replication process.

About this task

You use the **REMOVE REPLSERVER** command to remove a server from the replication environment. You must issue the command on both the source and the target replication servers. If you use this command on only one of the replication servers, the data is not replicated.

Procedure

1. Identify all nodes that replicated data on the target server that is being removed. On the source replication server, issue the **QUERY REPLNODE** command.
2. On the source replication server, issue the **QUERY REPLSERVER** command to identify the GUID for the target server you want to remove.
3. Take note of the GUID for the source server. You specify the GUID later in the procedure.
4. On the source replication server, issue the **REMOVE REPLSERVER** command, and specify the GUID for the target server. For example, to remove a replication server and all replication state information that is associated with a server, issue the following command:

   ```
   remove replserver 11.9c.54.e0.8a.b5.11.d6.b3.c3.00.06.29.45.c1.5b
   ```
5. Remove all of the replication state information for data that is replicated from the source server. On the target replication server, issue the **REMOVE REPLSERVER** command, and specify the GUID for the source server.
6. Remove all of the nodes that were replicated from the source server. On the target replication server, issue the **REMOVE REPLNODE** command.

Results

The node definitions on the source server that were replicated to the specified target server are in the replication SEND mode. You can continue to replicate these nodes to other replication servers by using the **SET REPLSERVER** command to modify the default replication server value.

Scenario: Automatic failover

You are replicating client node data from a source replication server to a target server. You want to ensure that client restore operations can automatically fail over to the target replication server, if the source replication server is unavailable.

In the following scenario, replication operations are set up to automatically fail over from the source replication server to the target replication server:

1. A Tivoli Storage Manager V7.1 client connects to a Tivoli Storage Manager V7.1 source replication server. The source replication server provides the connection information for the target replication server to the client.
2. If the source replication server is unavailable, the client uses the connection information for the target replication server and logs on to the target replication server.
3. If you replicated client node data to the target replication server, the client can restore the data. The client cannot store data to the target replication server.

Configuring a server for SSL communications

You can configure source and target replication servers to use Secure Sockets Layer (SSL) instead of TCP/IP.

Before you begin

A server that uses SSL can obtain a unique certificate that is signed by a certificate authority (CA), or the server can use a self-signed certificate. Before you start the source and target replication servers, install the certificates and add them to the key database files. Required SSL certificates must be in the key database file that belongs to each server. SSL support is active if the server options file contains the `SSLTCPPORT` or `SSLTCPADMINPORT` option or if a server is defined with **SSL=YES** at startup.

If the client is configured for SSL communication, the client uses the SSL port to communicate with the target server during failover.

Because the certificate is provided by a trusted authority, it is accepted by the server and, therefore, communication between servers can begin. You can query a server-defined password by issuing the **QUERY SSLKEYRINGPW** command. Because the server uses random characters, it is better to update the password by issuing the following command:

```
SET SSLKEYRINGPW newpw UPDATE=Y
```

The server and its database are updated with the new password. After you update the password, shut down the server, add the certificates, and start the server.

About this task

To determine whether a server uses SSL, issue the **QUERY SERVER** command.

Procedure

To update a server definition for SSL, issue the **UPDATE SERVER** command. For example, to update the server definition for server PHOENIX_SRV, issue the following command:

```
update server phoenix_srv ssl=yes
```

What to do next

Restriction: For event servers, library servers, and target replication servers, the name of the source replication server must match the value of the **SET SERVERNAME** command on the target server. Because the source replication server uses the name of the target replication server for SSL communications, the name is not available for the TCP/IP server definition that is required for enterprise events or library sharing.

If you enable SSL communications and are using the following functions, you must create separate source and target definitions that use TCP/IP for the corresponding server-to-server communications:

- Enterprise configuration
- Command routing
- Virtual volumes
- LAN-free

Configuring SSL-enabled servers for non-SSL communications:

If you use SSL with node replication, you must create separate server definitions for enterprise configuration, command routing, virtual volumes, and LAN-free communications.

About this task

Suppose that you want to use a source replication server to replicate data and to route commands. In the option file of the target replication server, the value of the `TCPPORT` option is `1500`. The value of the `SSLTCPPORT` option is `1542`.

Procedure

To create separate definitions, issue the following commands. For enterprise configuration, command routing, virtual volumes, and LAN-free communications, you can use the server name NOSSL. The server definition name must have a value that corresponds to the value of the **SET SERVERNAME** command on the target replication server.

```
define server nossl hladdress=1.2.3.4 lladdress=1500 ssl=no
      serverpasword=xxxxx
```

You can use the server name SSL for node replication:

```
define server ssl hladdress=1.2.3.4 lladdress=1542 ssl=yes
      serverpassword=xxxxx
```

Verifying a node replication setup before processing

Before you start a replication process, take steps to ensure that your configuration is set up correctly.

Validating a replication configuration

Before you start a replication process, you can display the controlling replication rules for data types in the file spaces of client nodes that are configured for replication. You can also verify the connection to the target replication server.

About this task

A *controlling rule* is the rule that the source replication server uses to replicate data in a file space. For example, suppose the replication rule for backup data in file space `/a` is DEFAULT. If the client-node rule for backup data is ALL_DATA, the controlling rule for the backup data in file space `/a` is ALL_DATA.

Procedure

To validate a replication configuration, complete one of the following steps:

- To display a list of file spaces and the controlling replication rules that apply to the file spaces, issue the **VALIDATE REPLICATION** command. For example, to display the controlling replication rules that apply to data types in the file spaces for NODE1 and NODE2, issue the following command:

  ```
  validate replication node1,node2
  ```

  All file spaces are displayed regardless of whether the state of the data types in the file spaces is enabled or disabled.
- To display the controlling replication rules and verify the connection with the target replication server, issue the following command:

  ```
  validate replication node1,node2 verifyconnection=yes
  ```

Previewing node replication results

Preview results before you start a replication process.

Procedure

To preview results, complete one of the following steps:

- To display the following information, issue the **REPLICATE NODE** command and specify `PREVIEW=YES`:
  - The names of client nodes to be replicated
  - The number of files to be replicated, deleted, or updated
  - The estimated time to complete the process
  - The tape volumes to be mounted

  For example, to display the estimated time to complete the replication process for a defined group of nodes, NODEGRP1, issue the following command:

  ```
  replicate node nodegrp1 preview=yes
  ```
- To display a list of files to be replicated, issue the **REPLICATE NODE** command and specify `PREVIEW=YES LISTFILES=YES`. For example, to display a list of files to be replicated for NODEGRP1, issue the following command:

  ```
  replicate node nodegrp1 preview=yes listfiles=yes
  ```

  Specifying the **LISTFILES** parameter signifies that the **WAIT** parameter is set to YES and that you cannot the issue the **WAIT** parameter from the server console.

Results

Remember: Information is displayed only if the following conditions are true:

- Data types in file spaces are enabled for replication. For example, suppose that a file space contains backup and archive files and that replication of backup data is enabled. Replication of archive data is disabled. In this case, only information about backup files is displayed.
- Client nodes are enabled for replication.
- If the controlling replication rule is a server rule, the server rule must be enabled.

Managing data replication

With Tivoli Storage Manager, you can have various methods to control node replication processing.

Replicating data by command

You can replicate data that belongs to one or more client nodes or the data that belongs to a defined group of client nodes. To replicate data automatically and at a scheduled frequency, define an administrative schedule.

Procedure

You can replicate data in one of the following ways:

- By issuing the **REPLICATE NODE** command on the command line
- By using the **DEFINE SCHEDULE** command to create an administrative schedule, and specify when to start replication processing

To use an administrative schedule, you create it by using the **DEFINE SCHEDULE** command. When you start replication by using an administrative schedule, node replication processing occurs at a specified frequency. Regardless of how you start replication, the command must be issued on the source replication server.
For example, suppose that you have three client nodes, NODE1, NODE2, and NODE3. To replicate data from all nodes daily at 6:00 a.m. in a schedule named REPL_SCHED, issue the following command on the source replication server:

```
define schedule repl_sched type=administrative cmd="replicate node
node*" active=yes"
description="replicate clients with name noden" starttime=06:00:00 period=1
perunits=days
```

Results

High-priority data is replicated before data with a normal priority. To schedule replication processes, you use the **DEFINE SCHEDULE** command.

If the data that belongs to a client node is being replicated, any attempt to replicate the data by issuing another **REPLICATE NODE** command fails. For example, suppose the backup data that belongs to a client node is scheduled for replication at 6:00 a.m. Replication of the archive data is scheduled for 8:00 a.m. Replication of the backup-data must complete before replication of the archive data starts.

Example

If you have many client nodes and are replicating a large amount of data, you can replicate data more efficiently by issuing several **REPLICATE NODE** commands in

separate schedules. For example, replicate the data that belongs to the most important client nodes first in a single command. After the data that belongs to those client nodes is replicated, replicate the data that belongs to the other nodes.

Tip: To ensure that replication for first group of client nodes finishes before the replication for the other nodes starts, specify `WAIT=YES` on the first **REPLICATE NODE** command. For example, if you want to replicate the data that belongs to NODE1 and NODE2 before the data that belongs to NODE3 and NODE4, issue the following commands:

```
replicate node node1,node2 wait=yes
replicate node node3,node4
```

You can use the REPLBATCHSIZE and REPLSIZETHRESH server options to increase the speed of a node replication process between two replicating servers. The REPLBATCHSIZE option specifies the number of client files that are to be replicated in a batch, within the same server transaction. The REPLSIZETHRESH option specifies, in megabytes, a threshold for the amount of data that is replicated as a batch, within the same server transaction.

Replicating data in individual file spaces

You can replicate the data in one or more file spaces that belong to individual client nodes or defined groups of client nodes. Replicating data by file space can be useful if, for example, you must replicate new data in a file space that belongs to a certain client node, and you cannot wait for a scheduled replication.

About this task

Data is replicated for a file space only when the following conditions are true:

- The replication state for data types in file spaces are enabled. For example, if the replication state for archive data in a file space is enabled, archive data in the file space is replicated.
- The controlling rule for the data type in the file space cannot be NONE. For example, suppose the replication rule for archive data in a file space is DEFAULT. If the file space rules and client node rules for archive data are both DEFAULT and the server rule for archive data is NONE, archive data in the file space is not replicated.

Procedure

To replicate data by file space, issue the **REPLICATE NODE** command and specify the file space name or file space identifier. For example, to replicate data in file space `/a` in NODE1, issue the following command:

```
replicate node node1 /a
```

What to do next

Tip: With the **REPLICATE NODE** command, you can also replicate data by priority and by data type. To achieve greater control over replication processing, you can combine replication by file space, data type, and priority.

To obtain information about the node replication process while it is running, issue the **QUERY PROCESS** command:

```
query process
```

For node replication purposes, each file space contains three logical file spaces:

- One for backup objects
- One for archive objects
- One for space-managed objects

By default, the **QUERY PROCESS** command reports results for each logical file space. Other factors also affect the output of the **QUERY PROCESS** command:

- If a file space has a replication rule that is set to NONE, the file space is not included in the count of file spaces that are being processed.
- If you specify data types in the **REPLICATE NODE** command, only those data types are included in the count of file spaces that are being processed, minus any file spaces that are specifically excluded.

Example: Backing up four file spaces with three object types

In this example, NODE1 has four file spaces with three object types. The **QUERY PROCESS** command generates the following output for node replication:

```
Process  Process Process Status
Number  Description
---------   ---------------  --------------
40  Replicate Node Replicating node(s) NODE1  File spaces
         complete: 11. File spaces identifying
         and replicating: 0. File spaces
          replicating: 1. File spaces not
         started: 0. Files current: 0. Files
         replicated: 747,915 of 913,581. Files
         updated: 0 of 0. Files deleted: 0 of 0.
         Amount replicated: 546 GB of 732 GB.
         Amount transferred: 389 GB. Elapsed time:
         0 Day(s), 4 Hour(s), 59 Minute(s).
```

Because the example includes four file spaces with three object types, 12 logical file spaces are processed for replication. The **QUERY PROCESS** command output shows that 11 logical file spaces completed replication.

Related concepts:

"Node replication processing" on page 911

Replicating data by type

You can replicate one or more data types for individual nodes or defined groups of client nodes. Replicating data by type can be useful if, for example, you have new backup data, and you cannot wait for a scheduled replication.

About this task

Remember: Replication of a data in a file space occurs only if the following conditions are true:

- Replication for data types in file spaces must be enabled. For example, if replication for archive data in a file space is disabled, archive data in the file space is not replicated.
- The controlling rule for the data type in the file space cannot be NONE. For example, suppose the replication rule for archive data in a file space is DEFAULT. If the file space rules and client node rules for archive data are both DEFAULT and the server rule for archive data is NONE, archive data in the file space is not replicated.

To replicate data by type, complete one of the following steps:

Procedure

- To replicate active and inactive backup data, issue the **REPLICATE NODE** command and specify DATATYPE=BACKUP. For example, to replicate the active and inactive backup data that belongs to NODE1, issue the following command:

  ```
  replicate node node1 datatype=backup
  ```

- To replicate only active backup data, issue the **REPLICATE NODE** command and specify DATATYPE=BACKUPACTIVE. For example, to replicate the active backup data that belongs to NODE1, issue the following command:

  ```
  replicate node node1 datatype=backupactive
  ```

- To replicate archive data, issue the **REPLICATE NODE** command and specify DATATYPE=ARCHIVE. For example, to replicate the archive data that belongs to NODE1, issue the following command:

  ```
  replicate node node1 datatype=archive
  ```

- To replicate space-managed data, issue the **REPLICATE NODE** command and specify DATATYPE=SPACEMANAGED. For example, to replicate the space-managed backup data that belongs to NODE1, issue the following command:

  ```
  replicate node node1 datatype=spacemanaged
  ```

- To replicate multiple data types, separate the data types by commas with no intervening spaces. For example, to replicate active backup data and archive data that belongs to NODE1, issue the following command:

  ```
  replicate node node1 datatype=backupactive,archive
  ```

Example

If you do not specify a type on the **REPLICATE NODE** command, all data types are replicated.

What to do next

Tip: Using the **REPLICATE NODE** command, you can also replicate data by file space and by priority. To achieve greater control over replication processing, you can combine replication by data type, file space, and priority.

Related concepts:

"Node replication processing" on page 911

Replicating data by priority

You can replicate high-priority data or normal-priority data that belongs to individual client nodes or defined groups of client nodes. Replicating data by priority can be useful if, for example, a client node recently acquired high-priority data, and you cannot wait for a scheduled replication.

About this task

Remember: Replication of data in a file space occurs only if the following conditions are true:

- Replication for data types in file spaces must be enabled. For example, if replication for archive data in a file space is disabled, archive data in the file space is not replicated.
- The controlling rule for the data type in the file space cannot be NONE. For example, suppose the replication rule for archive data in a file space is DEFAULT. If the file-space and client-node rules for archive data are both DEFAULT and the server rule for archive data is NONE, archive data in the file space is not replicated.

To replicate data by priority, complete one of the following steps:

Procedure

- To replicate only high-priority data, issue the **REPLICATE NODE** command and specify PRIORITY=HIGH. For example, to replicate the high-priority data that belongs to NODE1, issue the following command:

  ```
  replicate node node1 priority=high
  ```

- To replicate only normal-priority data, issue the **REPLICATE NODE** command and specify PRIORITY=NORMAL. For example, to replicate normal-priority data that belongs to NODE1, issue the following command:

  ```
  replicate node node1 priority=normal
  ```

Example

If you do not specify a priority on the **REPLICATE NODE** command, normal-priority and high-priority data is replicated.

What to do next

Tip: Using the **REPLICATE NODE** command, you can also replicate data by file space and by data type. To achieve greater control over replication processing, you can combine replication by priority, file space, and data type.

Related concepts:

"Node replication processing" on page 911

Scenario: Combining replication by file space, data type, and priority

You can combine replication by file space, data type, and priority to accomplish unique replication tasks. In this scenario, you replicate normal-priority archive and space-managed data in a file space that is common to two nodes.

About this task

The name of the file space is /a. It is common to NODE1 and NODE2.

Procedure

To replicate the data in the file space, issue the following command:

```
replicate node node1,node2 /a priority=normal datatype=archive,spacemanaged
```

Issuing this command replicates archive and space-managed data that is assigned the replication rule ALL_DATA.

Related concepts:

"Node replication processing" on page 911

Controlling throughput for node replication

Bandwidth can affect the performance of replication processing workload, processor capacity, the time available for replication, and the amount of network.

Managing the number of replication sessions

You can manage the maximum number of sessions that are used for sending data to the target replication server to improve replication performance.

About this task

Use the **MAXSESSIONS** parameter to specify the maximum number of sessions to use. When you calculate the value for the **MAXSESSIONS** parameter, consider the available network bandwidth and the processor capacity of source and target replication servers.

Consider the number of logical and physical drives that can be dedicated to the replication process. You must ensure that there are enough drives available for replication processing because other server processes or client sessions might also be using drives. The number of mount points and drives available for replication operations depends on the following factors:

- Tivoli Storage Manager server activity that is not related to replication
- System activity
- The mount limits of the device classes for the sequential-access storage pools that are involved
- The availability of a physical drive on the source and target replication servers, if the device type is not FILE
- The available network bandwidth and the processor capacity of source and target replication servers

Procedure

Issue the **REPLICATE NODE** command and specify the **MAXSESSIONS** parameter to determine the number of data sessions. For example, to set the maximum number of replication sessions to 6 for NODE_GROUP1, issue the following command:

```
replicate node node_group1 maxsessions=6
```

Restriction: The following values apply to the **MAXSESSIONS** parameter:

- The value that is specified by the **MAXSESSIONS** parameter applies only to data sessions. Data sessions are sessions during which data is sent to a target replication server. If you issue a **QUERY SESSION** command, the total number of sessions might exceed the number of data sessions. The difference can occur if you use short control sessions that you use to query and set up replication operations.
- The value of the **MAXSESSIONS** parameter represents the maximum allowable number of sessions. The number of sessions that are used for replication depends on the amount of data to be replicated. If you are replicating only a small amount of data, there is no advantage to increasing the number of sessions.

Tuning replication processing

You can determine how many bytes per hour are replicated over your network at a specific time by tuning the settings that affect replication and by testing replication processing. Based on the replication results and the total amount of node data that you replicate, you can choose which method to use for the initial replication.

About this task

Do not use a storage pool that is enabled for data deduplication to test replication. By using storage pools that are not enabled for data deduplication to test replication processing, you avoid processing extents that can increase the amount of preprocessing time of the replication process. By determining the data transfer and network capability of your replication operation without extent processing, you get a better representation of the capability of your system. Test replication processing with storage pools that are enabled for data deduplication if you want to determine the effect of data deduplication on replication performance alone.

You must calculate the bytes-per-hour value for each source replication server individually. You can determine which method is the most suitable for the server, based on its bytes-per-hour value.

Complete the following steps to determine how much data you can replicate during a specified timeframe so that you can tune replication processing for a server. Repeat these steps to obtain bytes-per-hour value for each server that you want to use for replication processing.

Procedure

1. Complete the following steps to select the appropriate data:
   a. Select one or more nodes and file spaces that have approximately 500 GB to 1 TB of total data.
   b. Select data that is typical of the data that you replicate on a routine basis.
   c. Select nodes that are configured for replication.
2. To display the amount of data in a file space, issue the **QUERY OCCUPANCY** command.
3. Select a timeframe during which replication is running normally.
4. If you plan to use Secure Sockets Layer (SSL) as the communication protocol for replication processing, ensure that SSL is enabled.

   Restriction: SSL can slow replication processing.
5. Start the replication process by issuing the **REPLICATE NODE** command.
6. After the replication is completed, review the summary message that is displayed. Use the `Amount transferred` and `Elapsed time` values that display in the summary message to determine the number of bytes per hour that can be replicated.

Results

When you determine the bytes-per-hour value for each server, you can determine a method to use for the initial replication.

Related tasks:

"Selecting a method for the initial replication" on page 931

Managing more workload during replication

By testing replication processing, you can determine whether your network can manage more workload.

Procedure

To see how your network manages more workload during replication, complete the following tasks:

1. Increase the value of the **MAXSESSIONS** parameter by 10 on the **REPLICATE NODE** command and run the test again.
2. Increase the number of replication sessions by 10 to transfer more data concurrently during replication. Alternatively, if you determine that 10 replication sessions (the default **MAXSESSIONS** value) cause your network to degrade below acceptable levels, decrease the value of the **MAXSESSIONS** parameter.
3. Repeat the process, and adjust the value of the **MAXSESSIONS** parameter to determine optimal data transfer capability.

Tuning mount points for data deduplication and replication

You can specify the maximum allowable number of data sessions that are used to send data to a target replication server. When you increase the number of data sessions to improve node replication throughput. Use storage pools that are enabled for data deduplication to improve replication performance.

Procedure

1. Determine the number of logical and physical drives that can be dedicated to the replication process. For example, if a library has 10 tape drives and 4 of the drives are used for another task, 6 drives are available for node replication.
2. Specify the number of mount points you require and ensure that there are drives available to complete node replication. For example, to set the number of point points to 6, issue the following **UPDATE DEVCLASS** command and specify the **MOUNTLIMIT** parameter:

   ```
   update devclass device_class_name mountlimit=6
   ```

3. Specify the number of data sessions that you want by using the **MAXSESSIONS** parameter on the **REPLICATE NODE** command. For example, issue the following **REPLICATE NODE** command and specify the **MAXSESSIONS** parameter:

   ```
   replicate node prodnode maxsessions=6
   ```

Disabling and enabling node replication

With Tivoli Storage Manager, you can temporarily disable node replication. You can enable node replication at any time.

Disabling and enabling replication of data types in a file space

Disabling replication of a data type in a file space can be useful if, for example, a file space has large amounts of backup data and archive data. You can enable the replication of backup data and disable replication of archive data. After the backup data is replicated, you can enable replication of archive data.

About this task

To determine the replication state of a data type in a file space, issue the **QUERY FILESPACE** command with the **FORMAT** parameter set to DETAILED.

Restriction: You cannot disable or enable replication for an entire file space. You can only disable and enable replication of a data type in a file space.

To disable and enable replication of data types in a file space, complete one or more of the following actions:

Procedure

- Backup data:
  - To disable replication, issue the following command:

    ```
    update filespace node_name file_space_name datatype=backup
      replstate=disabled
    ```

  - To enable replication, issue the following command:

    ```
    update filespace node_name file_space_name datatype=backup
      replstate=enabled
    ```

- Archive data:
  - To disable replication, issue the following command:

    ```
    update filespace node_name file_space_name datatype=archive
      replstate=disabled
    ```

  - To enable replication, issue the following command:

    ```
    update filespace node_name file_space_name datatype=archive
      replstate=enabled
    ```

- Space-managed data:
  - To disable replication, issue the following command:

    ```
    update filespace node_name file_space_name datatype=spacemanaged
      replstate=disabled
    ```

  - To enable replication, issue the following command:

    ```
    update filespace node_name file_space_name datatype=spacemanaged
      replstate=enabled
    ```

Related concepts:

"Replication state" on page 915

Disabling and enabling replication for individual client nodes

Disabling replication for a client node causes the server to skip the node during replication processing. Disabling replication for a client node can be useful if, for example, data that belongs to the node is corrupted or contaminated. After the problem is fixed, you can enable replication.

About this task

To determine the replication state of a node, issue the **QUERY NODE** command.

Procedure

- To disable replication for a node, issue the **UPDATE NODE** command and specify REPLSTATE=DISABLED. For example, to disable replication for NODE1, issue the following command:

  ```
  update node node1 replstate=disabled
  ```

- To enable replication for a node, issue the **UPDATE NODE** command and specify REPLSTATE=ENABLED. For example, to enable replication for NODE1, issue the following command:

  ```
  update node node1 replstate=enabled
  ```

What to do next

Remember: If you disable replication for a client node while data that belongs to the node is being replicated, the replication process is not affected. Replication of the data continues until all the data that belongs to the client node is replicated. However, replication for the client node will be skipped the next time that replication runs.

Related concepts:

"Replication state" on page 915

Disabling and enabling replication for all client nodes

You can disable and enable outbound and inbound sessions, including replication sessions, between servers. If sessions are disabled, you must enable them to restore function.

Disabling and enabling outbound or inbound sessions:

Disabling outbound or inbound sessions can be useful if, for example, you plan a network outage that affects communication between source and target replication servers. Disabling and enabling sessions affects not only node replication operations but also certain other types of operations.

About this task

To display the status and direction of sessions for a particular server, issue the **QUERY STATUS** command.

Remember:

- When you disable sessions for a particular server, you disable the following types of sessions in addition to replication:
  - Server-to-server event logging
  - Enterprise management
  - Server registration
  - LAN-free sessions between storage agents and the Tivoli Storage Manager server
  - Data storage that uses virtual volumes
- If you disable only outbound sessions on a source replication server, data that belongs to client nodes that store data on the source replication server do not have their data replicated. However, inbound sessions to the target replication server can occur.

  If a server is the target for multiple source replication servers and you disable outbound sessions on a single source replication server, the target replication server continues to receive replicated data from the other source replication servers.

To disable and enable replication sessions:

Procedure

- Between a single source replication server and a single target replication server:
  - To disable sessions, issue one of the following commands:
    - On the source replication server:

      ```
      disable sessions server target_server_name direction=outbound
      ```

- On the target replication server:

  ```
  disable sessions server source_server_name direction=inbound
  ```

– To enable sessions:

- If outbound sessions to the target replication server are disabled on the source replication server, issue the following command on the source replication server:

  ```
  enable sessions server target_server_name direction=outbound
  ```

- If inbound sessions from the source replication server are disabled on the target replication server, issue the following command on the target replication server:

  ```
  enable sessions server source_server_name direction=inbound
  ```

• Between multiple source replication servers and a single target replication server:

– To disable sessions, issue one of the following commands:

- On the source replication servers:

  ```
  disable sessions server * direction=outbound
  ```

- On the target replication server:

  ```
  disable sessions server * direction=inbound
  ```

– To enable sessions:

- If outbound sessions to the target replication server are disabled on the source replication servers, issue the following command on the source replication servers:

  ```
  enable sessions server * direction=outbound
  ```

- If inbound sessions to the target replication server are disabled on the source replication servers, issue the following command on the target replication server:

  ```
  enable sessions server * direction=inbound
  ```

Disabling and enabling outbound node replication processing:

When you disable outbound node replication processing, you prevent new replication processes from starting on a source replication server. Enabling outbound node replication processing is required after a database restore.

About this task

Restriction: When you restore the Tivoli Storage Manager database, replication is automatically disabled. Disabling replication prevents the server from deleting copies of data on the target replication server that are not referenced by the restored database. After a database restore, you must re-enable replication.

To display the status of replication processing for a particular server, issue the **QUERY STATUS** command.

Issue the following commands on the source replication server to disable and enable replication processing:

Procedure

• To disable replication, issue the **DISABLE REPLICATION** command.

• To enable replication, issue the **ENABLE REPLICATION** command.

Disabling and enabling replication rules

Disabling a replication rule suspends replication of all data to which the rule applies. For example, you can disable replication of active backup data in all client nodes and file spaces that have the replication rule ACTIVE_DATA.

About this task

Disabling a replication rule can be useful if, for example, you replicate groups of normal-priority and high-priority client nodes on different schedules. For example, suppose that the data that belongs to some client nodes is assigned the ALL_DATA_HIGH_PRIORITY replication rule. The data that belongs to other client nodes is assigned the ALL_DATA replication rule. The client nodes are separated into groups, in which some of the nodes in each group have high-priority data and other nodes in the group have normal-priority data.

You schedule replication for each group to take place at different times. However, a problem occurs, and replication processes take longer than expected to complete. As a result, the high-priority data that belongs to client nodes in groups that are scheduled late in the replication cycle is not being replicated.

To replicate the high-priority data as soon as possible, you can disable the ALL_DATA rule and rerun replication. When you rerun replication, only the client node data that is assigned the ALL_DATA_HIGH_PRIORITY rule is replicated. After high-priority data is replicated and the problem that caused the delay is fixed, you can enable the ALL_DATA replication rule and resume scheduled replication processing.

You can disable or enable the following replication rules:

ALL_DATA
: Replicates all backup, archive, and space-managed data. The data is replicated with a normal priority.

ACTIVE_DATA
: Replicates only the active backup data. The data is replicated with a normal priority.

ALL_DATA_HIGH_PRIORITY
: Replicates backup, archive, and space-managed data. The data is replicated before data that has a normal priority.

ACTIVE_DATA_HIGH_PRIORITY
: Replicates only active backup data. The data is replicated before data that has a normal priority.

To disable and enable replication rules, complete one of the following steps:

Procedure

- To disable a replication rule, issue the **UPDATE REPLRULE** command and specify STATE=DISABLED. For example, to disable the replication rule ACTIVE_DATA_HIGH_PRIORITY, issue the following command:

  ```
  update replrule active_data_high_priority state=disabled
  ```

- To enable a replication rule, issue the **UPDATE REPLRULE** command and specify STATE=ENABLED. For example, to enable the replication rule ACTIVE_DATA_HIGH_PRIORITY, issue the following command:

  ```
  update replrule active_data_high_priority state=enabled
  ```

Related concepts:

"Replication state" on page 915

Purging replicated data in a file space

You can purge the backup, archive, or space-managed data from a file space on the target replication server and prevent future replication of the data type.

Procedure

To prevent replication of a data type and purge the data from the file space on the target replication server, issue the **UPDATE FILESPACE** command and specify REPLSTATE=PURGEDATA. For example, to prevent replication of backup data in file space /a on NODE1 and delete the backup data in file space /a on the target replication server, issue the following command:

```
update filespace node1 /a datatype=backup replstate=purgedata
```

Data is purged the next time that replication runs for the file space. After the data is purged, the replication rule for the specified data type is set to DEFAULT. Replication for the data type is disabled.

Replicating client node data after a database restore

When you restore the Tivoli Storage Manager database on a source replication server, replication is automatically disabled. Before re-enabling replication, you can take steps to preserve the client node data that is on the target replication server.

About this task

Disabling replication prevents the Tivoli Storage Manager server from deleting copies of data on the target replication server that are not referenced by the restored database. Before re-enabling replication, determine whether copies of data that are on the target replication server are needed. If they are, complete the steps that are described in the following example. In the example, the name of the source replication server is PRODSRV. DRSRV is the name of the target replication server. NODE1 is a client node with replicated data on PRODSRV and DRSRV.

Restriction: You cannot use Secure Sockets Layer (SSL) for database restore operations.

Procedure

1. Remove NODE1 from replication on PRODSRV and DRSRV by issuing the **REMOVE REPLNODE** command:

   ```
   remove replnode node1
   ```

2. Update NODE1 definitions PRODSRV and DRSRV. When replication occurs, DRSRV sends the data to PRODSRV that was lost because of the database restore.
   a. On DRSRV, issue the **UPDATE NODE** command and specify the replication mode SYNCSEND:

      ```
      update node node1 replstate=enabled replmode=syncsend
      ```

   b. On PRODSRV, issue the **UPDATE NODE** command and specify the replication mode SYNCRECEIVE:

      ```
      update node node1 replstate=enabled replmode=syncreceive
      ```

3. On DRSRV, set the replication rules to match the rules on PRODSRV. For example, if only archive data was being replicated from PRODSRV to DRSRV,

set the rules on DRSRV to replicate only archive data from DRSRV to PRODSRV. Backup and space-managed data are not replicated to PRODSRV.

To set rules, you can issue the following commands:

- **UPDATE FILESPACE**
- **UPDATE NODE**
- **SET ARREPLRULEDEFAULT**
- **SET BKREPLRULEDEFAULT**
- **SET SPREPLRULE**

4. On DRSRV, issue the **SET REPLSERVER** command to set PRODSRV as the target replication server:

   ```
   set replserver prodsrv
   ```

5. On DRSRV, issue the **REPLICATE NODE** command to replicate data that is stored on NODE1:

   ```
   replicate node node1
   ```

   Replication processing changes the replication state of NODE1 to SEND on DRSRV and to RECEIVE on PRODSRV.

6. Remove NODE1 from replication on PRODSRV and DRSRV by issuing the **REMOVE REPLNODE** command:

   ```
   remove replnode node1
   ```

7. Update NODE1 definitions:
   a. On DRSRV, issue the **UPDATE NODE** command and specify the replication mode SYNCRECEIVE:

      ```
      update node node1 replstate=enabled replmode=syncreceive
      ```

   b. On PRODSRV, issue the **UPDATE NODE** command and specify the replication mode SYNCSEND:

      ```
      update node node1 replstate=enabled replmode=syncsend
      ```

8. On PRODSRV, enable replication by issuing the **ENABLE REPLICATION** command:

   ```
   enable replication
   ```

9. On PRODSRV, issue the **REPLICATE NODE** command to replicate data that is stored on NODE1:

   ```
   replicate node node1
   ```

   Replication processing changes the replication state of NODE1 to SEND on PRODSRV and to RECEIVE on DRSRV.

Results

The original replication configuration is restored. PRODSRV has all the data that was lost because of the database restore.

What to do next

Remember: In step 4, you set the PRODSRV as the target replication server for DRSRV. If in your original configuration you were replicating data from DRSRV to another server, you must reset the target replication server on DRSRV. For example, if you were replicating data from DRSRV to BKUPDRSRV, issue the following command on DRSRV:

```
set replserver bkupdrsrv
```

Canceling replication processes

To cancel all running node-replication processes, issue the **CANCEL REPLICATION** command.

Monitoring node replication processing and verifying results

Tivoli Storage Manager provides various methods for monitoring node replication processes and verifying results.

Displaying information about node replication settings

You can display information about replication settings for file spaces and client nodes. You can also display information about replication rules.

Displaying information about node replication settings for file spaces

To display the replication rules and the replication state of data types for file spaces, issue the **QUERY FILESPACE** command.

Displaying information about node replication settings for client nodes

To display information about the replication state, mode, and rules for client nodes, issue the **QUERY NODE** command.

Displaying information about node replication rules

To display information about replication rules, including the state, issue the **QUERY REPLRULE** command.

Displaying information about node replication processes

You can display information about running and ended replication processes, such as the estimated time for completion of a replication process.

Displaying information about running and completed node replication processes

A replication record is created when a replication process is started. The record is retained in the database of the source replication server.

Procedure

To display records about running and completed node-replication processes for a node or a file space that belongs to a node, issue the **QUERY REPLICATION** command.

Important: You cannot display information about running replication processes for client nodes that are being converted from import and export operations to replication operations. The data synchronization process might run for a long time, but it occurs only once for a client node that is being converted.
The default record-retention period for completed processes is 30 days. To display the retention period, issue the **QUERY STATUS** command and check the value in the `Replication Record Retention Period` field.
The record for a running process is updated only after a group of files is processed and committed. A *file group* consists of 2,000 files or 2 GB of data, whichever is smaller. For example, if a single file is 450 GB, the record is not updated for a relatively long time. If you notice that the number of files that are not yet replicated for a running process is not decreasing fast enough, network bandwidth or time might be insufficient to replicate the amount of data. Take one of the following actions:

- Provide more time for replication.
- Decrease the amount of data to replicate.
- Create more parallel data-transmission sessions between the source and target replication servers by increasing the value of the **MAXSESSIONS** parameter. Increase the value of the **MAXSESSIONS** parameter only if network bandwidth and processor resources for the source and target replication servers are sufficient.

Tip: To display summary of information about a running replication process, including file spaces and bytes replicated, issue the **QUERY PROCESS** command.

Displaying the activity log of a replication process

To display messages that are associated with a node replication process, review the source replication server activity log.

Procedure

The server activity log contains messages with the following information:

- The nodes that were enabled or disabled for replication
- The number of files that were eligible to be replicated compared to the number of those files that were already stored on the target replication server
- The number of files that were successfully replicated and the number of files that were missed
- The number of files on the target replication server that were deleted

Displaying summary records of replication processes

Summary records are created for each node that is processed by a **REPLICATE NODE** command. Summary records also document the entire replication operation at the process level.

Procedure

To view summary records, issue the following command:

```
select * from summary
```

Measuring the effectiveness of a replication configuration

A replication configuration is optimal if the number of replicated files that are stored on a target replication server is equal to the number of files that are stored on the source replication server. If there are fewer files on the target server, start replication processing.

Procedure

To display the number of files that are stored on source and target replication servers, issue the **QUERY REPLNODE** command. You can issue the command on a source or a target replication server.

Results

The information in the output for **QUERY REPLNODE** includes files that are stored at the time the command is issued. If a replication process is running, the information does not include files that are waiting to be transferred. Information is reported by data type. For example, you can determine the number of backup files that belong to a client node that are stored on the source and the target replication servers.

The number of replication-eligible files on the source replication server might be much larger than the number of replicated files on the target replication server, or getting larger over time. If so, one or more of the following problems might be the reason:

- Replication processes are being canceled prematurely.
- The number of sessions for transferring data is insufficient.
- The target replication server might be busy with operations not related to replication, and replication processing is not being given sufficient processor cycles.

To help diagnose problems with replication processing, issue the **QUERY REPLICATION** command.

Measuring the effects of data deduplication on node replication processing

The destination storage pool on the target replication server might be enabled for data deduplication. If it is, the source replication server sends only extents that are not in the storage pool.

Procedure

To display information about the effects of data deduplication on replication processing, issue the **QUERY REPLICATION** command and specify `FORMAT=DETAILED`.

Results

In the output, check the values in the fields that represent bytes replicated and bytes transferred for each data type:

- Replicated bytes are bytes that were replicated to the target replication server. If a file was stored in a deduplicated storage pool, the number of bytes in the stored file might be less than the number of bytes in the original file. This value in this field represents the number of physical bytes in the original file.
- Transferred bytes represent the number of bytes that were sent to the target replication server. For files stored in a deduplicated storage pool, the value in this field includes the number of bytes in the original file before duplicate extents were removed. If duplicate extents were already on the target replication server, the number of bytes in the original file is more than the number of bytes transferred.

Related concepts:

"Replication of deduplicated data" on page 919

"Active log mirror" on page 638

Related tasks:

Part 6, "Protecting the server," on page 819

Retaining replication records

To maintain adequate information about replication processes, you can adjust the length of time that the source replication server retains replication records in its database.

About this task

A replication record is created when a replication process is started. By default, replication records are retained for 30 calendar days. A calendar day comprises 24-hours, from midnight to midnight. For example, suppose that the retention period is two calendar days. If a replication process completes at 11:00 p.m. on day *n*, a record of that process is retained for 25 hours until midnight on day *n+1*.

To display the retention period for replication records, issue the **QUERY STATUS** command on the source replication server.

Procedure

To set the retention period for replication records, issue the **SET REPLRETENTION** command.

Example

Replication records that exceed the retention period are deleted from the database by Tivoli Storage Manager during automatic inventory-expiration processing. As a result, the amount of time that retention records are retained can exceed the specified retention period

If a replication process runs longer than the retention period, the record of the process is not deleted until the process ends, the retention period passes, and expiration runs.

What to do next

To display the contents of replication records, issue the **QUERY REPLICATION** command and specify `FORMAT=DETAILED`.

Recovering and storing client data after a disaster

If a source replication server is unavailable, client nodes can restore, retrieve, and recall data from the target replication server. If a source replication server is unavailable and client store operations are necessary, the nodes can back up, archive, and migrate data to the server.

Restoring, retrieving, and recalling data manually from a target replication server

To manually restore, retrieve, or recall data from a target replication server, update the client options file to point to the target replication server. Changes to node replication settings are not required.

Converting client nodes for store operations on a target replication server

If a source replication server is unavailable, you can convert client nodes to be non-replicating nodes. Non-replicating client nodes can back up, archive, or migrate data to a target replication server.

About this task

Important: Before you convert client nodes to be non-replicating nodes, carefully consider the consequences. For example, suppose that you remove multiple client nodes from replication on source replication server SRCSRV and back up, archive, and migrate the node data to target replication server TGTSRV. Later, you decide to resume backup, archive, and migration operations to SRCSRV and to replicate to TGTSRV. The client node data on the two servers must be synchronized. The synchronization process can be lengthy and resource-intensive.

To convert a client node for storage operations on a target replication server, complete the following steps:

Procedure

1. On the target replication server, issue the **REMOVE REPLNODE** command. For example, to modify the definition of NODE1 so that it is a non-replicating node, issue the following command:

   ```
   remove replnode node1
   ```

2. To back up, archive, or migrate data, update the client options file to point to the target replication server.

What to do next

If any schedules were defined on the source replication server, you can redefine them on the target replication server. Client node data on the target replication server is now managed by policies on the target replication server. For example, file expiration and deletion are managed by the target replication server.

Removing a node replication configuration

If you do not want to replicate data, you can remove a node replication configuration.

Before you begin

Delete any administrative schedules on source replication servers that issue the **REPLICATE NODE** command for the client nodes that are included in the configuration.

Procedure

1. Remove all nodes that are configured for replication by issuing the **REMOVE REPLNODE** command on all source and target replication servers that have the nodes that are configured for replication. For example, to remove NODE_GROUP1 from replication, issue the following command:

   ```
   remove replnode node_group1
   ```

2. Remove the target replication server by issuing the **SET REPLSERVER** command on the source replication server. Do not specify the name of the target replication server. For example, to remove target replication server TGTSRV, issue the following command:

   ```
   set replserver
   ```

What to do next

To verify that the target replication server was removed, issue the **QUERY STATUS** command on the source replication server. If the target replication server was removed, the field `Target Replication Server` is blank.

Tip: If you do not want to keep replicated node data on the target replication server, you can delete the replicated data.

Chapter 30. 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.

About this task

You can recover data at an alternative 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.

You can use complementary technologies to protect the Tivoli Storage Manager server and to provide an alternative to disaster recovery. For example, you can use DB2 HADR to replicate the Tivoli Storage Manager database or device-to-device replication.

To ensure high server availability, you can use node replication with clustering to minimize a disaster risk. In a clustered environment, a client is less likely to fail over to another server. If you replicate data from several source replication servers to one target replication server, there is a high dependency on the target replication server. A clustered environment eases the dependency on the target replication server.

The following list identifies key tasks:

- "Querying defaults for the disaster recovery plan file" on page 980
- "Specifying recovery instructions for your site" on page 985
- "Specifying information about your server and client node machines" on page 986
- "Specifying recovery media for client machines" on page 988
- "Creating and storing the disaster recovery plan" on page 989
- "Managing disaster recovery plan files stored on target servers" on page 991
- "Moving backup media" on page 994
- "Managing the Disaster Recovery Manager tasks" on page 999
- "Preparing for disaster recovery" on page 1001
- "Recovering from a disaster" on page 1003

Before you use DRM, familiarize yourself with Chapter 28, "Protecting and recovering the server infrastructure and client data," on page 857.

Note: Unless otherwise noted, you need system privilege class to perform DRM tasks.

Related reference:

"Disaster recovery manager checklist" on page 1013

"The disaster recovery plan file" on page 1018

© Copyright IBM Corp. 1993, 2013

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.

About this task

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.

About this task

The following table describes how to set defaults for the disaster recovery plan file.

Table 78. Defaults for the disaster recovery plan file

Process	Default
Primary storage pools to be processed	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. **The default at installation:** All primary storage pools. **To change the default:** SET DRMPRIMSTGPOOL For example, to specify that only the primary storage pools named PRIM1 and PRIM2 are to be processed, enter: `set drmprimstgpool prim1,prim2` **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. **To override the default:** Specify primary storage pool names in the PREPARE command

Table 78. Defaults for the disaster recovery plan file (continued)

Process	Default
Copy storage pools to be processed	When the recovery plan file is generated, you can limit processing to specified pools. **The default at installation:** All copy storage pools. **To change the default:** SET DRMCOPYSTGPOOL For example, to specify that only the copy storage pools named COPY1 and COPY2 are to be processed, enter: `set drmcopystgpool copy1,copy2` 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. **To override the default:** Specify copy storage pool names in the PREPARE command
Active-data pools to be processed	When the recovery plan file is generated, you can limit processing to specified pools. **The default at installation:** None **To specify the default:** SET DRMACTIVEDATASTGPOOL For example, to specify that only the active-data pools named ACTIVEPOOL1 and ACTIVEPOOL2 are to be processed, enter: `set drmactivedatastgpool activepool1,activepool2` To remove any specified active-data pool names, specify a null string ("") in SET DRMACTIVEDATASTGPOOL. Active-data pool volumes in MOUNTABLE state are processed only if you specify the active-data pools using the SET DRMACTIVEDATASTGPOOL command or the ACTIVEDATASTGPOOL 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. 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 ACTIVEDATASTGPOOL parameter: • 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. **To override the default:** Specify active-data pool names using the MOVE DRMEDIA, QUERY DRMEDIA, or PREPARE command.

Table 78. Defaults for the disaster recovery plan file (continued)

Process	Default
Identifier for replacement volume names	To restore a primary storage pool volume, mark the original volume *destroyed* 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. **The default identifier at installation: @** **To change the default:** SET DRMPLANVPOSTFIX For example, to use the character r, enter: `set drmplanvpostfix r`

Recovery instructions prefix	You can specify a prefix for the names of the recovery instructions source files in the recovery plan file.
	
	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 *Administrator's Reference* or enter HELP PREPARE from administrative client command line.
	
	To set a default: SET DRMINSTRPREFIX
	
	For example, to specify the prefix, enter:
	`set drminstrprefix /u/recovery/plans/rpp`
	
	The disaster recovery plan file will include, for example, the following file:
	`/u/recovery/plans/rpp.RECOVERY.INSTRUCTIONS.GENERAL`
	
	To override the default: The INSTRPREFIX parameter with the PREPARE command

Prefix for the recovery plan file	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.
	
	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 *Administrator's Reference* or enter HELP PREPARE from administrative client command line.
	
	To change the default: SET DRMPLANPREFIX
	
	For example, to specify the prefix, enter the following command:
	`set drmplanprefix /u/server/recoveryplans/`
	
	The disaster recovery plan file name created by PREPARE processing will be in the following format:
	`/u/server/recoveryplans/20000603.013030`
	
	To override the default: The PLANPREFIX parameter with the PREPARE command

Table 78. Defaults for the disaster recovery plan file (continued)

Process	Default
The disaster recovery plan expiration period	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: • 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. **The default at installation:** 60 days **To change the default:** SET DRMRPFEXPIREDAYS For example, to change the time to 90 days, enter: `set drmrpfexpiredays 90`

Specifying defaults for offsite recovery media management

You can set defaults for offsite recovery media management.

About this task

Table 79. Defaults for offsite recovery media management

Process	Default
Copy storage pool volumes to be processed	MOVE DRMEDIA and QUERY DRMEDIA can process copy storage pool volumes in the MOUNTABLE state. You can limit processing to specified copy storage pools. **The default at installation:** All copy storage pool volumes in the MOUNTABLE state **To change the default:** SET DRMCOPYSTGPOOL **To override the default:** COPYSTGPOOL parameter on MOVE DRMEDIA or QUERY DRMEDIA
Active-data pool volumes to be processed	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. **The default at installation:** None **To specify the default:** SET DRMACTIVEDATASTGPOOL **To override the default:** ACTIVEDATASTGPOOL parameter on MOVE DRMEDIA, QUERY DRMEDIA, or PREPARE
Executable commands file name	You can use MOVE DRMEDIA or QUERY DRMEDIA to generate executable commands and store them in a file. **The default file name at installation:** None To set a default: SET DRMCMDFILENAME. For example, enter: `set drmcmdfilename /drm/orm/exec.cmds` **To override the default:** CMDFILENAME parameter on MOVE DRMEDIA or QUERY DRMEDIA

Table 79. Defaults for offsite recovery media management (continued)

Process	Default
Location name for volumes that move to the NOTMOUNTABLE state	MOVE DRMEDIA generates a location name for volumes that move to the NOTMOUNTABLE state. **The default at installation:** NOTMOUNTABLE **To change the default:** SET DRMNOTMOUNTABLENAME For example, to specify a location named LOCAL, enter: `set drmnotmountablename local`
Location name for volumes that move to the COURIER or COURIERRETRIEVE state	MOVE DRMEDIA generates a location name for volumes that are changing from NOTMOUNTABLE to COURIER or from VAULTRETRIEVE to COURIERRETRIEVE. **The default at installation:** COURIER **To change the default:** SET DRMCOURIERNAME For example, to specify a courier named Joe's Courier Service, enter: `set drmcouriername "Joe's Courier Service"`
Reading labels of checked out volumes	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. **The default at installation:** DRM reads the volume labels. **To change the default:** SET DRMCHECKLABEL For example, to specify that DRM should not read the volume labels, enter: `set drmchecklabel no`
Expiration period of a database backup series	A database backup series (full plus incremental and snapshot) is eligible for expiration if all of these conditions are true: • 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. **The default at installation:** 60 days **To change the default:** SET DRMDBBACKUPEXPIREDAYS For example, to set the expiration value to 30 days, enter: `set drmdbbackupexpiredays 30`
Whether to process copy storage pool and active-data pool volumes of the FILE device type	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. **The default at installation:** Copy storage pool and active-data pool volumes of the FILE device type are not processed **To change the default:** SET DRMFILEPROCESS To allow processing, enter: `set drmfileprocess yes`

Table 79. Defaults for offsite recovery media management (continued)

Process	Default
Vault Name	MOVE DRMEDIA uses the vault name to set the location of volumes that are moving from the COURIER state to the VAULT state **The default at installation:** The vault name is set to VAULT. **To change the default:** SET DRMVAULTNAME 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"`

Specifying recovery instructions for your site

The plan file that DRM generates is a template that contains information, including commands for recovering the database, that might not apply to your replacement systems or to your particular recovery scenario. To modify the plan or to store additional instructions that you will need during recovery from an actual disaster, use the RECOVERY.INSTRUCTIONS stanzas.

About this task

Tip: Enter your site-specific information in the stanzas when you first create the plan file or after you test it.

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 Solaris operating system.
The install volume for the Tivoli Storage Manager server is INS001. If that is
lost, you must 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.

About this task

Tip: The plan file that DRM generates is a template that contains information, including commands for recovering the database, that might not apply to your replacement systems or to your particular recovery scenario. To modify the plan or to store additional instructions that you will need during recovery from an actual disaster, use the RECOVERY.INSTRUCTIONS stanzas. Enter your site-specific information in the stanzas when you first create the plan file or after you test it.

Use the following procedure to specify information about server and client machines and to store it in the server database:

Procedure

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 adsmserver=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 1007.

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:

```
> dsmadmc -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.

Procedure

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
```

Results

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.

About this task

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 1018.

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 primary storage pools" on page 871 and "Backing up the server database" on page 858 for details about these procedures.

If you manually send backup media offsite, see "Moving copy storage pool and active-data pool volumes offsite" on page 996. If you use virtual volumes, see "Using virtual volumes to store data on another server" on page 722.

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 18, "Automating server operations," on page 609).

Tips:

- The plan file that DRM generates is a template that contains information, including commands for recovering the database, that might not apply to your replacement systems or to your particular recovery scenario. To modify the plan or to store additional instructions that you will need during recovery from an actual disaster, use the RECOVERY.INSTRUCTIONS stanzas. Enter your site-specific information in the stanzas when you first create the plan file or after you test it.
- 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
  ```

Backup of Centera storage pools is not supported so commands for Centera primary storage pools should not be included in the recovery plan file. To work around this, do one of the following:

- Use the SET DRMPRIMSTGPOOL command or the PRIMSTGPOOL parameter with the PREPARE command to specify the names of the primary storage pools that will be eligible for DRM PREPARE processing and do not include Centera storage pools. If a specified pool name contains a wildcard, ensure that the expanded name will not match the name of any Centera storage pool defined to the Tivoli Storage Manager server.
- Or, if a Centera storage pool is included in the primary pool list when a PREPARE command is executed, update the recovery plan file that it generated. Delete or comment out the commands for the Centera storage pool that may be included in the following stanzas:
  - PRIMARY.VOLUMES.DESTROYED
  - PRIMARY.VOLUMES.REPLACEMENT
  - STGPOOLS.RESTORE

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.

About this task

For example, to store the recovery plan file locally in the /u/server/recoveryplans/ directory, enter:

```
prepare planprefix=/u/server/recoveryplans/
```

What to do next

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 (yyyymmdd.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.

About this task

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 724 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" on page 992.

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).

Procedure

1. **From the source server:** Issue QUERY RPFILE the 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 rpfile 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=RPFILE) or database snapshot backups (TYPE=RPFSNAPSHOT). For example:

   ```
   query volhistory type=rpfile
   ```

3. **From the target server:** Issue a QUERY RPFILE 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 rpfile 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 on object on the target server.

About this task

For an example of the contents of a recovery plan file, see "The disaster recovery plan file" on page 1018. You cannot issue the commands shown below from a server console. An output delay can occur if the plan file is located on tape.

Procedure

- **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 RPFILE command.

About this task

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*:

Procedure

- **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
```

What to do next

To display a list of recovery plan files, use the QUERY RPFILE command. See "Displaying information about recovery plan files" on page 992 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.

About this task

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 991.

Procedure

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.

About this task

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.

Procedure

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.

Before you begin

Task	Required Privilege Class
Send copy storage pool and active-data pool volumes offsite and back onsite	Unrestricted storage or operator

Procedure

1. Move new backup media offsite and update the database with their locations. See "Moving copy storage pool and active-data pool volumes offsite" on page 996 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 998 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
   ```

Results

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:

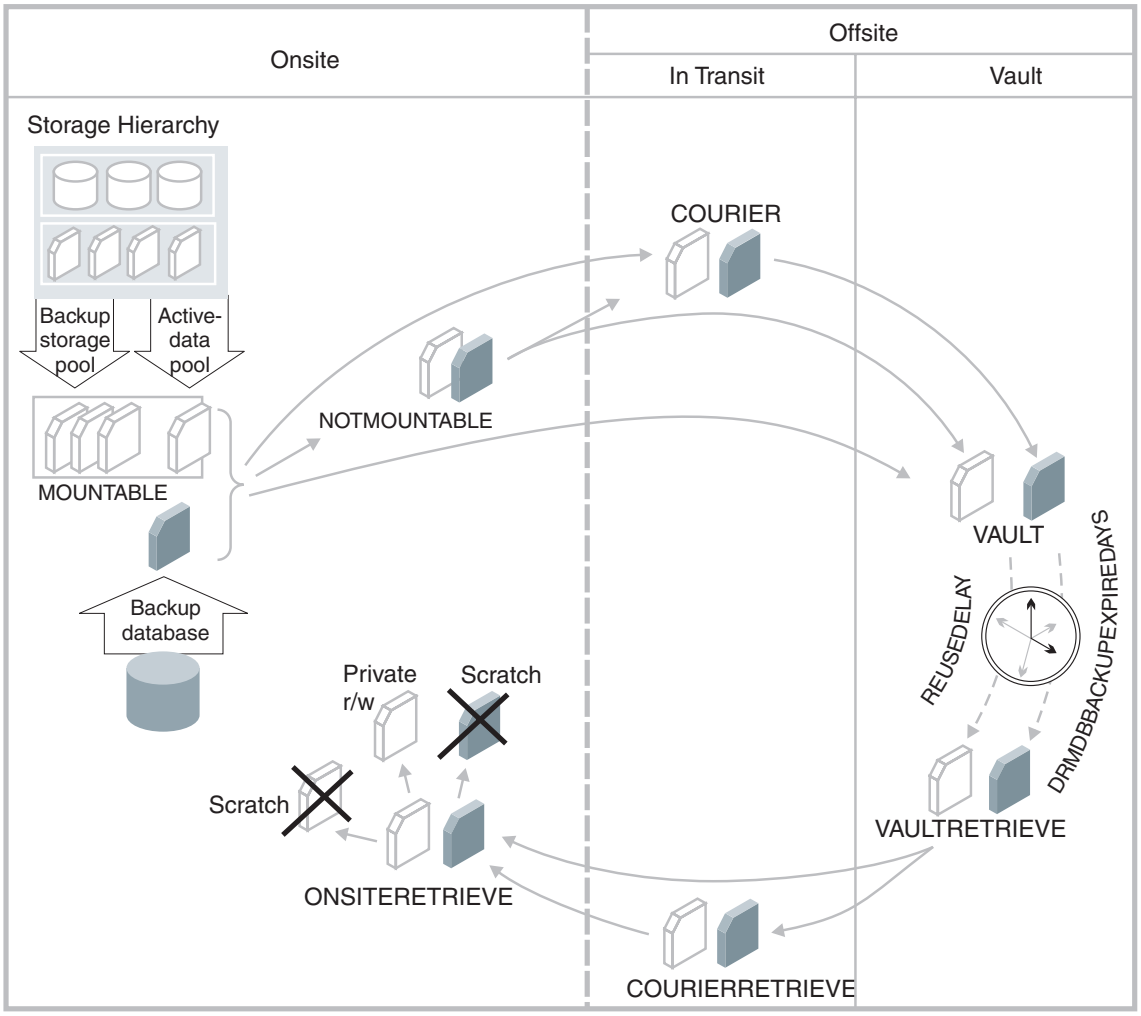


Figure 114. Recovery media life cycle

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 offsite volumes" on page 371. For information on expiration of database backup volumes, see step 1 on page 998.

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 offsite

After you created the backup copies of your primary storage pools and database, you can send your backup media offsite. To send media offsite, mark the volumes as unavailable to Tivoli Storage Manager and give them to the courier.

Procedure

Complete the following steps to identify the database backup, copy storage pool, and active-data pool volumes and move them offsite:

1. Identify the copy storage pool, active-data pool, and database backup volumes to be moved offsite For example, issue the following command:

   ```
   query drmedia * wherestate=mountable
   ```

   DRM displays information similar to the following output:

   ```
   Volume Name      State            Last Update          Automated
                                     Date/Time            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:

   ```
   move drmedia * wherestate=mountable
   ```

   Restriction: Do not run the **MOVE DRMEDIA** and **BACKUP STGPOOL** commands concurrently. Ensure that the storage pool backup processes are complete before you issue the **MOVE DRMEDIA** command.

   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 is not 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
```

3. Send the volumes to the offsite 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 is not yet issued, the default location is COURIER. For more information, see "Specifying defaults for offsite recovery media management" on page 983

4. 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 is not yet issued, the default location is VAULT. For more information, see "Specifying defaults for offsite recovery media management" on page 983.

5. 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 output:

```
Volume Name       State           Last Update          Automated
                                  Date/Time            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
```

6. 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 change 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 is not yet issued, the default location is VAULT.

Results

See "Preparing for disaster recovery" on page 1001 for an example that demonstrates sending server backup volumes offsite 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.

Procedure

1. 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. 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 983.

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 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 tostate=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.

Managing the Disaster Recovery Manager tasks

To protect data, you can use the Disaster Recovery Manager (DRM) during routine operations and during disaster recovery.

About this task

Ensure that you set up the DRM and perform the daily operations to protect the database, data, and storage pools.

Setup

1. License DRM by issuing the **REGISTER LICENSE** command.
2. Ensure that the device configuration and volume history files exist.
3. Back up the storage pools by issuing the **BACKUP STGPOOL** command.
4. Copy active data to active-data pools by using the **COPY ACTIVEDATA** command.
5. Do a full backup the database (for example, a database snapshot backup) by using the **BACKUP DB** command.
6. Define site-specific server recovery instructions in the DRM plan file.
7. Describe priority client machines by using the **DEFINE MACHINE** command.
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).

   Restriction: Ensure that the primary storage pool backup process is complete before you back up the database.
5. Mark the backup volumes as unavailable to Tivoli Storage Manager.

   Restriction: Ensure that the **BACKUP STGPOOL** command and the **BACKUP DB** command are complete before you issue the **MOVE DRMEDIA** command.
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).

   Restriction: Ensure that the primary storage pool backup process is complete before you back up the database.
4. Mark the backup volumes as unavailable to Tivoli Storage Manager.

   Restriction: Ensure that the **BACKUP STGPOOL** command and the **BACKUP DB** command are complete before you issue the **MOVE DRMEDIA** command.
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).

   Restriction: Ensure that the primary storage pool backup process is complete before you back up the database.
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 that were 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 by 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. Some or all of the inactive files might be deleted if the server determines that an inactive file must 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.

Preparing for disaster recovery

Stay prepared for a disaster by establishing a procedure for disaster recovery by using the Disaster Recovery Manager (DRM). The steps are performed by the onsite administrator unless otherwise indicated.

Procedure

Complete the following steps to prepare for disaster recovery:

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** commands)
   - The location of the boot recovery media (**DEFINE RECOVERYMEDIA** command)
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
     - Back up the database

Restriction: Ensure that the primary storage pool backup process is complete before you back up the 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
      ```

      Restriction: Ensure that the **BACKUP STGPOOL** command and the **BACKUP DB** command are complete before you issue the **MOVE DRMEDIA** command.

   b. Send the volumes offsite and record that the volumes were given to the courier:

      ```
      move drmedia * wherestate=notmountable
      ```

5. Create a 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 that is 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.`

   ```
   > dsmadmc -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 that are given to the courier by the vault:

    ```
    move drmedia * wherestate=vaultretrieve
    ```

Example

The following example of a custom maintenance script shows the recommended sequence of steps for a specific Tivoli Storage Manager server configuration:

```
PARALLEL
BACKUP STGPOOL PRIMDISK DRM_COPYTAPE WAIT=YES
BACKUP STGPOOL PRIMTAPE DRM_COPYTYPE WAIT=YES
SERIAL
PARALLEL
COPY ACTIVEDATA PRIMDISK DRM_ACTTAPE WAIT=YES
COPY ACTIVEDATA PRIMTAPE DRM_ACTTAPE WAIT=YES
SERIAL
BACKUP DB DEVCLASS=TS7650G_N34 TYPE=FULL WAIT=YES
MOVE DRMEDIA * WHERESTATE=MOUNTABLE TOSTATE=VAULT SOURCE=DBBACKUP WAIT=YES
PREPARE SOURCE=DBBACKUP WAIT=YES
MIGRATE STGPOOL PRIMDISK LOWMIG=0 WAIT=YES
EXPIRE INVENTORY SKIPDIRS=NO RESOURCE=4 WAIT=YES
PARALLEL
RECLAIM STGPOOL PRIMDISK THRESHOLD=50 WAIT=YES
RECLAIM STGPOOL PRIMTAPE THRESHOLD=50 WAIT=YES
```

Restriction: Ensure that the **BACKUP STGPOOL** command and the **BACKUP DB** command complete before you issue other commands, for example, the **MOVE DRMEDIA** command.

Related tasks:

"Creating a custom maintenance script" on page 618

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.

Procedure

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 1027).

Results

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.

Procedure

Complete the following steps.

Restriction: When you run the disaster recovery script or the commands that the script contains, the determination must be made whether to run as root or as the DB2 instance user ID.

1. Review the recovery steps that are 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 1018.) 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 following directory structures:
   - 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 that are 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

```
dsmserv removedb TSMDB1
```

f. Restore the original dsmserv.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 that are 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 894.

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 that is 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 that is 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 in the restored Tivoli Storage Manager database (see "Updating the device configuration file" on page 894

   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 must 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.

Procedure

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       IRONVAULT    POLARIS
                      vol3
```

   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
scsi1        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.

About this task

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.

About this task

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/4lb

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/4lb

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.

About this task

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/4lb

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.

About this task

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 80. Checklist

Activity	Start Date	End Date	Status	Responsible Person	Backup Person
Plan for DRM					
Evaluate your disaster recovery requirements • What are the business priorities for recovering your clients? • Where is the recovery site? • Is the recovery site hot, warm, or cold? • Do the clients have connectivity to recovery server? • Who are the system and Tivoli Storage Manager administrators? • Will you need to return to the original site? • Where are the offsite backups stored? • How does the vault handle the backup media? • How are the backups packaged or processed? • Who provides the courier service?					
Evaluate the current storage pool backup implementation • What primary storage pools are being backed up? • When are the backups performed? • Will the backups remain onsite or be sent offsite? • Naming conventions for replacement volumes for primary storage pools					

Table 80. Checklist (continued)

Activity	Start Date	End Date	Status	Responsible Person	Backup Person
Evaluate the current database backup implementation • When are the backups performed? • Backup purpose: offsite or onsite • Will you use snapshot database backups or full plus incremental database backups? • 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.					
Determine which primary storage pools are to be managed by DRM					
Determine which copy storage pools are to be managed by DRM • Offsite copy storage pools					
Determine which active-data pools are to be managed by DRM • Offsite active-data pools					
Where to Save the Recovery Plan File **Locally:** • What is the recovery plan file pathname prefix? • How will recovery plan files be made available at the recovery site? – Print and store offsite – Copy stored offsite – Copy sent/NFS to recovery site **On Another Server:** • What server is to be used as the target server? • What is the name of the target server's device class? • How long do you want to keep recovery plan files?					
Determine where you want to create the user-specified recovery instructions What is the prefix of the instructions pathname?					

Table 80. Checklist (continued)

Activity	Start Date	End Date	Status	Responsible Person	Backup Person
Analyze the sequence of steps related to the PREPARE command backup movement Document the flow of activities and timings • Sending of volumes offsite • Return of empty volumes • PREPARE timing					
Installation					
Receive and Install the Tivoli Storage Manager code					
License DRM • REGISTER LICENSE or • Update the server options					

Table 80. Checklist (continued)

Activity	Start Date	End Date	Status	Responsible Person	Backup Person
Set DRM defaults Issue: • SET DRMDBBACKUPEXPIREDAYS to define the database backup expiration • SET DRMPRIMSTGPOOL to specify the DRM-managed primary storage pools • SET DRMCOPYSTGPOOL to specify the DRM-managed copy storage pools • SET DRMACTIVEDATASTGPOOL to specify the DRM-managed active-data pools • SET DRMPLANVPOSTFIX to specify a character to be appended to new storage pools • SET DRMPLANPREFIX to specify the RPF prefix • SET DRMINSTRPREFIX to specify the user instruction file prefix • SET DRMNOTMOUNTABLENAME to specify the default location for media to be sent offsite • SET DRMCOURIERNAME to specify the default courier • SET DRMVAULTNAME to specify the default vault • SET DRMCMDFILENAME to specify the default file name to contain the commands specified with the CMD parameter on MOVE and QUERY DRMEDIA • SET DRMCHECKLABEL to specify whether volume labels are verified when checked out by the MOVE DRMEDIA command • SET DRMRPFEXPIREDAYS to specify a value for the frequency of RPF expiration (when plan files are stored on another server)					

Table 80. Checklist (continued)

Activity	Start Date	End Date	Status	Responsible Person	Backup Person
Define the site-specific recovery instructions Identify: • Target disaster recovery server location • Target server software requirements • Target server hardware requirements (storage devices) • Tivoli Storage Manager administrator contact • Courier name and telephone number • Vault location and contact person Create: • Enter the site-specific recovery instructions data into files created in the same path/HLQ as specified by SET DRMINSTRPREFIX					
Test disaster recovery manager					
Test the installation and customization • QUERY DRMSTATUS to display the DRM setup • 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. • Back up the Tivoli Storage Manager database • QUERY DRMEDIA to list the copy storage pool and active-data pool volumes • MOVE DRMEDIA to move offsite • PREPARE to create the recovery plan file					
Examine the recovery plan file created					
Test the recovery plan file break out • awk script planexpl.awk • Locally written procedure					
Set up the schedules for automated functions					

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.

Tip: The plan file that DRM generates is a template that contains information, including commands for recovering the database, that might not apply to your replacement systems or to your particular recovery scenario. To modify the plan or to store additional instructions that you will need during recovery from an actual disaster, use the RECOVERY.INSTRUCTIONS stanzas. Enter your site-specific information in the stanzas when you first create the plan file or after you test it.

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:

Tip: The plan file that DRM generates is a template that contains information, including commands for recovering the database, that might not apply to your replacement systems or to your particular recovery scenario. To modify the plan or to store additional instructions that you will need during recovery from an actual disaster, use the RECOVERY.INSTRUCTIONS stanzas. Enter your site-specific information in the stanzas when you first create the plan file or after you test it.

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 81 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 81. Administrative tasks associated with the disaster recovery plan file

Stanza Name	Tasks
PLANFILE.DESCRIPTION	None
PLANFILE.TABLE.OF.CONTENTS	None
SERVER.REQUIREMENTS	None
RECOVERY.INSTRUCTIONS.GENERAL	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.INSTRUCTIONS.OFFSITE	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.INSTRUCTIONS.INSTALL	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.INSTRUCTIONS.DATABASE	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.INSTRUCTIONS.STGPOOL	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.VOLUMES.REQUIRED	**During routine processing:** MOVE DRMEDIA
RECOVERY.DEVICES.REQUIRED	None
RECOVERY.SCRIPT. DISASTER.RECOVERY.MODE script	**During disaster recovery:** Edit and run (optional)
RECOVERY.SCRIPT. NORMAL.MODE script	**During disaster recovery:** Edit and run (optional)
DB.STORAGEPATHS	**During disaster recovery:** Edit (optional)
LICENSE.REGISTRATION macro	**During disaster recovery:** Edit and run (optional)
ACTIVEDATASTGPOOL.VOLUMES.AVAILABLE	**During routine processing:** MOVE DRMEDIA **During disaster recovery:** Edit and run (optional)

Table 81. Administrative tasks associated with the disaster recovery plan file (continued)

Stanza Name	Tasks
ACTIVEDATASTGPOOL.VOLUMES.DESTROYED	**During routine processing:** MOVE DRMEDIA **During disaster recovery:** Edit and run (optional)
COPYSTGPOOL.VOLUMES.AVAILABLE macro	**During routine processing:** MOVE DRMEDIA **During disaster recovery:** Edit and run (optional)
COPYSTGPOOL.VOLUMES.DESTROYED macro	**During routine processing:** MOVE DRMEDIA **During disaster recovery:** Edit and run (optional)
PRIMARY.VOLUMES.DESTROYED macro	**During disaster recovery:** Edit and run (optional)
PRIMARY.VOLUMES.REPLACEMENT macro	**During disaster recovery:** Edit and run (optional)
STGPOOLS.RESTORE macro	**During disaster recovery:** Edit and run (optional)
VOLUME.HISTORY.FILE configuration file	**During disaster recovery:** Copy (optional)
DEVICE.CONFIGURATION.FILE configuration file	**During disaster recovery:** Edit and copy (optional)
DSMSERV.OPT.FILE configuration file	**During disaster recovery:** Edit and copy (optional)
LICENSE.INFORMATION	None
MACHINE.GENERAL.INFORMATION	**During setup or periodic updates:** Issue DEFINE MACHINE ADSMSERVER=YES (optional)
MACHINE.RECOVERY.INSTRUCTIONS	**During setup or periodic updates:** Issue INSERT MACHINE RECOVERYINSTRUCTIONS (optional)
MACHINE.RECOVERY.CHARACTERISTICS	**During setup or periodic updates:** Issue INSERT MACHINE CHARACTERISTICS (optional)
MACHINE.RECOVERY.MEDIA	**During setup or periodic updates:** 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.

Tip: The plan file that DRM generates is a template that contains information, including commands for recovering the database, that might not apply to your replacement systems or to your particular recovery scenario. To modify the plan or to store additional instructions that you will need during recovery from an actual disaster, use the RECOVERY.INSTRUCTIONS stanzas. Enter your site-specific information in the stanzas when you first create the plan file or after you test it.

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.DESCRIPTION

```
begin PLANFILE.DESCRIPTION

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.DESCRIPTION
```

PLANFILE.TABLE.OF.CONTENTS

```
begin PLANFILE.TABLE.OF.CONTENTS

PLANFILE.DESCRIPTION
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 985 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 980 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.

Tip: The commands in the plan file that is generated by DRM might not work on your replacement systems. If necessary, use the recovery.instructions stanzas in the plan file to store information about the particular commands to be used during recovery from an actual disaster. Enter your site-specific information in the recovery.instructions stanzas when you first create the plan file or after you test it. 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 by using the administrative client. The disaster recovery plan file issues commands by 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 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 by 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 78 on page 980.

```
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\DEVCNFG.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\SERVER\DSMSERV" -k "Server1" restore db todate=09/26/2008 totime=13:28:52 +
source=dbb

rem "D:\TSM\SERVER\DSMSERV" -k "Server1" restore db todate=09/26/2008 totime=13:28:52 +
source=dbb on="D:\TSM\SERVER1\PLANPRE.DB.STORAGEPATHS" activelogdirectory="H:\tsmdata\Alog"

 rem Start the server.
start "Server1" "D:\TSM\SERVER\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.
dsmadmc -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.
dsmadmc -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.
dsmadmc -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.
dsmadmc -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.
dsmadmc -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.
dsmadmc -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
```

Related tasks:

"Restoring to a point-in-time in a shared library environment" on page 903
"Scenario: Protecting the database and storage pools" on page 886
"Scenario: Recovering a lost or damaged storage pool volume" on page 898
"Example: Restoring a library manager database" on page 891
"Example: Restoring a library client database" on page 893

Related reference:

"Recovery instructions stanzas" on page 1024

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 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 78 on page 980.

```
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
```

Related tasks:

"Restoring to a point-in-time in a shared library environment" on page 903

"Scenario: Protecting the database and storage pools" on page 886

"Scenario: Recovering a lost or damaged storage pool volume" on page 898

"Example: Restoring a library manager database" on page 891

"Example: Restoring a library client database" on page 893

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=READO 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=PRIMPOOL1

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 can 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 DSMSERV 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 *volhprefix*.
- If the server option file does not contain VOLUMEHISTORY options, the server uses the default name *volhprefix* followed by *drmvolh.txt*. The directory where the server is started from is used as the *volhprefix*.

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 894.

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 DISABLESCHEDS 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            kbytes    used   avail capacity  Mounted on
  /dev/dsk/c0t3d0s0      69191   33036   29245    53%    /
  /dev/dsk/c0t3d0s6     214143  162727   30006    84%    /usr
  /proc                      0       0       0     0%    /proc
  fd                         0       0       0     0%    /dev/fd
  swap                  225520      32  225488     0%    /tmp
  /dev/dsk/c0t3d0s7     288391  237438   22123    91%    /opt

end 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: SUNOPSYS
                 Type: Other
         Volume Names:
             Location: IRONMNT
          Description: SunOS 5.5.1 Installation CD
              Product:
  Product Information:
end MACHINE.RECOVERY.MEDIA.REQUIRED
```

Chapter 31. Integrating disaster recovery manager and node replication into your disaster recovery strategy

Many tools, architectures, and technologies are available for you to use to implement a disaster recovery strategy. For example, you can use disaster recovery manager (DRM) to vault of offsite volumes, generate a disaster recovery plan, and automate the recovery of the Tivoli Storage Manager server. To maintain the same level of files on two networked servers, you can use Tivoli Storage Manager node replication.

The framework for evaluating disaster recovery strategies consists of the following tiers:

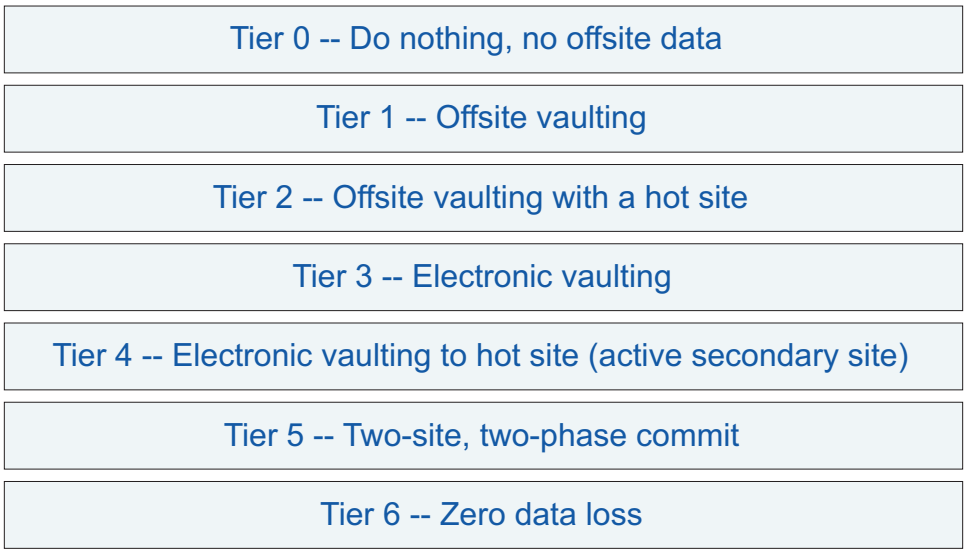


Figure 115. Tiers of disaster recovery

Each tier corresponds to different recovery times and potentials for data loss. For example, in a tier 1 production site data is typically saved only selectively, and volumes that are stored at an offsite facility can be difficult to track. In addition, recovery time is unpredictable. After a disaster, hardware and software must be restored, and storage volumes must be sent back to the production site.

A tier 6 strategy consists of synchronous remote-copy technologies between peer production sites. Sites with tier 6 protection offer fast recovery times with no loss of data. Data is considered lost only if a transaction started, for example if the user presses Enter, but the transaction request was not completed at both sites.

© Copyright IBM Corp. 1993, 2013

Plan for a disaster recovery strategy

For optimal results, plan the integration of disaster recovery manager (DRM) and node replication.

Use the following questions as a guide to help you in the planning process:

Cost How much can you afford for your disaster recovery implementation?

Performance
Do you want a low or a high performance disaster recovery solution?

Recovery Time Objective (RTO) and Recovery Point Objective (RPO)
What are your system requirements?

Current disaster recovery strategy
What disaster recovery strategy is implemented in your environment?

Data What data do you need? Categorize and prioritize the data that you require.

When you plan a disaster recovery strategy that might be suitable for your site, consider using DRM and Tivoli Storage Manager node replication for these reasons:

- DRM is an effective tool for managing offsite vaulting. With DRM, you can configure and automatically generate a disaster recovery plan that contains the information, scripts, and procedures that are required to automatically restore the server and recover client data after a disaster.

  DRM also manages and tracks the media on which client data is stored, whether the data is on site, in-transit, or in a vault, so that the data can be more easily located if disaster strikes. DRM also generates scripts that assist you in documenting information-technology systems and recovery procedures that you can use, including procedures to rebuild the server.

  Use DRM alone to meet the disaster recovery objectives in tier 1, or use it together with other backup-and-recovery tools and technologies in tiers 2, 3 and 4.
- Tivoli Storage Manager node replication meets the objectives of tier 5. After a successful node replication, the target server contains all metadata updates and data that is stored on the source server.

  In addition to fast recovery and minimal potential data loss, Tivoli Storage Manager node replication offers the following advantages:
  - Node replication is easier to manage than device-based replication. Device-based replication requires that you keep the database and the data it represents synchronized. You manually schedule database backups to match the point in time when the device synchronizes.
  - Results for Tivoli Storage Manager operations are reported in terms such as "node names" and "file names." In contrast, device-based replication results are reported in terms of "disks," "sectors," and "blocks."

Tier 0: No disaster recovery capability

Tier 0 is defined as a single-site data center that does not back up data or, if data is backed up, backups are left on site. The data center does not have a disaster recovery plan or backup hardware and software.

In the following figure, the Tivoli Storage Manager server and database, tape libraries, and tapes are in a single facility. If a disaster occurs, recovery time is unpredictable. Tier 0 is not recommended and data might never be recovered.

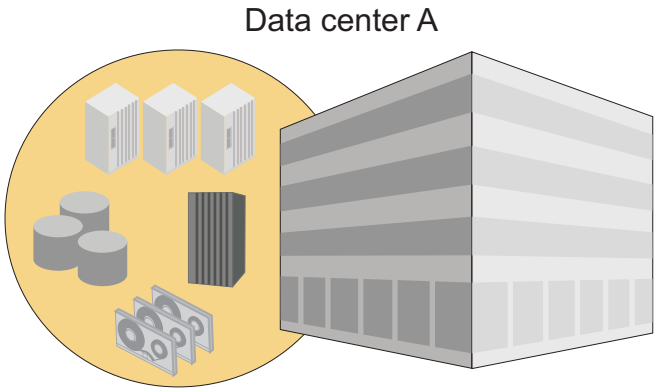


Figure 116. Tier 0: No disaster recovery capability

Tier 1: Offsite vaulting from a single production site

A tier-1 disaster-recovery strategy requires a disaster recovery plan and careful management of volumes that are stored in an offsite facility. To manage offsite volumes, generate a disaster recovery plan, and automate the Tivoli Storage Manager server recovery process, use disaster recovery manager.

As shown in the following figure, storage volumes, such as tape cartridges and media volumes, and are vaulted at an offsite location. Transportation is typically handled by couriers. If a disaster occurs, the volumes are sent back to the production site after hardware and the Tivoli Storage Manager server is restored.

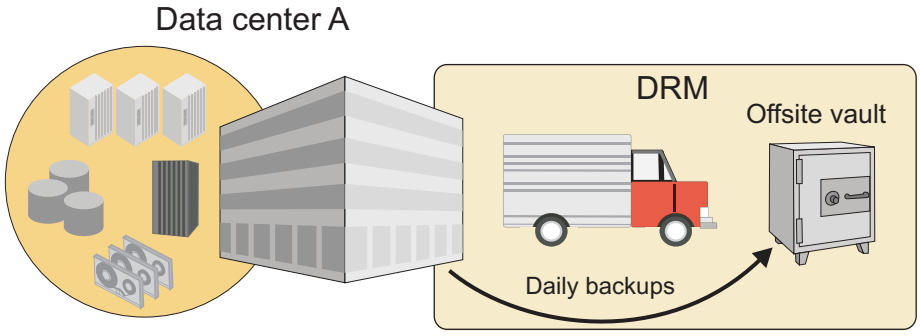


Figure 117. Tier 1: Offsite vaulting from a single production site

Because it is relatively inexpensive, a tier 1 strategy is used by many organizations. However, tier 1 capability is low and recovery time is slow. The recovery time depends on how long it takes to restore the hardware and software and to transport and load tapes into libraries. Before you implement a tier 1 strategy,

consider that an extended recovery time can impact business operations for several months or longer.

Tier 2: Offsite vaulting with a recovery site

A tier-2 disaster-recovery strategy includes a physically separate recovery site in addition to offsite vaulting. The recovery site has the hardware and software that are necessary to recover client data. To manage offsite volumes, generate a disaster recovery plan, and automate the server recovery process, use disaster recovery manager.

Like tier 1 organizations, tier 2 organizations rely on a courier to move storage volumes from the production site to an offsite storage facility. However, if a disaster occurs, tier 1 organizations move the data from the offsite storage facility to the recovery site. A Tier 2 disaster recovery strategy has low capability and is slow. See the following figure:

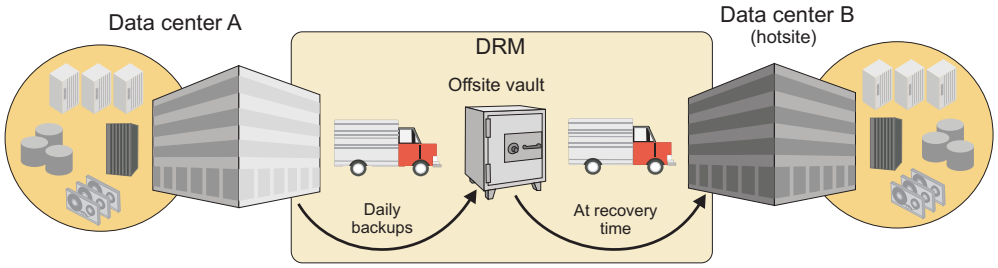


Figure 118. Tier 2: Offsite vaulting with a recovery site

A dedicated recovery site can reduce recovery time compared to the single production site in tier 1. The potential for data loss is also less. However, tier 2 architecture increases the cost of disaster recovery because more hardware and software must be maintained. The recovery site must also have hardware and software that are compatible with the hardware and software at the primary site. For example, the recovery site must have compatible tape devices and Tivoli Storage Manager server software. Before the production site can be recovered, the hardware and software at the recovery site must be set up and running. Transporting the storage volumes to the recovery site also affects recovery time.

Consider using Tivoli Storage Manager server-to-server communications for enterprise configuration of the Tivoli Storage Manager servers at the primary and recovery sites and routing commands from server to server.

Tier 3: Electronic vaulting of critical data

A tier-3 disaster-recovery strategy includes a recovery site with a running Tivoli Storage Manager server. Critical data is vaulted electronically from the production site to the recovery site. Disaster recovery manager is also used for offsite vaulting of non-critical data.

Electronic vaulting moves critical data offsite faster and more frequently than traditional courier methods. Recovery time is reduced because critical data is already stored at the recovery site. The potential for lost or misplaced data is also reduced. However, because the recovery site runs continuously, a tier 3 strategy is relatively more expensive than a tier 1 or a tier 2 strategy.

Critical data can include database backups, infrastructure setup files, and active client-backup data. To vault data electronically, consider using Tivoli Storage Manager virtual volumes over a TCP/IP connection. Data is stored as archive files on the server at the recovery site.

As shown in the following figure, the recovery site is physically separated from the production site. Often, the recovery site is a second data center that is operated by the same organization or by a storage service provider. If a disaster occurs at the primary site, storage media with the non-critical data are transported from the offsite storage facility to the recovery site.

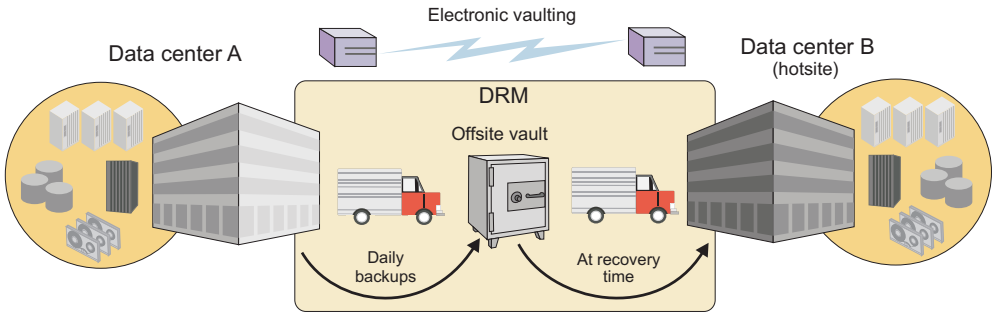


Figure 119. Tier 3: Electronic vaulting of critical data

If you implement a tier 3 strategy, you can use Tivoli Storage Manager server-to-server communications for enterprise configuration of the Tivoli Storage Manager servers and command routing.

Tier 4: Active data management at peer sites

In a tier-4 disaster-recovery strategy, data is actively managed at two physically separate sites. The servers at each site are set up in a peer-to-peer relationship. Critical data is replicated asynchronously. Copies of the data are available at both sites, and each server is able to recover the server at the alternate site. As part of this strategy, backup storage media are vaulted offsite and tracked by using disaster recovery manager.

As shown in the following figure, critical data is replicated the two sites by using high-bandwidth connections and data replication technology, for example, Peer-to-Peer Remote Connection (PPRC). Data is transmitted over long distances by using technologies such as extended storage area network (SAN), Dense Wave Division Multiplexing (DWDM), and IP/WAN channel extenders.

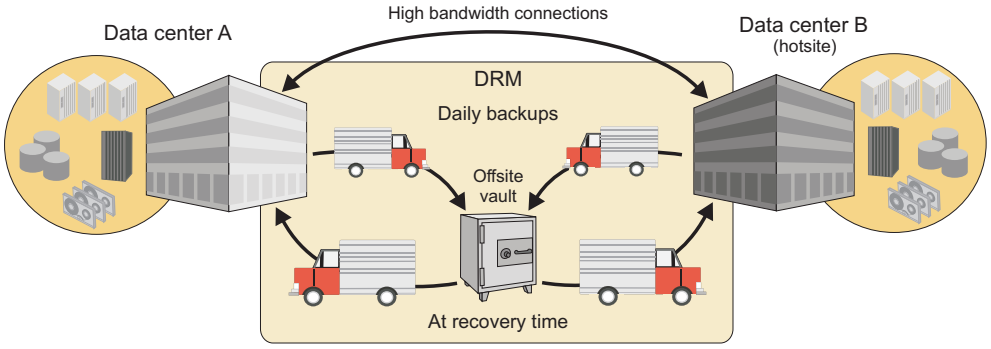


Figure 120. Tier 4: Active data management at peer-to-peer sites

Non-critical backups from both sites are moved to a single offsite storage facility. If a disaster occurs, the backup volumes are recovered by courier from the offsite vault and transported the designated recovery site.

If you implement a tier-4 disaster-recovery strategy, you can use Tivoli Storage Manager server-to-server communications for enterprise configuration of multiple Tivoli Storage Manager servers and command routing.

Recovery time for a tier 4 strategy is faster than the recovery time for a tier 1, tier 2, or tier 3 strategy. Recovery time is faster because hardware, software, and data are available or can be made available at two sites.

Tier 5: Synchronous replication

Tier 5 encompasses all the requirements of tier 4, including peer-peer sites. However, data, including database backups and copy storage pools, is replicated synchronously. Selected data is maintained in image status so that updates are applied to both the local and the remote copies of the database. Data, metadata, and the inventory information for the database are synchronized. The data at both the sites must be updated before the update request is considered successful. Physical media is not vaulted offsite.

As shown in the following figure, data is transmitted continuously between the two sites and synchronized by using high-bandwidth connections such as Peer-to-Peer Remote Copy (PPRC). Transmission over long distances is achieved by using technologies such as extended storage area network (SAN), Dense Wave Division Multiplexing (DWDM), and IP/WAN channel extenders. Copies of critical data are available at both sites, and each server is able to recover the server at the alternate site. With a minimum amount of data to recover, recovery time is reduced significantly.

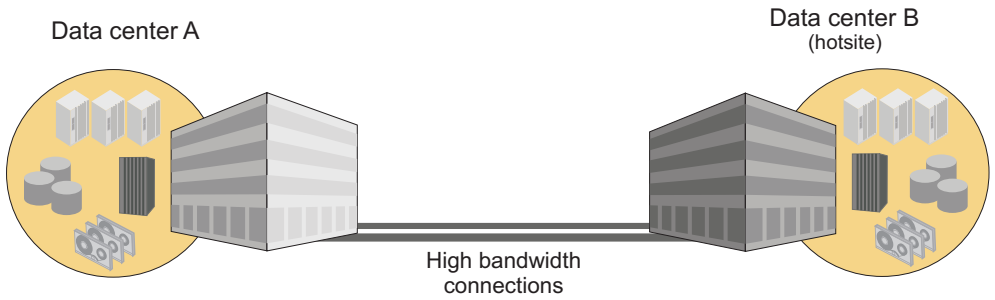


Figure 121. Tier 5: Synchronous replication

Copies of critical data are available at both sites, and each server is able to recover the server at the alternate site. Only the data transactions that are in-flight are lost during a disaster.

If you implement a tier-5 disaster-recovery strategy, you can also use Tivoli Storage Manager server-to-server communications to configure multiple Tivoli Storage Manager servers and command routing.

Tier 6: Synchronous replication with coupling or clustering applications

In Tier 6, local and remote copies of all data are updated synchronously, and dual online storage is used with a full network-switching capability. Systems are coupled with automated failover and failback capabilities when required.

As shown in the following figure, two sites are fully synchronized by using a high-bandwidth connection.

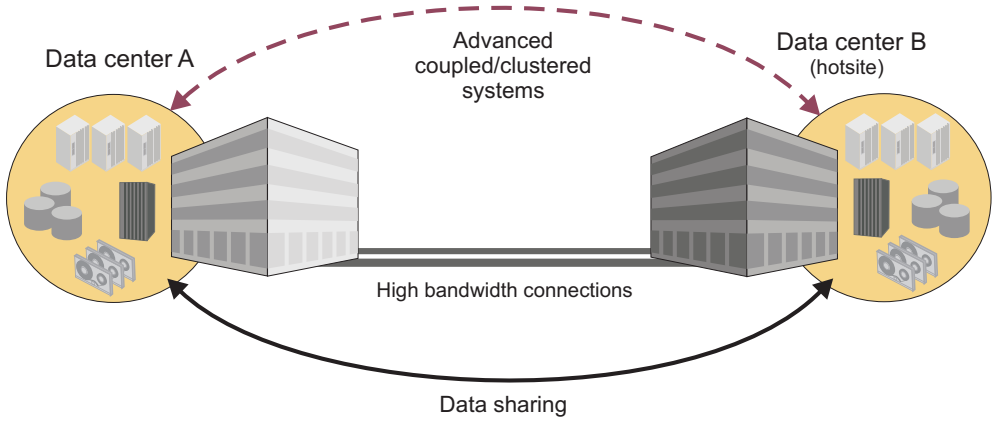


Figure 122. Tier 6: Synchronous replication with coupling or clustering applications

Tier 6 is the most expensive disaster recovery strategy because it requires coupling or clustering applications, additional hardware to support data sharing, and high-bandwidth connections over extended distances. However, this strategy also offers the fastest recovery time and the least amount of data loss. The typical length of time for recovery is normally a few minutes.

Part 7. Appendixes

© Copyright IBM Corp. 1993, 2013

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 121

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 'SCRTCH') must be returned in the response. When 'SCRTCH' 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:

© Copyright IBM Corp. 1993, 2013

```
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.

About this task

For each external library defined to the server, the following must occur during server initialization:

Procedure

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.

About this task

The server completes the following steps to process mount requests:

Procedure

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.

About this task

The server completes the following steps to process release requests:

Procedure

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.

About this task

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.

Procedure

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

The Tivoli Storage Manager server can mount volumes as needed. Alternatively, volumes can be manually mounted by issuing the **MOUNT** command.

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 that is 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 request must be in the following format:

```
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 `SCRTCH`.

accessmode
Specifies the access mode that is 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 must be specified 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
- GENERICTAPE

- LTO_ULTRIUM
- LTO_ULTRIUM2
- LTO_ULTRIUM3
- LTO_ULTRIUM4
- LTO_ULTRIUM5
- LTO_ULTRIUM6
- 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
- STK_T10000B
- STK_T10000C

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 volume number 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, the volume might have been ejected from the library and it must be returned to the library before the mount operation can proceed. The location information is posted by the agent so that the operator can obtain the volume and return it to the library.

The external program response is in the following format:

```
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 that is 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, set the value to `/dev/null`.

The external program must ensure that the special file is closed before the response is returned to the server.

resultcode
The result code can be one of the following values:

- 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/IBMadsm-s/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 Chapter 26, "Logging IBM Tivoli Storage Manager events to receivers," on page 797 and *Administrator's Reference*.

Sample user exit declarations

userExitSample.h contains declarations for a user-exit program.

The environment is:

Oracle Solaris

Figure 123. 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)   */
```

© Copyright IBM Corp. 1993, 2013

```
  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 elEventRecvData.eventType) *
 *******************************************/

#define TSM_SERVER_EVENT       0x03  /* Server Events */
#define TSM_CLIENT_EVENT       0x05  /* Client Events */

/***************************************************
 * Application Types (in elEventRecvData.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 elEventRecvData.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 124. Sample user exit program

```
/*****************************************************************************
 * Name:            userExitSample.c
 * Description:     Example user-exit program invoked by the server
 * Environment:     Solaris
 *****************************************************************************/

#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 82. 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 (YYYYMMDDDHHmmSS)
0048-0111	Server name (right padded with spaces)
0113-0176[1]	Node name
0178-0193[1]	Communications method name
0195-0258[1]	Owner name

Table 82. Readable text file exit (FILETEXTEXIT) format (continued)

Column	Description
0260-0323[1]	High-level internet address (n.n.n.n)
0325-0356[1]	Port number from high-level internet address
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.

Appendix C. Tivoli support information

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

From the IBM Support Portal at 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.

Communities and other learning resources

In addition to product documentation, many forms of assistance are available to help you get started as you deploy and use the Tivoli Storage Manager family of products. These resources can also help you to solve problems that you might have.

You can use forums, wikis, and other social media tools to ask questions, talk to experts, and learn from others.

User groups

Tivoli Global Storage Virtual User Group

Access this user group at http://www.tivoli-ug.org/storage.

This group makes it possible for individuals from many different industries and types of organizations to share information and work directly with the IBM product experts. Local chapters also exist where members meet in person to share experiences and hear from guest speakers.

ADSM.ORG

Access this mailing list at http://adsm.org.

This independently managed Storage Management discussion forum started when Tivoli Storage Manager was known as ADSTAR Distributed Storage Manager (ADSM). The members of this forum have many years of experience with Tivoli Storage Manager in almost every type of IT environment.

To subscribe to the forum, send an email to listserv@vm.marist.edu. The body of the message must contain the following text: `SUBSCRIBE ADSM-L` *your_first_name your_family_name*.

Tivoli Storage Manager community on Service Management Connect

Access Service Management Connect at http://www.ibm.com/developerworks/servicemanagement. In the Storage Management community of Service Management Connect, you can connect with IBM in the following ways:

- Become involved with transparent development, an ongoing, open engagement between users and IBM developers of Tivoli products. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
- Connect one-on-one with the experts to collaborate and network about Tivoli and the Tivoli Storage Manager community.
- Read blogs to benefit from the expertise and experience of others.

© Copyright IBM Corp. 1993, 2013

- Use wikis and forums to collaborate with the broader user community.

Tivoli Storage Manager wiki on developerWorks®

Access this wiki at https://www.ibm.com/developerworks/servicemanagement/sm/index.html.

Find the latest best practices, white papers, and links to videos and other resources. When you log on, you can comment on content, or contribute your own content.

Tivoli Support Technical Exchange

Find information about upcoming Tivoli Support Technical Exchange webcasts at http://www.ibm.com/software/sysmgmt/products/support/supp_tech_exch.html. Replays of previous webcasts are also available.

Learn from technical experts who share their knowledge and then answer your questions. The sessions are designed to address specific technical issues and provide in-depth but narrowly focused training.

Other social media sites

LinkedIn

You can join groups on LinkedIn, a social media site for professionals. For example:

- **Tivoli Storage Manager Professionals**: http://www.linkedin.com/groups/Tivoli-Storage-Manager-Professionals-54572
- **TSM**: http://www.linkedin.com/groups?gid=64540

Twitter

Follow `@IBMStorage` on Twitter to see the latest news about storage and storage software from IBM.

Tivoli education resources

Use these education resources to help you increase your Tivoli Storage Manager skills:

Tivoli Education and Certification website

View available education at http://www.ibm.com/software/tivoli/education.

Use the Search for Training link to find local and online offerings of instructor-led courses for Tivoli Storage Manager.

Education Assistant

Access resources at http://publib.boulder.ibm.com/infocenter/ieduasst/tivv1r0/index.jsp.

Scroll to view the list of available training videos. Recorded product demonstrations are also available on a YouTube channel.

Searching knowledge bases

If a problem occurs while you are using one of the Tivoli Storage Manager family of products, you can search several knowledge bases.

Begin by searching the Tivoli Storage Manager Information Center at http://pic.dhe.ibm.com/infocenter/tsminfo/v7r1. Within the information center, you can enter words, phrases, or message numbers in the **Search** field to find relevant topics.

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 the problem.

To search multiple Internet resources, go to the IBM support website at http://www.ibm.com/support/entry/portal/. 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 website, you can search various resources:

- IBM technotes.
- IBM downloads.
- IBM Redbooks® publications.
- IBM Authorized Program Analysis Reports (APARs). Select the product and click **Downloads** to search the APAR list.

Using IBM Support Assistant

IBM Support Assistant is a complimentary software product that can help you with problem determination. It is available for some Tivoli Storage Manager and Tivoli Storage FlashCopy Manager products.

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

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

You can find more information and download the IBM Support Assistant web page at http://www.ibm.com/software/support/isa.

You can also 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. Find add-ons for specific products at http://www.ibm.com/support/docview.wss?uid=swg27012689.

Finding product fixes

A product fix to resolve a software problem might be available from the IBM software support website.

Procedure

Determine what fixes are available by checking the IBM software support website at http://www.ibm.com/support/entry/portal/.

If you previously customized the site based on your product usage:

1. Click the link for the product, or a component for which you want to find a fix.
2. Click **Downloads**, and then click **Search for recommended fixes**.

If you have not previously customized the site:

Click **Downloads** and search for the product.

Receiving notification of product fixes

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

Procedure

1. From the support page at http://www.ibm.com/support/entry/portal/, click **Sign in** and sign in using your IBM ID and password. If you do not have an ID and password, click **register now** and complete the registration process.
2. Click **Manage all my subscriptions** in the Notifications pane.
3. Click the **Subscribe** tab, and then click **Tivoli**.
4. Select the products for which you want to receive notifications and click **Continue**.
5. Specify your notification preferences and click **Submit**.

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.

Procedure

1. Ensure that you have completed the following prerequisites:
   a. Set up a subscription and support contract.
   b. Determine the business impact of the problem.
   c. Describe the problem and gather background information.
2. Follow the instructions in "Submitting the problem to IBM Software Support" on page 1074.

Setting up and managing support contracts

You can set up and manage your Tivoli support contracts by enrolling in IBM Passport Advantage. The type of support contract that you need depends on the type of product you have.

Procedure

Enroll in IBM Passport Advantage in one of the following ways:

- **Online:** Go to the Passport Advantage website at http://www.ibm.com/software/lotus/passportadvantage/, click **How to enroll**, and follow the instructions.
- **By telephone:** For critical, system-down, or high-severity issues, you can call 1-800-IBMSERV (1-800-426-7378) in the United States. For the telephone 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 level	Description
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 gathering 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 re-created? 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 telephone.

Online

Go to the IBM Software Support website at 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 telephone

For critical, system-down, or severity 1 issues, you can call 1-800-IBMSERV (1-800-426-7378) in the United States. For the telephone 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**.

Appendix D. Accessibility features for the Tivoli Storage Manager product family

Accessibility features help users who have a disability, such as restricted mobility or limited vision to use information technology products successfully.

Accessibility features

The IBM Tivoli Storage Manager family of products includes the following accessibility features:

- Keyboard-only operation using standard operating-system conventions
- Interfaces that support assistive technology such as screen readers

The command-line interfaces of all products in the product family are accessible.

Tivoli Storage Manager Operations Center provides the following additional accessibility features when you use it with a Mozilla Firefox browser on a Microsoft Windows system:

- Screen magnifiers and content zooming
- High contrast mode

The Operations Center and the Tivoli Storage Manager Server can be installed in console mode, which is accessible.

The Tivoli Storage Manager Information Center is enabled for accessibility. For information center accessibility information, see "Accessibility features in the information center" (http://pic.dhe.ibm.com/infocenter/tsminfo/v7r1/topic/com.ibm.help.ic.doc/iehs36_accessibility.html).

Vendor software

The Tivoli Storage Manager product family includes certain vendor software that is not covered under the IBM license agreement. IBM makes no representation about the accessibility features of these products. Contact the vendor for the accessibility information about its products.

IBM and accessibility

See the IBM Human Ability and Accessibility Center (http://www.ibm.com/able) for information about the commitment that IBM has to accessibility.

© Copyright IBM Corp. 1993, 2013

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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

© Copyright IBM Corp. 1993, 2013

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758
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 is for planning purposes only. The information herein is subject to change before the products described become available.

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® 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 and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

LTO and Ultrium are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Glossary

This glossary provides terms and definitions for Tivoli Storage Manager, Tivoli Storage FlashCopy Manager, and associated products.

The following cross-references are used in this glossary:

- *See* refers you from a nonpreferred term to the preferred term or from an abbreviation to the spelled-out form.
- *See also* refers you to a related or contrasting term.

For other terms and definitions, see the IBM Terminology website at www.ibm.com/software/globalization/terminology.

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 mode, 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.

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.

ACK See acknowledgment.

acknowledgment (ACK)
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. See also server storage, storage pool, storage pool volume.

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. See also inactive file system.

active policy set
The activated policy set that contains the policy rules currently in use by all client nodes assigned to the policy domain. See also policy domain, 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. See also backup version, 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 server. See also 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 central scheduler, client schedule, schedule.

administrative privilege class
See privilege class.

© Copyright IBM Corp. 1993, 2013

administrative session
A period of time during which an administrator user ID communicates with a server to perform administrative tasks. See also client node session, session.

administrator
A person responsible for administrative tasks such as access authorization and content management. Administrators can also grant levels of authority to users.

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, 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.

application client
A program that is installed on a system to protect an application. The 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. See also 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. See also copy group.

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
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.

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 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 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 demand migration, threshold migration.

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. See also 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. See also copy group.

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 storage. More than one backup version can exist in storage, but only one backup version is the active version. See also active version, copy group, inactive version.

bind To associate a file with a management class name. See also archive-retention grace period, management class, rebind.

C

cache To place a duplicate copy of a file on random access media when the server migrates a file to another storage pool in the hierarchy.

cache file
A snapshot of a logical volume created by Logical Volume Snapshot Agent. Blocks are saved immediately before they are modified during the image backup and their logical extents are saved in the cache files.

CAD See client acceptor daemon.

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 also administrative command schedule, client schedule.

client A software program or computer that requests services from a server. See also server.

client acceptor
A 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.

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. See also administrative session.

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 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-polling scheduling mode
A method of operation in which the client queries the server for work. See also 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, central scheduler, 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. See also client user-options file, 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. See also client system-options file, options file.

closed registration
A registration process in which only an administrator can register workstations as client nodes with the server. See also 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 to be consistent.

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, strings of characters, or binary data from the data being processed and replaces characters with control characters. Compression reduces the amount of storage space that is required for 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 enterprise configuration, managed server, profile.

conversation
A connection between two programs over a session that allows them to communicate with each other while processing a transaction. See also session.

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,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 destination, primary storage pool, server storage, storage pool, storage pool volume.

D

daemon
A program that runs unattended to perform continuous or periodic functions, such as network control.

damaged file
A physical file in which read errors have been detected.

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. See also database snapshot, full backup.

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, full backup.

data center
In a virtualized environment, a container that holds hosts, clusters, networks, and data stores.

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.

data store
In a virtualized environment, the location where virtual machine data is stored.

deduplication
The process of creating representative records from a set of records that have been identified as representing the same entities.

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.

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. See also automatic migration, selective migration, threshold migration.

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. See also copy storage pool.

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 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.
2. 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.

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 recover 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 also policy domain.

DRM See disaster recovery manager.

DSMAPI
See data storage-management application-programming interface.

dynamic serialization
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. See also shared dynamic serialization, shared static serialization, static serialization.

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, subscription.

enterprise logging
The process of sending events from a 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 An occurrence of significance to a task or system. Events can include completion or failure of an operation, a user action, or the change in state of a process. See also enterprise logging, receiver.

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, real and virtual storage capacities and channel volume, 2 to the power of 60 or 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, from space management, or from both backup and space management.

exclude-include list
See include-exclude list.

expiration
The process by which files, data sets, or objects are identified for deletion because their expiration date or retention period has passed.

expiring file
A migrated or premigrated file that has been marked for expiration and removal from storage. If a stub file or an original copy of a premigrated file is deleted from a local file system, or if the original copy of a premigrated file is updated, the corresponding migrated or premigrated file is marked for expiration the next time reconciliation is run.

extend
To increase the portion of available space that can be used to store database or recovery log information.

extended attribute (EA)
Names or value pairs that are associated with files or directories. There are three

classes of extended attributes: user attributes, system attributes, and trusted attributes.

external library
A collection of drives that is managed by the media-management system other than the storage management server.

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 migrated file, premigrated file, resident 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, database snapshot, incremental backup.

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.

G

GB See gigabyte.

General Parallel File System (GPFS)
A high-performance shared-disk file system that can provide data access from nodes in a clustered system environment. See also information lifecycle management.

gigabyte (GB)
For processor storage, real and virtual storage, and channel volume, 10 to the

power of nine or 1,073,741,824 bytes. For disk storage capacity and communications volume, 1,000,000,000 bytes.

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.

Globally Unique Identifier (GUID)
An algorithmically determined number that uniquely identifies an entity within a system. See also Universally Unique Identifier.

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. See also hierarchical storage management client, recall, storage hierarchy.

hierarchical storage management client (HSM client) A client program that works with the server to provide hierarchical storage management (HSM) for a system. See also hierarchical storage management, management class.

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. See also 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. See also active version, backup 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. See also include-exclude file.

incremental backup
The process of backing up files or directories, or copying pages in the database, that are new or changed since the last full or incremental backup. See also selective backup.

individual mailbox restore
See mailbox restore.

information lifecycle management (ILM)
A policy-based file-management system for storage pools and file sets. See also General Parallel File System.

inode The internal structure that describes the individual files on AIX, UNIX, or Linux systems. An inode contains the node, type, owner, and location of a file.

inode number
A number specifying a particular inode file in the file system.

IP address
A unique address for a device or logical unit on a network that uses the Internet Protocol 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. See also migration job.

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

KB See kilobyte.

kilobyte (KB)
For processor storage, real and virtual storage, and channel volume, 2 to the power of 10 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.

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
1. Pertaining to a device, file, or system that is accessed directly from a user system, without the use of a communication line. See also remote.
2. For hierarchical storage management products, pertaining to the destination of migrated files that are being moved. See also remote.

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 volume
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, physical file, physical occupancy.

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. See also physical occupancy.

logical unit number (LUN)
In the Small Computer System Interface (SCSI) standard, a unique identifier 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.

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. See also Tivoli Storage Manager command script.

mailbox restore
A function that restores Microsoft Exchange Server data (from IBM Data Protection for Microsoft Exchange backups) at the mailbox level or mailbox-item level.

managed object
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.

managed server
A 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, enterprise configuration, profile, subscription.

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 bind, copy group, hierarchical storage management client, policy set, rebind.

maximum transmission unit (MTU)
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.

media server
In a z/OS® environment, a program that provides access to z/OS disk and tape storage for Tivoli Storage Manager servers that run on operating systems other than z/OS.

megabyte (MB)
For processor storage, real and virtual storage, and channel volume, 2 to the 20th power or 1,048,576 bytes. For disk storage capacity and communications volume, 1,000,000 bytes.

metadata
Data that describes the characteristics of data; descriptive data.

migrate
To move data to another location, or an application to another computer system.

migrated file
A file that has been copied from a local file system to 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 file state, premigrated file, resident file, stub file.

migration
The process of moving data from one computer system to another, or an application to another computer system.

migration job
A specification of files to migrate, and actions to perform on the original files after migration. See also job file, threshold migration.

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 disks at the same time. The mirroring of data protects it against data loss within the database or 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 also absolute mode, modified 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, 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
A logical drive through which volumes are accessed in a sequential access device class. For removable media device types, such as tape, a mount point is a logical drive associated with a physical drive. For the file device type, a mount point is a logical drive associated with an I/O stream. 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 file server
See network-attached storage file server.

NAS file server node
See NAS node.

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 server. See also non-native data format.

NDMP
See Network Data Management Protocol.

NetBIOS (Network Basic Input/Output System)
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-attached storage file server (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
See NetBIOS.

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. See also native format.

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. See also 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. See also client system-options file, client user-options file.

originating file system
The file system from which a file was migrated. When a file is recalled, it is returned to its originating file system.

orphaned stub file
A file for which no migrated file can be found on the 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.

P

packet In data communication, a sequence of binary digits, including data and control signals, that are 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.

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, logical file, physical occupancy.

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 logical file, logical occupancy, physical file.

plug-in
A separately installable software module that adds function to an existing program, application, or interface.

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. See also active policy set, 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, management class.

premigrated file
A file that has been copied to server 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 server storage. Premigrated files occur on UNIX and Linux file systems to which space management has been added. See also file state, migrated file, resident file.

premigrated files database
A database that contains information about each file that has been premigrated to server storage.

premigration
The process of copying files that are eligible for migration to server 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 copy storage pool, server storage, storage pool, storage pool volume.

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 authority, node privilege class, operator privilege class, policy privilege class, storage privilege class, 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, enterprise configuration, managed server.

profile association
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.

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.

rebind
To associate all backed-up versions of a file with a new management class name. For example, a file that has an active backup version is rebound when a later version of the file is backed up with a different management class association. See also bind, management class.

recall To copy a migrated file from server storage back to its originating file system using the hierarchical storage management client. See also selective recall.

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 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 ensuring consistency between the original data repository and the larger system where the data is stored for backup. Examples of larger systems where the data is stored for backup are storage servers or other storage systems. During the reconciliation process, data that is identified as no longer needed is removed.

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.

remote
For hierarchical storage management products, pertaining to the origin of migrated files that are being moved. See also local.

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 server 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 server storage and modified. See also file state.

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. See also archive.

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 also administrative command schedule, 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. See also volume.

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. See also 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. See also incremental backup.

selective migration
The process of copying user-selected files from a local file system to server storage and replacing the files with stub files on the local file system. See also demand migration, threshold migration.

selective recall
The process of copying user-selected files from server storage to a local file system. See also recall, transparent recall.

serialization
The process of handling files that are modified during backup or archive processing. See also shared dynamic serialization, shared static serialization, static serialization.

server A software program or a computer that provides services to other software programs or other computers. See also client.

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. See also 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 hierarchical storage management client nodes (space-managed files). See also active-data pool, copy storage pool, primary storage pool, storage pool volume, 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 for the duration of the session. See also administrative session.

session resource usage
The amount of wait time, processor time, and space that is used or retrieved during a client session.

shadow copy
A snapshot of a volume. The snapshot can be taken while applications on the system continue to write data to the volumes.

shadow volume
The data stored from a snapshot of a volume. The snapshot can be taken while applications on the system continue to write data to the volumes.

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. The backup-archive client retries the backup or archive operation a number of times; if the file is being modified during each attempt, the backup-archive client will back up or archive the file on its last try. See also dynamic serialization, serialization, shared static serialization, static serialization.

shared library
A library device that is used by multiple storage manager servers. See also library.

shared static serialization
A copy-group serialization value that specifies that a file must not be modified during a backup or archive operation. The client 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 dynamic serialization, serialization, shared dynamic serialization, 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 hierarchical storage management (HSM) client. The HSM client recalls the file to the client node on demand.

space management
See hierarchical storage management.

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. Stanzas are usually separated by blank lines or colons, and each stanza has a name.

startup window
A time period during which a schedule must be initiated.

static 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 backup-archive client cannot back up or archive the file. See also dynamic serialization, serialization, shared dynamic serialization, 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 tailored to a specific environment, combining servers, systems, storage products, networking products, software, and services.

storage hierarchy
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.

storage pool
A named set of storage volumes that is the destination that is used to store client

data. See also active-data pool, copy storage pool, primary storage pool, storage hierarchy.

storage pool volume
A volume that has been assigned to a storage pool. See also active-data pool, copy storage pool, primary storage pool, server storage, volume.

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 server storage. It also contains additional information that can be used to eliminate the need to recall a migrated file. See also migrated file, resident 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 server 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 storage environment, the process of identifying the subscribers to which the profiles are distributed. See also enterprise configuration, managed server.

system privilege class
A privilege class that gives an administrator the authority to issue all server commands. See also privilege class.

T

tape library
A set of equipment and facilities that support an installation's tape environment. The tape library can include tape storage racks, mechanisms for automatic tape mounting, a set of tape drives, and a set of related tape volumes mounted on those drives.

tape volume prefix
The high-level-qualifier of the file name or the data set name in the standard tape label.

target node
A client node for which other client nodes (called agent nodes) have been granted proxy authority. The proxy authority allows the agent nodes to perform operations such as backup and restore on behalf of the target node, which owns the data.

TCA See trusted communications agent.

TCP/IP
See Transmission Control Protocol/Internet Protocol.

threshold migration
The process of moving files from a local file system to server storage based on the high and low thresholds that are defined for the file system. See also automatic migration, demand migration, migration job, selective migration.

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.

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. See also macro file, script.

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. See also communication method.

transparent recall
The process that is used to automatically recall a migrated file to a workstation or file server when the file is accessed. See also 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.

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 many classical and historical texts.

Unicode-enabled file space
Unicode file space names provide support for multilingual workstations without regard for the current locale.

Universally Unique Identifier (UUID)
The 128-bit numeric identifier that is used to ensure that two components do not have the same identifier. See also Globally Unique Identifier.

Universal Naming Convention (UNC)
The server name and network name combined. These names together identify the resource on the domain.

UTF-8 Unicode Transformation Format, 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. See also UCS-2.

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 mount point
A directory branch of a file system that is defined as a virtual file system. The virtual file system is backed up to its own file space on the server. The server processes the virtual mount point as a separate file system, but the client operating system does not.

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, server storage, storage pool, 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 (VSS)
A set of Microsoft application-programming interfaces (APIs) that are used 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 Microsoft Exchange data. This copy can be stored on local shadow volumes or on Tivoli Storage Manager server storage.

VSS Fast Restore
An operation that restores data from a local snapshot. The snapshot is the VSS backup that resides on a local shadow volume. The restore operation retrieves the data by using a file-level copy method.

VSS Instant Restore
An operation that restores data from a local snapshot. The snapshot is the VSS backup that resides on a local shadow volume. The restore operation retrieves the data by using a hardware assisted restore method (for example, a FlashCopy operation).

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 Microsoft 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 Microsoft 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.

workload partition (WPAR)
A partition within a single operating system instance.

workstation
A terminal or personal computer at which a user can run applications and that is usually connected to a mainframe or a network.

worldwide name (WWN)
A 64-bit, unsigned name identifier that is unique.

WPAR See workload partition.

WWN See worldwide name.

Index

Special characters

$$CONFIG_MANAGER$$ 710

Numerics

3480 tape drive
  cleaner cartridge 163
  device support 40
  mixing drive generations 180
3490 tape drive
  cleaner cartridge 163
  device support 40
  mixing drive generations 180
3494 automated library device 42
3494 library 109
  configuration with a single drive device 108
  migrating 117
3494SHARED server option 67
3570 tape drive
  ASSISTVCRRECOVERY server option 67
  device support 40
3590 tape drive
  defining device class 63, 174, 175
  device support 43
3592 drives and media
  as element of storage hierarchy 256
  cleaning 162
  data encryption 155, 182, 530
  defining device class 63
  DEVICETYPE parameter 133
  enabling for WORM media 138
  mixing drive generations 180
4mm tape device support 175
8mm tape device support 175

A

absolute mode, description of 497
access authority, client 440, 441
access mode, volume
  changing 254
  description 255
  determining for storage pool 241, 406
access rights
  setting
    before server startup 589
access, managing 821, 834
accessibility features 1075
accounting record
  description of 793
  monitoring 793
accounting variable 793
ACSLS (Automated Cartridge System Library Software)
  StorageTek library 41
  description 44
  mixing 3592 drive generations 180
  Tivoli Storage Manager server options for 67
ACSLS library 106
ACTIVATE POLICYSET command 504
active data 932
active data, protecting with active-data pools 237
active files, storage-pool search order 239
active log 865
  description 637
  increasing the size 668
  move to another directory 671
  out of space 668
  space requirements 646
active log mirror 865
  description 638
active log size
  reducing the size 669
ACTIVE policy set
  creating 494, 504
  replacing 472
active-data pool
  auditing volumes in 878
  backup-set file source 537
  collocation on 360
  defining 406
  export file source 737, 746, 747
  import operations 756
  overview 237, 261
  reclamation of 370
  simultaneous-write function 327
  specifying in policy definition 492
  storage pool search-and-selection order 239
ACTIVELOGDIRECTORY server option 668, 671
ACTIVELOGSIZE 669
ACTIVELOGSIZE server option 668, 669
activity log
  description of 789
  logging events to 799
  monitoring 789
  querying 790
  setting size limit 791
  setting the retention period 791
adding space to the database 664
administrative client
  description of 3
  viewing information after IMPORT or EXPORT 761
administrative clients
  preventing from accessing the server 734
administrative commands
  ACCEPT DATE 600
  ASSIGN DEFMGMTCLASS 504
  AUDIT LIBVOLUME 147
  AUDIT LICENSE 581
  AUDIT VOLUME 882
  BACKUP NODE 224, 226
  BEGIN EVENTLOGGING 799
  CANCEL PROCESS 601
  CANCEL RESTORE 466
  CANCEL SESSION 463
  CHECKIN LIBVOLUME 133
  CHECKOUT LIBVOLUME 145
  CLEAN DRIVE 160
  COMMIT 630
  COPY ACTIVEDATA 239, 261
  COPY CLOPTSET 461
  COPY DOMAIN 494

© Copyright IBM Corp. 1993, 2013

administrative commands *(continued)*
COPY POLICYSET 494
COPY SCHEDULE 562
COPY SCRIPT 624
COPY SERVERGROUP 720
DEFINE ASSOCIATION 556
DEFINE BACKUPSET 542
DEFINE CLIENTACTION 575
DEFINE CLIENTOPT 575
DEFINE CLOPTSET 458
DEFINE COPYGROUP 496, 501, 502
DEFINE DATAMOVER 171, 220
DEFINE DEVCLASS
3592 180
FILE device classes 184
LTO device classes 188
REMOVEABLEFILE device classes 183
SERVER device classes 191
tape device classes 175
VOLSAFE device classes 192
DEFINE DRIVE 170
DEFINE GRPMEMBER 719
DEFINE LIBRARY 42, 169
DEFINE MACHINE 986
DEFINE MACHNODEASSOCIATION 986
DEFINE PATH 172
DEFINE POLICYSET 494, 495
DEFINE PROFASSOCIATION 697, 698
DEFINE PROFILE 697
DEFINE RECMEDMACHASSOCIATION 989
DEFINE RECOVERYMEDIA 988
DEFINE SCHEDULE 611
DEFINE SCRIPT 617
DEFINE SERVER 684, 716, 722
DEFINE SERVERGROUP 719
DEFINE STGPOOL 245, 247, 257, 258
DEFINE SUBSCRIPTION 709
DEFINE VIRTUALFSMAPPING 232
DEFINE VOLUME 50, 252
DELETE ASSOCIATION 565
DELETE BACKUPSET 546
DELETE COPYGROUP 526
DELETE DOMAIN 527
DELETE DRIVE 165
DELETE EVENT 567
DELETE GRPMEMBER 721
DELETE KEYRING 827
DELETE LIBRARY 153
DELETE MGMTCLASS 526
DELETE POLICYSET 526
DELETE PROFASSOCIATION 702
DELETE PROFILE 704
DELETE SCHEDULE 562
DELETE SCRIPT 626
DELETE SERVER 690
DELETE SERVERGROUP 721
DELETE STGPOOL 409
DELETE SUBSCRIBER 714
DELETE SUBSCRIPTION 704, 710
DELETE VOLHISTORY 606
DELETE VOLUME 411, 412
DISABLE EVENTS 798
DISABLE SESSIONS 465
DSMSERV DISPLAY DBSPACE 661
DSMSERV DISPLAY LOG 661
ENABLE EVENTS 798
ENABLE SESSIONS 465

administrative commands *(continued)*
END EVENTLOGGING 799
EXPIRE INVENTORY 31
EXPORT ADMIN 731
EXPORT NODE 745
EXPORT POLICY 745
EXPORT SERVER 745
EXTEND DBSPACE 662, 666
GENERATE BACKUPSET 538
GRANT AUTHORITY 835
HALT 598
HELP 607
IMPORT 760, 761
IMPORT ADMIN 748
IMPORT NODE 748, 757
IMPORT POLICY 748
IMPORT SERVER 748, 757
LABEL LIBVOLUME 125
LOCK ADMIN 842
LOCK NODE 434
LOCK PROFILE 701, 702
MOVE DATA 397
MOVE NODEDATA 402
NOTIFY SUBSCRIBERS 702
PING SERVER 722
PREPARE 989
QUERY ACTLOG 790
QUERY BACKUPSETCONTENTS 545
QUERY CONTENT 384
QUERY COPYGROUP 523, 755
QUERY DB 661
QUERY DBSPACE 661
QUERY DEVCLASS 195
QUERY DOMAIN 525
QUERY DRIVE 153
QUERY DRMSTATUS 980
QUERY ENABLED 812
QUERY EVENT 557
QUERY FILESPACE 456
QUERY LIBRARY 151
QUERY LICENSE 581
QUERY MGMTCLASS 524
QUERY NODE 438
QUERY NODEDATA 394
QUERY OCCUPANCY
backed-up, archived, and space-managed files 395
client file spaces 393
client nodes 393
device classes 394
storage pools 394
QUERY OPTION 782
QUERY POLICYSET 525
QUERY PROCESS 400
QUERY RESTORE 466
QUERY RPFCONTENT 991
QUERY RPFILE 991
QUERY SCHEDULE 557
QUERY SCRIPT 625
QUERY SERVERGROUP 720
QUERY STGPOOL 378, 388, 751
QUERY SUBSCRIPTION 709
QUERY SYSTEM 783
QUERY VOLUME 380, 401
RECONCILE VOLUMES 729
REGISTER ADMIN 837
REGISTER LICENSE 580
REMOVE ADMIN 841

administrative commands *(continued)*
REMOVE NODE 434
RENAME ADMIN 840
RENAME FILESPACE 759
RENAME NODE 433
RENAME SCRIPT 626
RENAME SERVERGROUP 720
RENAME STGPOOL 405
RESTORE DB 599
RESTORE NODE 224, 226
RESTORE STGPOOL 904
ROLLBACK 631
RUN 626
SELECT 784
SET ACCOUNTING 793
SET AUTHENTICATION 855
SET CLIENTACTDURATION 575
SET CONFIGMANAGER 693, 696
SET CONFIGREFRESH 709
SET CONTEXTMESSAGING 798
SET CROSSDEFINE 685, 688
SET DBREPORTMODE 661
SET DRMCHECKLABEL 983
SET DRMCOPYSTGPOOL 980
SET DRMCOURIERNAME 983
SET DRMDBBACKUPRXPIREDAYS 983
SET DRMFILEPROCESS 983
SET DRMINSTPREFIX 980
SET DRMNOTMOUNTABLE 983
SET DRMPLANPOSTFIX 980
SET DRMPLANPREFIX 980
SET DRMPRIMSTGPOOL 980
SET DRMRPFEXPIREDAYS 991
SET DRMVAULTNAME 983
SET EVENTRETENTION 567, 616
SET INVALIDPWLIMIT 853
SET LICENSEAUDITPERIOD 581
SET MAXCMDRETRIES 574
SET MAXSCHEDSESSIONS 571
SET MINPWLENGTH 854
SET PASSEXP 850
SET QUERYSCHEDPERIOD 573
SET RANDOMIZE 572
SET REGISTRATION 417
SET RETRYPERIOD 574
SET SCHEDMODES 569
SET SERVERHLADDRESS 685, 688
SET SERVERLLADDRESS 685, 688
SET SERVERNAME 684, 688, 782
SET SERVERPASSWORD 684, 685, 688
SET SUBFILE 546
SET SUMMARYRETENTION 788
SETOPT 605
UNLOCK PROFILE 701, 702
UPDATE ADMIN 837
UPDATE ARCHIVE 550
UPDATE BACKUPSET 544
UPDATE CLIENTOPT 461
UPDATE CLOPTSET 461
UPDATE COPYGROUP 496, 502
UPDATE DEVCLASS 175
UPDATE DOMAIN 494
UPDATE DRIVE 154
UPDATE LIBRARY 152
UPDATE LIBVOLUME 50, 145
UPDATE MGMTCLASS 495
UPDATE NODE 428
administrative commands *(continued)*
UPDATE POLICYSET 494
UPDATE RECOVERYMEDIA 989
UPDATE SCHEDULE 611
UPDATE SCRIPT 623, 624
UPDATE SERVER 690
UPDATE SERVERGROUP 720
UPDATE VOLUME 252
VALIDATE LANFREE 126
administrative privilege class
description 835
granting authority 835
reducing 840
revoking all 840
administrative user ID
creating automatically 442
description of 416
preventing automatic creation of 442
administrative web interface
description 17
administrative Web interface
limitation of browser for script definitions 617
administrator
authorizing to manage a policy domain 835
locking 842
managing registration 579
querying 838
registering 837
removing 841
renaming 840
restrictions when registering 837
unlocking 842
updating 837, 840
viewing information about 838
administrators
managing 834
aggregates
controlling the size of 258
estimating size 384
RECONSTRUCT parameter 401
reconstructing 364, 371, 401
viewing information about 386, 393
ANR8914I message 164
ANR9999D message 798
application client
adding node for 416
description 4
policy for 517
application program interface (API)
client, registering 420
compression option 420
deletion option 420
registering to server 420
simultaneous-write function, version support for 329
application programming interface (API)
description of 3
ARCHFAILOVERLOGDIRECTORY server option 674
archive
allowing while file changes 502
backup set, uses for 8, 12
determining storage usage 395
directory 549
instant 8, 12
package 549
policy, defining 491
policy, introduction 26
process description 489

archive *(continued)*
storage usage, minimizing 549, 550
storage usage, reducing 549, 550
uses for 8, 12
archive copy group 26
defining 502, 504
deleting 526
description of 476
archive data
expiration 508
managing 548
protection 508
archive failover log 865
description 639
move to another directory 671
archive log 865
description 638
move to another directory 671
space requirements 646
archiving
file 475, 489
file management 475
FILE-type volume, archiving many small objects to 185
ASCII restriction for browser script definition 617
ASSIGN DEFMGMTCLASS command 504
association, client with schedule
defining 556
deleting 565
association, file with management class 483, 484
association, object with profile
administrative command schedule 701
administrator 697, 713
client option set 698
deleting 702
policy domain 699
script 698
AUDIT LIBVOLUME command 147
AUDIT LICENSE command 581
AUDIT VOLUME command 875, 882
auditing
LDAP directory server 902
library's volume inventory 147
license, automatic by server 581
multiple volumes in sequential access storage pool 882
single volume in sequential access storage pool 883
volume in disk storage pool 882
volume, reasons for 875
volumes by date 883
volumes by storage pool 884
authority
client access 441
granting to administrators 835
privilege classes 835
server options 835
authorizing to start server
root user IDs 591
autoconf utility 85
AUTOFSRENAME parameter 450
AUTOLABEL parameter for tape volumes 132
Automated Cartridge System Library Software (ACSLS)
StorageTek library
description 44
mixing 3592 drive generations 180
Tivoli Storage Manager server options for 67
automated library device
auditing 147
changing volume status 145
automated library device *(continued)*
checking in volumes 133
defining 42
informing server of new volumes 133
labeling volumes 131
overflow location 241
removing volumes 145
scratch and private volumes 51
updating 152
volume inventory 51
automatic deployment 428
verifying 433
automatic failover
overview 908
automatic failover scenario 956
automatically renaming file spaces 450
automating
administrative commands 31
client operations 554
server operations 610
server startup 593
awk script 987, 1018

B

background mode 594
background processes 600
backup
amount of space used by client 395
comparison of types 10, 12
default policy 471
defining criteria for client files 491
differential, for NAS node 9, 56
file 474, 486, 488
file management 474
file while open 496
FILE-type volume, backing up small objects to 185
frequency for file 497
full, for NAS node 56
group 11
incremental 474, 486
logical volume 489
NAS file server 209
NAS file server to Tivoli Storage Manager server 228
NAS filer to Tivoli Storage Manager server 227
policy 26
selective 474, 488
SnapMirror to Tape 233
snapshot, using hardware 9, 11
subfiles, server set-up 24, 546
types available 10, 12
BACKUP command 670
backup copy group 26
defining 496, 501
deleting 526
description of 476
frequency 486
mode 486
serialization 486
backup data, protecting active 237
BACKUP DEVCONFIG command 867
backup media
move 994
backup period, specifying for incremental 571
backup set
adding subfiles to 548
deleting 546

backup set *(continued)*
description of 538
displaying contents of 545
example of generating 540
generating 538
how the server manages and tracks 543
media, selecting 539
moving to other servers 542
OST extension on 539
selecting a name for 539
selecting retention period for 543
suggested usage 8, 24
updating 544
use as archive 12, 14, 24
backup sets
displaying information 544
generate to a point-in-time 540
generate with multiple data types 541
backup sets, managing 543
BACKUP VOLHISTORY command 866
backup volumes
create single set 541
backup-archive client
description of 3
operations summary 10
performing operations for 529, 561, 568
policy for 480
registering node 416
scheduling operations for 554
using to back up NAS file server 201, 226
bar-code reader
auditing volumes in a library 147
barcode reader
checking in volumes for a library 137
labeling volumes in a library 132
base file 546
batch file, scheduling on client 558
binding a file to a management class 483
browser, limited to ASCII entry for script definition 617

C

cache
deleting files from 280, 397
description of 21
disabling for disk storage pools 280
effect on performance 280
effect on statistics 281
enabling for disk storage pools 241, 280
monitoring utilization on disk 391
calculate replication 930
CANCEL PROCESS command 389, 601
CANCEL RESTORE command 466
CANCEL SESSION command 463
capacity planning
database space requirements
estimates based on number of files 642
estimates based storage pool capacity 644
starting size 642
recovery log space requirements
active and archive logs 646
active log mirror 658
archive failover log 659
archive log space for database reorganization 658
capacity, tape 196
cartridge
cleaner cartridge 163
cartridge *(continued)*
device support 40
mixing drive generations 180
categories, 349X library 42
Celerra
file server integrated checkpoints 234
Centera libraries 99
Centera SDK
installing 99
Centera storage device
concurrent access 194
overview 46
restore improve 194
unsupported functions 245
unsupported server operations 194
Centera storage pool, backing up 874
central monitoring 679
Central Processing Unit
CPU 582
central scheduling
client operations 529, 553, 561, 568
controlling the workload 571
coordinating 568
description of 28, 31, 553
server operations 610
certificate
adding to the key database 826, 827
homegrown certificate authority 827
changing date and time on server 600
changing hostname 604
characteristics, machine 987
check in
cleaner cartridge 162
library volume 133
setting a time interval for volume 179
VolSafe-enabled volumes 192
CHECKIN LIBVOLUME command 133
checking the log file generated by processed schedules 567
checklist for DRM project plan 1013
CHECKOUT LIBVOLUME command 145
CHECKTAPEPOS server option 67
class, administrator privilege
description 835
granting authority 835
reducing 840
revoking all 840
class, device
3590 174, 175
3592 175
4MM 174, 175
8MM 174, 175
amount of space used 394
CENTERA 46
defining 174
description of 19
DISK 174
DLT 174, 175
ECARTRIDGE 175
FILE 174
FORMAT parameter 177
GENERICTAPE 174, 175
LTO 188
REMOVABLEFILE 183
requesting information about 195
selecting for import and export 744
sequential 175
SERVER 174, 175, 724

class, device *(continued)*
StorageTek devices 175, 192
tape 175, 183
Ultrium, LTO 175
updating 175, 183
VOLSAFE 192
WORM 174, 175
class, policy privilege
description 834, 835
granting 839
revoking 839, 840
class, storage privilege
description 835
granting 839
reducing 839
revoking 840
CLEAN DRIVE command 160
cleaner cartridge
checking in 162
how often to use 161
operations with 163
restrictions on cleaning 161
CLEANFREQUENCY parameter 161
client
access user ID 441
administrative 3
API (application program interface) 420
API (application programming interface) 4
application client 4, 517
backup-archive 3
how to protect 7
operations summary 10
options file 421
restore without primary volumes available 901
Tivoli Storage Manager for Space Management (HSM client) 4, 480
using to back up NAS file server 209, 226
client file
allowing archive while changing 471
allowing backup while changing 471, 496
archive package 549
associating with management class 483, 484
damaged 901
delaying migration of 274
deleting 410
deleting from a storage pool 409
deleting from cache 280
deleting when deleting a volume 410
duplication when restoring 901
eligible for archive 471, 486
eligible for backup 471, 486
eligible for expiration 473
eligible for space management 490
how IBM Tivoli Storage Manager stores 258
on a volume, querying 384
server migration of 267
client migration 490
client node
adding 415
agent 436
amount of space used 393
creating backup sets for 538
file spaces, QUERY OCCUPANCY command 393
finding tapes used by 387
immediate processing 575
importing 757
locking 434
client node *(continued)*
managing registration 416, 425, 579
options file 421
performing operations for 529, 561, 568
privilege class for scheduling operations for 555
proxy node relationships 435
querying 438
reducing archive packages for 550
registering 420
removing 434
renaming 433
scheduling operations for 554
setting password authentication 855
setting scheduling mode 570
setting up subfile backups 547
target 436
unlocking 434
updating 428
viewing information about 438
client nodes
defining 445
file spaces 445
managing 23, 425
managing across a firewall 426
prevent from accessing server 734
client option
TXNBYTELIMIT 259
VIRTUALMOUNTPOINT 445
client option set
adding client options to 459
assigning clients to 460
copying 461
creating 459
deleting 461
deleting an option from 461
for NAS node 209
requesting information about 461
updating description for 461
client options
UNIX and Linux 422
client options file
create 422
update 422
client point-in-time restore, enabling 522
client queries to the server, setting the frequency 573
client restartable restore session
canceling 466
interrupting, active 467
requesting information about 466
client schedule
define 555
client session
canceling 463
DSMC loop 461
held volume 461
managing 461
querying 462, 780
viewing information about 462, 780
client system options file 420
client-polling scheduling 569, 573
client, application
adding node for 416
description 4
policy for 517
client/server, description of 3
clients
adding through the command line client 424

closed registration
description 417
process 417
setting 417
collocation
active-data pools 360
changing, effect of 359
copy storage pools 360
definition 241, 353, 406
description of 21
determining whether to use collocation 241, 353, 406
effects on operations 355
effects on volume reclamation 374
enabling 361
enabling for sequential storage pool 241, 353, 406
how the server selects volumes when disabled 359
migration thresholds 275
planning 361
selecting volumes when enabled 357
command file, scheduling on client 558
command retry attempts
setting the amount of time between 574
setting the number of 574
command routing 716
command script 617
commands
grant authority 838
commands, administrative
ACCEPT DATE 600
ASSIGN DEFMGMTCLASS 504
AUDIT LIBVOLUME 147
AUDIT LICENSE 581
AUDIT VOLUME 882
BACKUP NODE 224, 226
BEGIN EVENTLOGGING 799
CANCEL PROCESS 601
CANCEL RESTORE 466
CANCEL SESSION 463
CHECKIN LIBVOLUME 133
CHECKOUT LIBVOLUME 145
CLEAN DRIVE 160
COMMIT 630
COPY ACTIVEDATA 239, 261
COPY CLOPTSET 461
COPY DOMAIN 494
COPY POLICYSET 494
COPY SCHEDULE 562
COPY SCRIPT 624
DEFINE ASSOCIATION 556
DEFINE BACKUPSET 542
DEFINE CLIENTACTION 575
DEFINE CLIENTOPT 575
DEFINE CLOPTSET 458
DEFINE COPYGROUP 496, 501, 502
DEFINE DATAMOVER 171, 220
DEFINE DEVCLASS
3592 180
FILE device classes 184
LTO device classes 188
REMOVEABLEFILE device classes 183
SERVER device classes 191
tape device classes 175
VOLSAFE device classes 192
DEFINE DRIVE 170
DEFINE GRPMEMBER 719
DEFINE LIBRARY 42, 169
DEFINE MACHINE 986
commands, administrative *(continued)*
DEFINE MACHNODEASSOCIATION 986
DEFINE PATH 172
DEFINE POLICYSET 494, 495
DEFINE PROFASSOCIATION 697, 698
DEFINE PROFILE 697
DEFINE RECMEDMACHASSOCIATION 989
DEFINE RECOVERYMEDIA 988
DEFINE SCHEDULE 611
DEFINE SCRIPT 617
DEFINE SERVER 684, 716, 722
DEFINE SERVERGROUP 719
DEFINE STGPOOL 245, 247, 257, 258
DEFINE SUBSCRIPTION 709
DEFINE VIRTUALFSMAPPING 232
DEFINE VOLUME 50, 252
DELETE ASSOCIATION 565
DELETE BACKUPSET 546
DELETE COPYGROUP 526
DELETE DOMAIN 527
DELETE DRIVE 165
DELETE EVENT 567
DELETE GRPMEMBER 721
DELETE LIBRARY 153
DELETE MGMTCLASS 526
DELETE POLICYSET 526
DELETE PROFASSOCIATION 702
DELETE PROFILE 704
DELETE SCHEDULE 562
DELETE SCRIPT 626
DELETE SERVER 690
DELETE SERVERGROUP 721
DELETE STGPOOL 409
DELETE SUBSCRIBER 714
DELETE SUBSCRIPTION 704, 710
DELETE VOLHISTORY 606
DELETE VOLUME 411, 412
DISABLE EVENTS 798
DISABLE SESSIONS 465
DSMSERV DISPLAY DBSPACE 661
DSMSERV DISPLAY LOG 661
ENABLE EVENTS 798
ENABLE SESSIONS 465
END EVENTLOGGING 799
EXPIRE INVENTORY 31
EXPORT ADMIN 731
EXPORT NODE 745
EXPORT POLICY 745
EXPORT SERVER 745
EXTEND DBSPACE 662, 666
GENERATE BACKUPSET 538
GRANT AUTHORITY 835
HALT 598
HELP 607
IMPORT 760, 761
IMPORT ADMIN 748
IMPORT NODE 748, 757
IMPORT POLICY 748
IMPORT SERVER 748, 757
LABEL LIBVOLUME 125
LOCK ADMIN 842
LOCK NODE 434
LOCK PROFILE 701, 702
MOVE DATA 397
MOVE NODEDATA 402
NOTIFY SUBSCRIBERS 702
PING SERVER 722

commands, administrative *(continued)*
PREPARE 989
QUERY ACTLOG 790
QUERY BACKUPSETCONTENTS 545
QUERY CONTENT 384
QUERY COPYGROUP 523, 755
QUERY DB 661
QUERY DBSPACE 661
QUERY DEVCLASS 195
QUERY DOMAIN 525
QUERY DRIVE 153
QUERY DRMSTATUS 980
QUERY ENABLED 812
QUERY EVENT 557
QUERY FILESPACE 456
QUERY LIBRARY 151
QUERY LICENSE 581
QUERY MGMTCLASS 524
QUERY NODE 438
QUERY NODEDATA 394
QUERY OCCUPANCY
backed-up, archived, and space-managed files 395
client file spaces 393
client nodes 393
device classes 394
storage pools 394
QUERY OPTION 782
QUERY POLICYSET 525
QUERY PROCESS 400
QUERY RESTORE 466
QUERY RPFCONTENT 991
QUERY RPFILE 991
QUERY SCHEDULE 557
QUERY SCRIPT 625
QUERY SERVERGROUP 720
QUERY STGPOOL 378, 388, 751
QUERY SUBSCRIPTION 709
QUERY SYSTEM 783
QUERY VOLUME 380, 401
RECONCILE VOLUMES 729
REGISTER ADMIN 837
REGISTER LICENSE 580
REMOVE ADMIN 841
REMOVE NODE 434
RENAME ADMIN 840
RENAME FILESPACE 759
RENAME NODE 433
RENAME SCRIPT 626
RENAME SERVERGROUP 720
RENAME STGPOOL 405
RESTORE DB 599
RESTORE NODE 224, 226
RESTORE STGPOOL 904
ROLLBACK 631
RUN 626
SELECT 784
SET ACCOUNTING 793
SET AUTHENTICATION 855
SET CLIENTACTDURATION 575
SET CONFIGMANAGER 693, 696
SET CONFIGREFRESH 709
SET CONTEXTMESSAGING 798
SET CROSSDEFINE 685, 688
SET DBREPORTMODE 661
SET DRMCHECKLABEL 983
SET DRMCOPYSTGPOOL 980
SET DRMCOURIERNAME 983
commands, administrative *(continued)*
SET DRMDBBACKUPRXPIREDAYS 983
SET DRMFILEPROCESS 983
SET DRMINSTPREFIX 980
SET DRMNOTMOUNTABLE 983
SET DRMPLANPOSTFIX 980
SET DRMPLANPREFIX 980
SET DRMPRIMSTGPOOL 980
SET DRMRPFEXPIREDAYS 991
SET DRMVAULTNAME 983
SET EVENTRETENTION 567, 616
SET INVALIDPWLIMIT 853
SET LICENSEAUDITPERIOD 581
SET MAXCMDRETRIES 574
SET MAXSCHEDSESSIONS 571
SET MINPWLENGTH 854
SET PASSEXP 850
SET QUERYSCHEDPERIOD 573
SET RANDOMIZE 572
SET REGISTRATION 417
SET RETRYPERIOD 574
SET SCHEDMODES 569
SET SERVERHLADDRESS 685, 688
SET SERVERLLADDRESS 685, 688
SET SERVERNAME 684, 688, 782
SET SERVERPASSWORD 684, 685, 688
SET SUBFILE 546
SET SUMMARYRETENTION 788
SETOPT 605
UNLOCK PROFILE 701, 702
UPDATE ADMIN 837
UPDATE ARCHIVE 550
UPDATE BACKUPSET 544
UPDATE CLIENTOPT 461
UPDATE CLOPTSET 461
UPDATE COPYGROUP 496, 502
UPDATE DEVCLASS 175
UPDATE DOMAIN 494
UPDATE DRIVE 154
UPDATE LIBRARY 152
UPDATE LIBVOLUME 50, 145
UPDATE MGMTCLASS 495
UPDATE NODE 428
UPDATE POLICYSET 494
UPDATE RECOVERYMEDIA 989
UPDATE SCHEDULE 611
UPDATE SCRIPT 623, 624
UPDATE SERVER 690
UPDATE SERVERGROUP 720
UPDATE VOLUME 252
VALIDATE LANFREE 126
commands, administrative COPY SERVERGROUP command
COPY SERVERGROUP 720
COMMIT command 630
Common Agent Services
CAS 582
Common Inventory Technology 582
COMMTIMEOUT server option 462, 464
communication set up
among servers 682
command routing, for 686
cross definition 683, 684, 688
enterprise configuration, for 682
enterprise event logging, for 682, 811
security 685
server-to-server virtual volumes 724
using Secure Sockets Layer 823

communication set up *(continued)*
using Transport Layer Security (TLS) 823
compression
choosing client or drive 197
option for API 420
options for clients 418
setting 418
tape volume capacity, effect on 197
concurrent access
Centera volumes 194
configuration files, editing 92
configuration information, enterprise management
administrative command schedule 694, 701
administrator 697, 713
changing 701
client option set 694, 698
client schedule 694, 695, 699
copy group 694, 699
deleting 702, 704
distributing 691, 698, 702
management class 699
policy domain 694, 695, 699
refreshing 702, 709, 711
script 694, 698
server 700
server group 700
configuration manager
communication setup 682
default profile 693, 700
scenario 693
setting up 693, 696, 697
configure 85
configuring
connect 349x library to server 217
connecting ACSLS library to server 218
devices, manual library example 123
NDMP operations for NAS file servers 207
planning your storage environment 65
shared library 110
Solaris Zones 95
VTL 116
configuring libraries
3494 102
ACSLS 102
SCSI 102
VTL 102
configuring storage 101
console mode 761
contents of a volume 384
context messaging for ANR9999D 798
continuation characters, using 621
conventions
typographic xv
COPY CLOPTSET command 461
COPY DOMAIN command 494
copy group
archive, description of 476
backup, description of 476
defining archive 502
defining backup 496
deleting 526
COPY MGMTCLASS command 495
COPY POLICYSET command 494
COPY SCHEDULE command 562, 614
COPY SCRIPT command 624
copy storage pool
compared with primary 408
copy storage pool *(continued)*
defining a 406
restore from multiple 901
role in storage pool migration 279
simultaneous-write function 327
creating
backup sets 24
new policy 26
server scripts 617
creating backup sets
benefits of 538
example for 540
cross definition 683, 684, 688
customer support
contacting 1072
cyclic redundancy check
during a client session 529
for storage pool volumes 878
for virtual volumes 722
performance considerations for nodes 530
performance considerations for storage pools 881

D

daily monitoring disk storage pools 772
daily monitoring of databases 769
daily monitoring of server processes 768
daily monitoring scheduled operations 776
daily monitoring sequential access storage pools 773
damaged files 884, 885
data
active backup versions, storing 237
considering user needs for recovering 64
exporting 731
importing 731
data compression 418
data deduplication xx, 313, 314
checklist for configuration 289
client-side 310
changing location 312
client and server settings 281, 306
multiple nodes 311
overview 282
single node 311
controlling duplicate-identification manually 308
data deduplication 291, 292, 293, 294, 295, 296, 297, 320, 321, 322, 323, 324, 325, 326
DEDUPLICATION parameter 306
DEDUPREQUIRESBACKUP server option 303
definition 281
detecting security attacks 298
duplicate-identification processes 302, 307, 309, 310
IDENTIFY DUPLICATES command 308
limitations 284
managing 302
memory xx
moving or copying data 304
node replication 919, 975
options for 314
planning 287
processing 301
protecting data 303
reclamation 303
requirements 289
server-side 281, 282, 306
specifying the size of objects to be deduplicated 312

data deduplication *(continued)*
statistics
displaying information about files with links to a volume 315
querying a duplicate-identification process 315, 316, 317, 319
querying a storage pool 314
testing
restore operations 299
space savings 300
Tivoli Storage Manager for Virtual Environments 322
turning off 305
virtual volumes, server-to-server
data deduplication 304
data format
NATIVE 234
data format for storage pool 202, 205, 234
definition 241
operation restrictions 245
data movement, querying 400
data mover
defining 171, 220
description 49, 50
managing 204
NAS file server 49, 50
data protection with WORM media 138
data retention protection 508
data retention using Centera
overview 46
unsupported functions 245
data shredding
BACKUP STGPOOL command 535
COPY ACTIVEDATA command 535
DEFINE STGPOOL command 535
DELETE FILESPACE, command 535
DELETE VOLUME, command 535
description 533
enforcing 535
EXPIRE INVENTORY command 535
EXPORT NODE command 535, 732
EXPORT SERVER command 535, 732
GENERATE BACKUPSET command 535, 537
MOVE DATA command 398, 535
setting up 534
UPDATE STGPOOL command 535
data storage
active-data pools 237
client files, process for storing 5
concepts overview 15
considering user needs for recovering 64
deleting files from 410
evaluating 65
example 239
managing 18
monitoring 875
planning 65
server options affecting 67
tailoring definitions 755
using another IBM Tivoli Storage Manager server 722
using disk devices 69
using the storage hierarchy 266
data validation
during a client session 529
for storage pool volumes 878
for virtual volumes 722
performance considerations for nodes 530
performance considerations for storage pools 881
database
adding directories 664
audits 634
backup 858, 859, 863, 864
buffer size 634
description of 633
increasing the size 662, 666
log files, alternative locations 672
managing 633
monitoring 661
move to another directory 670
moving 599
protecting 886
relocating on a server 669
reorganization, table and index
archive log, space requirements 658
index 668
overview 666
restrictions 667
table 668
restoring 888, 891, 893
most current state 863, 889, 891
point in time 863, 889
setting user data limit 640
SQL queries 634
statistics collection 634
transactions 633, 675
database backup 860
database backup and recovery
general strategy 722
methods 722
providing 722
database manager 644, 645
database space 644
database, IBM Tivoli Storage Manager
description of 32
ensuring integrity of 34
querying using SQL 784
date and time, changing on the server 600
day of the week parameter 612
DB2 tools, use of 633
DB2 Universal Database
Enterprise Extended Edition 435
db2profile 590
deactivating policy 472
deduplicate-identification processes 307
DEDUPTIER2FILESIZE 313, 314
DEDUPTIER3FILESIZE 313, 314
default management class 26
assigning for a policy set 504
binding files to 485
description of 476
purpose 481
recommendation for using 483
default policy 471
default profile 693, 700, 707
default scheduling mode
modifying 568
DEFINE ASSOCIATION command 556
DEFINE BACKUPSET command 542
DEFINE CLIENTACTION command 575
DEFINE CLIENTOPT command 459
DEFINE CLOPTSET command 458
DEFINE COPYGROUP command 496, 501, 502, 504
DEFINE DEVCLASS command 175, 183
define drive 116
DEFINE DRIVE command 170

DEFINE GRPMEMBER command 719
DEFINE LIBRARY command 169
DEFINE MACHNODEASSOCIATION command 986
DEFINE MGMTCLASS command 495
define path 116
DEFINE POLICYSET command 494
DEFINE PROFASSOCIATION command 698
DEFINE PROXYNODE command 436
DEFINE RECMEDMACHASSOCIATION command 989
DEFINE RECOVERYMEDIA command 988
DEFINE SCHEDULE command 31, 611
DEFINE SCRIPT command 617
DEFINE SERVER command 684, 716, 722
DEFINE STGPOOL command 245, 247, 257, 258
DEFINE SUBSCRIPTION command 709
DEFINE VIRTUALFSMAPPING command 232
DEFINE VOLUME command 19, 20, 252
defining
  archive copy group 26
  backup copy group 26
  client nodes 445
  management class 26
  policy 26
  policy set 26
delaying migration for files 274
delaying reuse of volumes 374
DELETE ASSOCIATION command 565
DELETE BACKUPSET command 546
DELETE CLIENTOPT command 461
DELETE COPYGROUP command 526
DELETE DOMAIN command 527
DELETE DRIVE command 165
DELETE EVENT command 567, 616
DELETE FILESPACE command 457
DELETE GRPMEMBER command 721
DELETE KEYRING command 827
DELETE LIBRARY command 153
DELETE MGMTCLASS command 526
DELETE POLICYSET command 526
DELETE PROFASSOCIATION command 702
DELETE PROFILE command 704
DELETE SCHEDULE command 562, 615
DELETE SCRIPT command 626
DELETE SERVER command 691
DELETE SERVERGROUP command 721
DELETE STGPOOL command 409
DELETE SUBSCRIBER command 714
DELETE SUBSCRIPTION command 710
DELETE VOLHISTORY command 606
DELETE VOLUME command 411, 412
deleting
  cached files on disk 397
  empty volume 411, 606
  file spaces 457
  files 410, 506
  scratch volume 250, 606
  storage volume 411
  subfile 548
  volume history information 606
  volume with residual data 412
deletion hold 509
deployment
  command-line interface 428, 429
  importing 430
  schedule 431
  verifying 433
descriptions, for archive packages 549, 550
DESTINATION parameter (storage pool) 471, 496
destroyed volume access mode 255, 899
determining
  cause of ANR9999D messages 798
  the time interval for volume check in 179
device 86
  attaching to server 215
  multiple types in a library 57
  name 84, 85
  supported by IBM Tivoli Storage Manager 40
device class
  3590 174, 175
  3592 175
  4MM 174, 175
  8MM 174, 175
  amount of space used 394
  CENTERA 46
  defining 174
  description of 19
  DISK 174
  DLT 174, 175
  ECARTRIDGE 175
  FILE 174
  FORMAT parameter 177
  GENERICTAPE 174, 175
  LTO 188
  REMOVABLEFILE 183
  requesting information about 195
  selecting for import and export 744
  sequential 175
  SERVER 174, 175, 724
  StorageTek devices 175, 192
  tape 175, 183
  Ultrium, LTO 175
  updating 175, 183
  VOLSAFE 192
  WORM 174, 175
device classes
  database backups 859
device configuration file 867, 894
device driver 91
  configuring 89
  for automated library devices 82
  for IBM 3490, 3570, and 3590 tape drives 87
  for IBM 3494 or 3495 libraries 89
  for manual tape devices 81
  IBM Tivoli Storage Manager, installing 81, 82
  installing 81, 83
  requirements 81, 83
device drivers 83
  installing 87
device sharing 65
device special file names 86
  removing 86
device support 15
device type
  3590 175
  4MM 174, 175
  8MM 174, 175
  CENTERA 46
  DISK 174
  DLT 174, 175
  ECARTRIDGE 175
  FILE 174
  GENERICTAPE 174, 175
  LTO 176, 188
  multiple in a single library 57

device type *(continued)*
REMOVABLEFILE 174
SERVER 174, 175, 724, 726
VOLSAFE 192
WORM 174, 175
device utilities 85
device, storage
disk 69
manual library device 123
optical device 119, 123
removable media device 119, 183
required IBM Tivoli Storage Manager definitions 63
supported devices 40
devices 85
defining 169
diagnosing ANR9999D messages 798
differential backup
compared to incremental 12
of image, description 9, 56
direct-to-tape, policy for 516
directories
deleting from archive packages 550
directory-level backup 231
preventing archive of 551
storage usage for archive packages 549
disability 1075
DISABLE EVENTS command 798
DISABLE SESSIONS command 465
disaster recovery
auditing storage pool volumes 885
general strategy 722
methods 34, 722
node replication as a method for 976
providing 722
server
disaster recovery 1004
server recovery 1004
disaster recovery manager
awk script 1018
client recovery information 979
creating a disaster recovery plan 989
customizing 980
displaying a disaster recovery plan 991
expiring a disaster recovery plan 991
features 979
moving volumes back on-site 998
project plan, checklist 1013
querying a disaster recovery plan 991
recovery media 988
saving machine characteristics 986
stanzas, recovery instructions 985
storing a disaster recovery plan 989
disaster, protecting against 34
disk device class, defined 174
disk devices
random access 71
sequential access 71
disk space
for the database and recovery log 640
freeing 79
disk storage
and file-system requirements 69
random access (DISK) 45
sequential access (FILE) 45
disk storage pool
cache, use of 280
deleting cached files from 397
disk storage pool *(continued)*
estimating space 376
estimating space for archived files 377
estimating space for backed up files 376
migration threshold 269
setting up 69
disk subsystems
requirements 69
disk support 69
disk systems support 69
DISMOUNT VOLUME command 149
display information 438
specific client node 438
DISPLAY OBJNAME command 446
distribute workloads
to reduce backup and restore time 435
DLT WORM media 138
domain, policy
active-data pools, specifying 492
assigning client node 506
associating a client with 26
changing 472
creating 494
deleting 527
description of 476
distributing via profile 522, 695
for NAS file server node 208
querying 525
updating 490, 492
drive
cleaning 160
defining 170
defining path for 172
deleting 165
detecting changes on a SAN 127
element address 170, 172
multiple device types in a library 104
querying 153
serial number 170
server retries for acquiring 67
simultaneous-write function, requirements for 349
updating 154
updating to use for NDMP operations 204
drive configuration 115
DRIVEACQUIRERETRY server option 67
DRIVEENCRYPTION parameter
3592 device class 182
ECARTRIDGE device class 193, 194
LTO device class 190
driver, device
configuring 89
for automated library devices 82
for IBM 3490, 3570, and 3590 tape drives 87
IBM Tivoli Storage Manager, installing 82
installing 81, 83
requirements 81, 83
drives 115
dsm.opt file 421, 458, 553
dsmaccnt.log 793
DSMADMC command 740, 754, 761
DSMC loop session 461
dsmqsan module, role in SAN discovery 127
dsmsched.log file 567
DSMSERV DISPLAY DBSPACE command 661
DSMSERV DISPLAY LOG command 661, 668
DSMSERV_ACCOUNTING_DIR 793
duplicate-identification processes 302, 307

duplication of restored data 901
DVD-RAM support
  defining and updating a device class 183
dynamic serialization, description of 496, 502

E

ECARTRIDGE device class 175
element address 170
ENABLE EVENTS command 798
ENABLE SESSIONS command 465
encoding events to UTF-8 804
encryption
  changing method 532
  choosing a method 531
  DRIVEENCRYPTION parameter
    3592 Generation 2 182
    ECARTRIDGE 193, 194
    LTO-4 190
  methods 155, 530
END EVENTLOGGING command 799
Enterprise Administration
  description 677
enterprise configuration
  communication setup 682
  description 678, 691
  procedure for setup 692
  profile for 693
  scenario 680, 692
  subscription to 695
enterprise event logging 682, 811
environment variable, accounting 793
error analysis 783
error checking for drive cleaning 164
error reporting for ANR9999D messages 798
error reports for volumes 382
establishing server-to-server communications
  enterprise configuration 682
  enterprise event logging 682
  virtual volumes 690
estimate network bandwidth 929
estimate replication 929
estimated capacity for storage pools 378
estimated capacity for tape volumes 382
event logging 797, 804
event record (for a schedule)
  deleting 567, 616
  description of 557, 565
  managing 615
  querying 615
  removing from the database 567, 616
  setting retention period 567, 616
event server 811
example
  assigning a default management class 504
  register three client nodes with CLI 424
  validating and activating a policy set 506
expiration 79
expiration date, setting 613
expiration processing 31
  description 875
  files eligible 473, 506
  of subfiles 473, 498, 506, 548
  starting 506
  using disaster recovery manager 508
EXPIRE INVENTORY command 31
  duration of process 507
export
  administrator information 741
  client node information 741
  data from virtual volumes 763
  decided when 733
  directly to another server 734
  labeling tapes 736, 744
  monitoring 760
  options to consider 735
  planning for sequential media 744
  policy information 742
  PREVIEW parameter 743
  previewing results 739
  querying about a process 760
  querying the activity log 762
  replacing definitions before 736
  server data 742
  using scratch media 744
  viewing information about a process 760
EXPORT ADMIN command 745
export and import data
  sequential media volumes 743
EXPORT commands 760, 761
EXPORT NODE command 746
EXPORT POLICY command 747
EXPORT SERVER command 743, 747
exporting
  administrator data 745
  client node data 746
  data to tape 745
  description of 731
  policy data 747
  server data 747
  subfiles 548
EXPQUIET server option 507
EXTEND DBSPACE command 662, 666
EXTERNAL library type 1056
external media management
  IBM Tivoli Storage Manager setup 121
  initialization requests 1056
  interface description 1051
  overview 121
  processing during server initialization 1052
  using with IBM Tivoli Storage Manager
    media-managed storage pools, deleting 122
  volume dismount requests 1061
  volume mount requests 1058
  volume release requests 1057

F

failover 956
file data, importing 731
file deletion option
  setting 421
FILE device type
  active-data pools 237
  backing up or archiving many small objects 185
  benefits 45
  concurrent access to FILE volumes 45
  defining device class 174
  deleting scratch volumes 606
  free space in directories 396
  setting up storage pool 77
file exit 797
  logging events to 800
file name for a device 84

file path name 446
file retrieval date 281
file server, network-attached storage (NAS)
  backup methods 209
  registering a NAS node for 219
  using NDMP operations 55, 199
file size, determining maximum for storage pool 241
file space
  deleting, effect on reclamation 364
  deleting, overview 457
  description of 445
  merging on import 735, 749
  names that do not display correctly 456
  QUERY OCCUPANCY command 393
  querying 445
  renaming 759
  Unicode enabled 456
  viewing information about 445
file space identifier (FSID) 456
file spaces
  defining 445
FILE volumes
  shared 173
file-level restore
  managing 230
  planning 229
file-system requirements for disk storage 69
file, client
  allowing archive while changing 471
  allowing backup while changing 471, 496
  archive package 549
  associating with management class 483, 484
  damaged 901
  delaying migration of 274
  deleting 410
  deleting from a storage pool 409
  deleting from cache 280
  deleting when deleting a volume 410
  duplication when restoring 901
  eligible for archive 471, 486
  eligible for backup 471, 486
  eligible for expiration 473
  eligible for space management 490
  how IBM Tivoli Storage Manager stores 258
  on a volume, querying 384
  server migration of 267
files, damaged 884, 885, 901
files, unreadable 884, 885
firewall, client nodes
  client-initiated sessions 426
  server-initiated sessions 427
fixes, obtaining 1072
format for storage pool 202, 205
  definition 241
  operation restrictions 245
formatting
  event classes 802
  storage pool volume 76, 251
frequency of backup 497
FSID 456
full image backup, NAS node 56
full library 146

G

GENERATE BACKUPSET command 538
GENERICTAPE device type 183
Global Security Kit (GSKit) 823
glossary 1081
GRANT AUTHORITY command 835
group backup, on the client 11
group, server
  copying 720
  defining 719
  deleting 721
  member, deleting 721
  moving a member 721
  querying 720
  renaming 720
  updating description 720

H

HALT command 598
halting the server 598
hardware scans
  VMware environment 444, 587
held volume in a client session 461
HELP command 607
hierarchy, storage 21
  copying active backup data 237
  defining in reverse order 245, 257
  establishing 256
  example 239
  for LAN-free data movement 257
  how the server stores files in 258
  next storage pool
    definition 257
    deleting 410
    migration to 267, 388
  restrictions 257
  staging data on disk for tape storage 266
HL ADDRESS 427
hostname
  changing 604
how to cause the server to accept date and time 600

I

IBM device driver
  installing 87, 89
IBM device drivers 83
IBM error analysis 783
IBM Support Assistant 1071
IBM Tivoli Monitoring 795
IBM Tivoli Storage Manager (Tivoli Storage Manager)
  introduction 3
  server network 32
IDLETIMEOUT server option 462, 464
image backup
  policy for 517, 518
  suggested use 8, 11
import
  data from virtual volumes 763
  monitoring 760
  options to consider 749
  PREVIEW parameter 743, 751
  querying about a process 760
  querying the activity log 762
  recovering from an error 759
  replace existing definitions 750
  viewing information about a process 760
IMPORT ADMIN command 748

IMPORT commands 760, 761
IMPORT NODE command 748, 757
IMPORT POLICY command 748
IMPORT SERVER command 748, 757
importing
  active-data pools 756
  data 748
  data storage definitions 753, 755
  date of creation 750, 757
  description of 731
  directing messages to an output file 740, 754
  duplicate file spaces 756
  file data 756
  node replication restriction 921
  policy definitions 753
  server control data 755
  subfiles 548
  subsets of information 758
include-exclude file 29
  creating 29
  description of 26, 482
  for policy environment 476, 482
incomplete copy storage pool, using to restore 901
incremental backup 486
incremental backup, client
  file eligibility for 486
  frequency, specifying 571
  full 486
  partial 487
  progressive 12
incremental replication 933
inheritance model for the simultaneous-write function 336
initial replication 932
initial start date for schedule 612
initial start time for schedule 612
initializing
  tape volumes 19, 20
installing IBM Tivoli Storage Manager 416
instance user ID 597
instant archive
  creating on the server 537
  description of 8, 12
interface, application program
  client, registering 420
  compression option 420
  deletion option 420
  description of 3
  registering to server 420
  simultaneous-write function, version support for 329
interfaces to IBM Tivoli Storage Manager 17
Internet, searching for problem resolution 1071, 1072
introduction to IBM Tivoli Storage Manager 3

K

keepalive, TCP
  enabling 207
  overview 206
  specifying connection idle time 207
key database
  adding certificates 826, 827
  password change 826, 827
keyboard 1075
knowledge bases, searching 1071

L

label
  automatic labeling in SCSI libraries 132
  checking media 137
  overwriting existing labels 130, 132
  sequential storage pools 129, 252
  volume examples 131
  volumes using a library device 131
LABEL LIBVOLUME command
  identifying drives 130
  insert category 133
  labeling sequential storage pool volumes 130
  manually mounted devices 123
  overwriting existing volume labels 130
  removable media volumes 130
  restrictions for VolSafe-enabled drives 192
  using a library device 131
  using a manual library 125
  volume labeling examples 131
LAN-free data movement 125
  description 13, 54
  storage pool hierarchy restriction 257
  suggested usage 9
LDAP-authenticated password
  configuring an LDAP directory server 845
  configuring the server 847
  policy 846
  query admin 850
  query node 850
  register nodes and admin IDs 848
  scenarios 851
  transport layer security 829
  update node or admin 849
libraries
  NDMP operations 213
  virtual tape library 114
library
  ACSLS (Automated Cartridge System Library Software) 44
  adding volumes 133
  attaching for NAS file server backup 215
  auditing volume inventory 147
  automated 144
  categories for volumes in IBM 3494 42
  configuration 102
  configuration example 123
  configure for more than one device type 57
  defining 153, 169
  defining path for 172
  deleting 153
  detecting changes to, on a SAN 127, 170
  external 44
  full 146
  IBM 3494 42
  managing 151
  manual 41, 123
  mixing device types 57, 180, 188
  mode, random or sequential 83
  overflow location 241
  querying 151
  SCSI 42
  serial number 170
  sharing among servers 110
  type 52
  updating 152
  volume inventory 51
  VTL 42

library client, shared library 53, 112, 113, 893, 904
library drive usage
determining 212
library manager, shared library 53, 891, 903
library sharing 59
license
compliance 581
features 580
monitoring 581
registering 580
using 579
limitation for script definition on administrative Web interface 617
LL ADDRESS 427
location, volume
changing 254
overflow for storage pool 241
querying volume 383
LOCK ADMIN command 842
LOCK NODE command 434
LOCK PROFILE command 701, 702
log mirror 637, 668
logical block protection
enabling 157
overview 156
read/write operations 159
storage pool management 160
supported drives 157
logical devices 77
logical volume on client
backup 474
management class for 483
policy for 486, 517
process for backup 489
restore 475
logical volume, raw 22, 76, 249, 251
loop session, DSMC 461
low migration thresholds 273
LTO Ultrium devices and media
device class, defining and updating 188
encryption 155, 190, 530
WORM 138, 192
LUN
using in paths 172

M

machine characteristics 987
machine recovery information 987
macro
commit individual commands 630
continuation characters 628
controlling command processing 630
running 630
scheduling on client 558
substitution variables 629
testing 631
using 627
writing commands 627
writing comments 628
MACRO administrative command, using 424
magnetic disk devices 45, 69
maintenance plan
modify 619
maintenance script
create 618
custom 618
maintenance script *(continued)*
modify 31
managed server
changing the configuration manager 708, 714
communication setup 682
deleting a subscription 710
description 678
managed objects 678, 707
refreshing configuration information 711
renaming 715
returning managed objects to local control 712
setting up 695
subscribing to a profile 695, 707, 708, 709
management class
assigning a default 504
associating a file with 483
binding a file to 483
configuration 480
controlling user access 480
copying 490, 495
default 29, 481
define new 521
defining 495
deleting 526
description of 476, 480
querying 524
rebinding a file 485
updating 484, 490, 495
managingserver operation 30
manual library
defining devices 123
manual library device 123
MAXSCRATCH parameter 241, 253, 406
media
tape rotation 60, 141
media label
checking 137
for tape 130
recording 130
merging file spaces 735, 749
messages
determining cause of ANR9999D message 798
directing import messages to an output file 740, 754
for automated libraries 149
for drive cleaning 164
getting help on 607
severe 798
MIGRATE STGPOOL command 278
migrating a file 475, 490
migration, client
automatic, for HSM client
demand 475
files, eligible 490
threshold 475
premigration for HSM client 475
reconciliation 476
selective, for HSM client 475
stub file on HSM client 475
migration, server
canceling the server process 389
controlling by file age 274
controlling duration 278
controlling start of, server 272
copy storage pool, role of 279
defining threshold for disk storage pool 272
defining threshold for tape storage pool 275
delaying by file age 274

migration, server *(continued)*
description, server process 269
minimizing access time to migrated files 275
monitoring thresholds for storage pools 388
multiple concurrent processes
random access storage pool 241, 269
sequential access storage pool 241, 278
problems, diagnosing and fixing 267
providing additional space for server process 390
starting manually 278
starting server process 266, 272
threshold for a storage pool
random access 269
sequential access 275, 276
mirroring 865
description of 34
MIRRORLOGDIRECTORY server option 674
mixed device types in a library 57, 180, 188
mobile client support 546
mode
client backup 497
library (random or sequential) 83
scheduling 569
modified mode, description of 497
modifying schedules 562
monitoring
server-to-server export 740
mount
count of number of times per volume 383
library 177
limit 178
operations 149
query 149
retention period 179
wait period 179
mount point 966
preemption 602
queue, server option 67
relationship to mount limit in a device class 178, 187, 195
requirements for simultaneous-write operations 348
settings for a client session 417
MOVE DATA command 397
MOVE DRMEDIA command 998
MOVE NODEDATA 402
moving a backup set
benefits of 542
to another server 542
moving data 457
from offsite volume in a copy storage pool 398
monitoring the movement of 401
procedure 399
requesting processing information 400
to another storage pool 398
to other volumes in same storage pool 397
multipath I/O 88
multiple
copy storage pools, restoring from 901
managing IBM Tivoli Storage Manager servers 32
managing Tivoli Storage Manager servers 677
multiple drive device types 109
multiple server instances 597
multiple servers 715
completing tasks 715
multistreaming, concurrent for database backups and restores 860, 888

N

name of device 84
NAS file server, NDMP operations
backing up a NAS file server 226
backing up a NAS file server to native pools 227, 228
configuration checklist 207
data format 202
data mover, description 49, 50, 171
defining a data mover 171, 220
defining a storage pool 213
defining paths to drives
drives attached only to file server 221
drives attached to file server and Tivoli Storage Manager server 220
obtaining names for devices attached to file server 222
defining paths to libraries 223
differential image backup, description 56
full image backup, description 56
interfaces used with 201
managing NAS nodes 203
path, description 49, 50, 172
planning 211
policy configuration 208, 518
registering a NAS node 219, 419
requirements for set up 199
restoring a NAS file server 226
scheduling a backup 224
storage pools for NDMP operations 213
NAS node
defining 219
deleting 204
registering 219
renaming 203
NAS nodes 234
NATIVE data format 202
NDMP
operations 202
NDMP backup data 234
NDMP operations 234
NDMP operations for Celerra data movers 234
NDMP operations for NAS file servers
backing up a NAS file server 226
backing up a NAS file server to native pools 227, 228
configuration checklist 207
data format 202
data mover, description 49, 50, 171
defining a data mover 171, 220
defining a storage pool 213
defining paths to drives
drives attached only to file server 221
drives attached to file server and Tivoli Storage Manager server 220
obtaining names for devices attached to file server 222
defining paths to libraries 223
differential image backup, description 56
full image backup, description 56
interfaces used with 201
managing NAS nodes 203
path, description 49, 50, 172
planning 211
policy configuration 208, 518
prevent closing of inactive connections
enabling TCP keepalive 207
overview 206
specifying connection idle time 207
registering a NAS node 219, 419
requirements for set up 199

NDMP operations for NAS file servers *(continued)*
restoring a NAS file server 226
scheduling a backup 224
storage pools for NDMP operations 213
NetApp file server
data format for backup 202
international characters 230
NETAPPDUMP data format 202, 213
NetView 797
Network Appliance file server
backup methods 209
requirements 199
storage pool for backup 245
tape device for backup 199
using NDMP operations 55, 199
network attached storage
virtual file spaces 224
network bandwidth 929
network of IBM Tivoli Storage Manager servers 32
network of Tivoli Storage Manager servers 677
network-attached nodes
comparing to local nodes 423
network-attached storage (NAS) file server
backup methods 209
registering a NAS node for 219
using NDMP operations 55, 199
next storage pool
definition 257
deleting 410
migration to 267, 388
node
registering 442, 472
node privilege class
description of 440
granting 441
node replication 933, 966
configuration
effectiveness, measuring 974
removing 977
setting up the default 936, 938
source and target replication servers 909
validating 958
database requirements 928
database restore, replicating after 971
deduplicated data 919, 975
disabling and enabling
rules 970
disabling and enabling replication
all client nodes 968, 969
data types in file spaces 966
individual client nodes 967
disaster recovery
recovering data from the target 976
store operations on the target replication server 977
file spaces
data types, disabling and enabling 966
purging data in 971
querying replication results 973
rules, changing 940
import-and-export operations
converting from 949
restriction 921
migration by HSM for Windows client 921
mode, replication 918
nodes
adding for replication 950
node replication *(continued)*
nodes *(continued)*
all client nodes, disabling and enabling replication 968, 969
attributes updated during replication 920
individual client nodes, disabling and enabling replication 967
removing from replication 951
resetting the replication mode 952
overview 907
planning 927
policy management 910
previewing results 958
process information
activity log 974
file spaces 973
record retention 976
summary records 974
records
displaying 973
retaining 976
replicating
canceling processes 973
data by file space 960
data by priority 962
data by type 961
scheduling or starting manually 959
throughput, managing 964
restoring, retrieving, and recalling data from the target 976
results, previewing 958
retention protection, archive 921
rules
attributes 912
definitions 911
disabling and enabling 970
file spaces 940
hierarchy 912
nodes, individual 942
processing example 913
server 944
Secure Sockets Layer (SSL) 956, 957
servers
communications, setting up 936
configurations 909
source, adding 953
target 953, 977
settings, displaying
file spaces 973
nodes 973
rules 973
SSL (Secure Sockets Layer) 956, 957
state, replication 915
task tips
monitoring processes 926
nodes, adding and removing 924
previewing results 925
processing, managing 925
rules, changing replication 923
servers, managing 924
validating a configuration 925
verifying results 926
node replication method 931, 932
node replication synchronization 931
node replication tiering 1042
node, client
adding 415

node, client *(continued)*
agent 436
amount of space used 393
creating backup sets for 538
file spaces, QUERY OCCUPANCY command 393
finding tapes used by 387
immediate processing 575
importing 757
locking 434
managing registration 416, 425, 579
options file 421
performing operations for 529, 561, 568
privilege class for scheduling operations for 555
proxy node relationships 435
querying 438
reducing archive packages for 550
registering 420
removing 434
renaming 433
scheduling operations for 554
setting password authentication 855
setting scheduling mode 570
setting up subfile backups 547
target 436
unlocking 434
updating 428
viewing information about 438
nodes
moving nodes from schedule 564
overview of client and server 415
non-global zone
displaying devices 98
non-global zones
removing devices 99
NOPREEMPT server option 602
NORETRIEVEDATE server option 281
NOTIFY SUBSCRIBERS command 702
number of times mounted, definition 383

O

occupancy, querying 393
off-site volume
limiting the number to be reclaimed 373
offsite recovery media
specify defaults 983
offsite recovery media (for DRM)
volumes
moving back on-site 998
sending offsite 996
states 994
offsite volume access mode 256
offsite volumes
limiting the number to be reclaimed 241
moving data in a copy storage pool 398
one-drive library, volume reclamation 241, 368
open registration
description 417
enabling 424
process 418
setting 417
operations available to client 10
operator privilege class
reducing 840
revoking 840
optical device
reclamation for media 369
option set, client
adding client options to 459
assigning clients to 460
copying 461
creating 459
deleting 461
deleting an option from 461
for NAS node 209
requesting information about 461
updating description for 461
option, server
3494SHARED 67
ACSLS options 67
ASSISTVCRRECOVERY 67
AUDITSTORAGEstorage audit 581
changing with SETOPT command 605
CHECKTAPEPOS 67
COMMTIMEOUTcommunication timeout 462, 464
DRIVEACQUIRERETRY 67
EXPINTERVAL 507
EXPQUIET 507
IDLETIMEOUTidle timeout 462, 464, 780
NOPREEMPT 67, 602
NORETRIEVEDATEfile retrieval date 281
overview 30
QUERYAUTH 835
REQSYSAUTHOUTFILE 835
RESOURCETIMEOUT 67
RESTOREINTERVALrestore interval 465, 473, 507
SEARCHMPQUEUE 67
THROUGHPUTDATATHRESHOLD 464
THROUGHPUTTIMETHRESHOLD 464
TXNGROUPMAXmaximum transaction group size 259
options file, client 421
options, querying
VIRTUALMOUNTPOINT client option 446
overflow location 241
owner authority, client 440, 442

P

PARALLEL command 620
Passport Advantage 1073
passthru driver 83
password
changing the key database 826, 827
LDAP-authenticated policy 846
resetting an administrative 840
setting authentication for a client 855
setting expiration 850
setting invalid limit 853
setting minimum length 854
path
defining 172
description 49, 50, 212
paths
defining 169
pending, volume state 383
per product ID (PID) 585
PERFORM LIBACTION 116
performance
cache, considerations for using 78, 280
clients, optimizing restore 237
concurrent client/server operation considerations 571
data protection, increasing with simultaneous-write function 327
data validation for nodes 530

performance *(continued)*
data validation for storage pools 881
file system effects on 76
random-access disk (DISK) 69, 251
FILE-type volumes, backing up many small objects to 185
fragmentation, private FILE volumes for reducing disk 45
migration, multiple concurrent processes 241, 278
mobile client 546
reclamation, multiple concurrent processes
copy storage pools 371
primary sequential access storage pools 241, 367
storage pool backup, reducing time required for 327
storage pool volume 275
volume frequently used, improve with longer mount retention 179
period, specifying for an incremental backup 571
plan
Disaster Recovery Manager 1042
DRM 1042
planning, capacity
database space requirements
estimates based on number of files 642
estimates based storage pool capacity 644
starting size 642
recovery log space requirements
active and archive logs 646
active log mirror 658
archive failover log 659
archive log space for database reorganization 658
point-in-time restore
enable for clients 8, 522
policy
default 5, 471
deleting 525
description of 476
distributing with enterprise management 522
effect of changing 504, 505
for application clients 517
for clients using SAN devices 519
for direct-to-tape backup 516
for logical volume backups 517
for NAS file server node 208
for point-in-time restore 522
for server as client 521
for space management 471, 490, 495
importing 753
managing 469
operations controlled by 474
planning 470
querying 523
policy domain
active-data pools, specifying 492
assigning client node 506
associating a client with 26
changing 472
creating 494
define 520
deleting 527
description of 476
distributing via profile 522, 695
for NAS file server node 208
querying 525
updating 490, 492
policy privilege class
description 834, 835
granting 839
revoking 839, 840
policy set
activating 505
changing, via the active policy set 472
copying 472, 490, 494
defining 494
deleting 526
description of 476
querying 525
updating 494
validating 504, 506
pool, storage
3592, special considerations for 180
active-data pool 237
amount of space used 394
auditing a volume 875
comparing primary and copy types 408
copy 237
creating a hierarchy 256
data format 202, 241, 245
defining 241
defining a copy storage pool 406
defining for disk, example 245, 257
defining for NDMP operations 213
defining for tape, example 245, 257
deleting 409
description of 236
destination in copy group 496, 502
determining access mode 241, 406
determining maximum file size 241
determining whether to use collocation 241, 353, 406
disk 22
duplicate, using to restore 901
enabling cache for disk 241, 280
estimating space for archived files on disk 377
estimating space for backed up files on disk 376
estimating space for disk 376
estimating space for sequential 377
estimating space in multiple 256
incomplete, using to restore 901
increasing sizes 22
LTO Ultrium, special considerations for 188
managing 235
monitoring 378
moving files 397
moving files between 398
multiple, using to restore 901
next storage pool
definition 257
deleting 410
migration to 267, 388
overview 48
policy use 496, 502
primary 236
querying 378
renaming 405
search-and-selection order for active files 239
simultaneous-write function 327
updating 241
updating for disk, example 247, 258
using cache on disk 241, 280
validation of data 878
viewing information about 378
portable media
description of 5, 8, 537
restoring from 542
preemption
mount point 602

preemption *(continued)*
volume access 603
prefix, for recovery instructions 980
prefix, for recovery plan file 980
prefix, server 717
premigration 475
PREPARE command 989
PREVIEW parameter 743, 751
primary volumes unavailable for restore 901
private category, 349X library 42
private volumes 50
privilege class, administrator
description 835
granting authority 835
reducing 840
revoking all 840
privilege class, policy
description 834, 835
granting 839
revoking 839, 840
privilege class, storage
description 835
granting 839
reducing 839
revoking 840
problem determination
describing problem for IBM Software Support 1073
determining business impact for IBM Software Support 1073
migration 267
submitting a problem to IBM Software 1074
process
background 600
canceling 601
drive clean error checking 164
expiration 875
number for migration 241, 269
reclamation 363, 372
processor value unit 582
Product ID (PID) 582
profile
associating configuration information with 697
changing 697, 701, 702
default 700, 707
defining 697, 698
deleting 702, 704
description 697
getting information about 704
locking 701
problems with synchronization 713
unlocking 701
progressive incremental backup 12
protecting your data 34, 138
active-data pools 237
data deduplication 303
simultaneous-write function 327
protection options
client 7
server 13, 34
proxy node relationships 437
publications
download xv
PVU
estimation 582
VMware environment 444, 587

Q

query
authority 835
for general information 381
policy objects 523
QUERY ACTLOG command 762, 790
QUERY ADMIN command 838
query association output 564
QUERY BACKUPSETCONTENTS command 545
QUERY CONTENT command 384
QUERY COPYGROUP command 523, 755
QUERY DB command 661
QUERY DBSPACE command 661
QUERY DEVCLASS command 744
QUERY DIRSPACE 396
QUERY DOMAIN command 525
QUERY DRIVE command 153
QUERY DRMSTATUS command 980
QUERY ENABLED command 812
QUERY EVENT command 565, 615
QUERY FILESPACE command 445
QUERY LIBRARY command 151
QUERY LICENSE command 581
QUERY MGMTCLASS command 524
QUERY NODE command 438
QUERY OCCUPANCY command 393, 394, 395
QUERY OPTION command 782
QUERY POLICYSET command 525
QUERY PROCESS command 601, 760, 781
identification numbers of migration processes 389
information about data movement process 400
QUERY PVUESTIMATE 585
QUERY RESTORE command 466
QUERY RPFCONTENT command 991
QUERY RPFILE command 991
QUERY SCHEDULE command 557
QUERY SCRIPT command 625
QUERY SERVERGROUP command 720
QUERY SESSION command 462, 780
QUERY SHREDSTATUS command 534
QUERY STATUS command 782
QUERY STGPOOL command 378, 388, 391
QUERY SUBSCRIPTION command 709
QUERY SYSTEM command 783
QUERY VOLHISTORY command 606
QUERY VOLUME command 380, 381, 401
QUERYAUTH server option 835

R

random access storage
file systems 75
raw logical volumes 75
random mode for libraries 83
randomize, description of 572
raw logical volume 22, 76, 249, 251
re-installing 91
read-only access mode 255
read/write access mode 255
rebinding
description of 485
file to a management class 485
recalling a file
selective 476
transparent 476
receiver 797

RECLAIM STGPOOL command 367
reclaiming storage space 664
reclamation 373
  active-data pools 370
  aggregate reconstruction 364
  controlling duration 367
  delayed start of process 364
  delaying reuse of volumes 374, 875
  description of 21
  effects of collocation 374
  effects of DELETE FILESPACE 364
  multiple concurrent processes
    copy storage pools 371
    primary sequential access storage pools 241, 367
  off-site volume
    controlling when reclamation occurs 372
  setting a threshold for sequential storage pool 241, 363, 407
  starting reclamation manually 367
  storage pool for 241
  virtual volumes 370
  with single drive 368
RECONCILE VOLUMES command 729
reconstructing aggregates 364, 371, 401
recovery instructions file 1024
recovery log 633, 637
  active log 636, 637, 639
  active log mirror 638
  alternative file locations
    overview 672
    specifying with ARCHFAILOVERLOGDIRECTORY option or parameter 673
    specifying with ARCHLOGDIRECTORY parameter 673
    specifying with RECOVERYDIR parameter 674
  archive failover log 636, 639
  archive log 636, 638, 639
  description of 32, 633
  increasing the size 668
  log mirror 636, 638, 639
  managing 633
  monitoring 661
  out of space 668
recovery logs
  move to another directory 670
  relocating on a server 669
recovery plan file
  break out stanzas 1018
    using VBScript procedure 1018
  creating 989
  example 1021
  prefix 980
  stanzas 1018
recovery, disaster
  auditing storage pool volumes 885
  general strategy 722
  media 988
  methods 722
  providing 722
redistributing data for database directories 664
REGISTER ADMIN command 837
REGISTER LICENSE command 580
REGISTER NODE command 442
registering
  client option sets 419
  workstation 420
registration
  description of 417
  licensing for a client node 579
  licensing for an administrator 579
  managing client node 416, 425
  setting for a client node 417
  source server 420
relationships
  among clients, storage, and policy 478
remote access to clients 439
removable file system device
  labeling requirements 121
  REMOVABLEFILE device type, defining and updating 183
  support for 119, 183
removable media 45
REMOVE ADMIN command 841
REMOVE NODE command 434
remove replication server 955
REMOVE REPLSERVER 955
RENAME ADMIN command 840
RENAME FILESPACE command 759
RENAME NODE command 433
RENAME SCRIPT command 626
RENAME SERVERGROUP command 720
RENAME STGPOOL command 405
renamed file spaces 455
renaming
  administrator ID 840
  NAS node 203
  storage pool 405
renaming the host 604
renaming the server 604
replicate data 929
replicate NAS node 234
replication 929, 966
  node replication 907
  recovering an LDAP server 902
replication method 932, 933
replication performance 965, 966
replication target server 954
replication time 930
replication workload 966
reporting and monitoring 795
reporting ANR9999D messages 798
REQSYSAUTHOUTFILE server option 835
requirements for disk subsystems 69
resetting
  administrative password 840
  user password expiration 850
RESOURCETIMEOUT server option 67
restartable export 737
restartable restore session, client
  canceling 466
  interrupting, active 467
  requesting information about 466
restore
  selecting individual files 542
RESTORE DB command 599
restore operations 860
RESTORE STGPOOL command 904
restore to point-in-time, enabling for clients 522
RESTOREINTERVAL server optionrestore interval for restartable restore sessions 465, 473, 507
restoring
  clients, optimizing restore 237
  file 474

restoring *(continued)*
storage pools with incomplete volumes 901
restoring image data
from backup sets 542
restriction
ASCII characters in administrative Web interface 617
drive cleaning 161
non-root users performing backups 842
serial number detection 128
retain extra versions, description of 471, 498
retain only version, description of 471, 498
retaining data using Centera
overview 46
unsupported functions 245
retention grace period
description of archive 492
description of backup 492
for backup sets 540
using archive 492
using backup 492
RETEXTRA parameter 471, 498
RETONLY parameter 471, 498
retrieval date for files 281
retrieval from archive
archive package 549
file 475
reuse of sequential volumes
delaying 374, 875
storage pool volumes 141
volume pending state 383
rmstdev 86
rmstdev utility 86
roll-forward recovery 637
ROLLBACK command 631
routing commands 679
routing commands to servers 716
RUN command 626

S

SAN (storage area network)
client access to devices 54
device changes, detecting 127
LAN-free data movement 54
NDMP operations 55, 199
policy for clients using LAN-free data movement 519
sharing a library among servers 52, 110
storage agent role 54
SAN discovery, running for non-root users
correcting device special file names 85
obtaining device information for LAN-free 127
SANDISCOVERY 85
scale capacity 181
scenarios
policy configuration 516
schedule
administrative command 609
associating client node with 556
checking the log file 567
coordinating 568
copying 562, 614
database backups 864
day of the week 612
defining 555, 604, 611
deleting 562, 615
description of 553
expiration date 613
schedule *(continued)*
failed, querying 557, 566
for NAS file server backup 224
frequency of service 612
initial start date 612
initial time 612
maintenance 619
mode, setting 569
priority 613
querying 557
results of 565, 615
server administrative command 610
startup window 571, 612
type of action 613
uncertain status 566, 616
updating 611
viewing information about 557
schedule event
managing 565, 615
querying 565, 615
viewing information about 565, 615
schedule replication 933
scheduled operations, setting the maximum 571
scheduler workload, controlling 571
scheduling
administrative commands 31
scheduling mode
client-polling 569
overview of 569
selecting 569
server-prompted 569
setting on a client node 570
setting on the server 569
scratch category, 349X library 42
scratch volume
deleting 250, 606
description 50
FILE volumes 79
number allowed in a storage pool 241, 406
using in storage pools 253
script
maintenance 619
script, scheduling on client 558
script, server
continuation characters 621
copying 624
defining 617
deleting 626
EXIT statement 622
GOTO statement 623
IF clause 622
querying 625
renaming 626
routing commands in 717
running 626
running commands in parallel 620
running commands serially 620
substitution variables 621
updating 623, 624
used with SNMP 805
Web browser, restricted to ASCII entry 617
scripts 787
SCSI
automatic labeling of volumes 132
library with different tape technologies 180
SCSI libraries
define a library client 111

SCSI libraries *(continued)*
define a library server 111
SCSI library
connect to NAS file server 216
connecting to the server 216
SCSI tape library
setting up for NDMP operations 213
SEARCHMPQUEUE server option 67
secure sockets layer 828
configuration 828
Secure Sockets Layer 822
Secure Sockets Layer (SSL)
certificate
adding CA-signed 827
adding to key database 826
communication using 821
digital certificate file protection 869
Global Security Kit 823
security
client access, controlling 441
data encryption
3592 generation 2 530
3592 Generation 2 182
3592 Generation 2 and later 155
3592 generation 3 530
ECARTRIDGE 193, 194
IBM LTO Generation 4 155, 190, 530
Oracle StorageTek T10000B 155, 530
Oracle StorageTek T10000C 155, 530
features, overview 25
for the server 821
locking and unlocking administrators 842
locking and unlocking nodes 434
managing access 821, 834
password expiration for nodes 850
privilege class authority for administrators 835
Secure Sockets Layer (SSL) for node replication 956, 957
server options 835
security, replicating node data 907
SELECT command 784
customizing queries 786
selective backup 474, 488
selective recall 476
sending commands to servers 716
sequence number 459, 461
sequential mode for libraries 83
sequential storage pool
auditing a single volume in 883
auditing multiple volumes in 882
collocation 360
criteria for migrating 277
estimating space 377
migration threshold 275
reclamation 363
SERIAL command 620
serial number
automatic detection by the server 127, 170, 172
for a drive 170
for a library 170, 172
serialization parameter 471, 496, 502
server
backing up subfiles on 546
canceling process 601
changing the date and time 600
deleting 690
description of 3
disabling access 465
server *(continued)*
disaster recovery 34
enabling access 465
halting 598
importing subfiles from 548
instances
multiple on single system 597
owner ID 597
maintaining, overview 17
managing multiple 32
managing operations 579
managing processes 600
messages 798
monitoring 679
multiple instances 597
network of IBM Tivoli Storage Manager 32
network of Tivoli Storage Manager servers 677
options
adding or updating 605
prefix 717
protecting 34
querying about processes 601, 781
querying options 782
querying status 782
running multiple servers 597
setting the server name 604
starting 588, 594
AIX 588
alternative options 596
HP-UX 588
Linux 588
root user ID 592
Solaris 588
stand-alone mode 595
stopping 598
updating 690
viewing information about 782
viewing information about processes 601, 781
server console
logging events to 799
server console, description of 833
SERVER device type 237, 722
server group
copying 720
defining 719
deleting 721
member, deleting 721
moving a member 721
querying 720
renaming 720
updating description 720
server option
3494SHARED 67
ACSLS options 67
ACTIVELOGDIRECTORY 668, 671
ACTIVELOGSIZE 668
ASSISTVCRRECOVERY 67
AUDITSTORAGEstorage audit 581
changing with SETOPT command 605
CHECKTAPEPOS 67
COMMTIMEOUTcommunication timeout 462, 464
DRIVEACQUIRERETRY 67
EXPINTERVAL 507
EXPQUIET 507
IDLETIMEOUTidle timeout 462, 464, 780
NOPREEMPT 67, 602
NORETRIEVEDATEfile retrieval date 281

server option *(continued)*
overview 30
QUERYAUTH 835
REQSYSAUTHOUTFILE 835
RESOURCETIMEOUT 67
RESTOREINTERVALrestore interval 465, 473, 507
SEARCHMPQUEUE 67
THROUGHPUTDATATHRESHOLD 464
THROUGHPUTTIMETHRESHOLD 464
TXNGROUPMAXmaximum transaction group size 259
server options 605
ARCHFAILOVERLOGDIRECTORY 674
MIRRORLOGDIRECTORY 674
TECUTF8EVENT 804
server options file 868
server script
continuation characters 621
copying 624
defining 617
deleting 626
EXIT statement 622
GOTO statement 623
IF clause 622
querying 625
renaming 626
routing commands in 717
running 626
running commands in parallel 620
running commands serially 620
substitution variables 621
updating 623, 624
used with SNMP 805
Web browser, restricted to ASCII entry 617
server scripts 616
server session
states 462
server storage
active-data pools 237
client files, process for storing 5
concepts overview 15
considering user needs for recovering 64
deleting files from 410
evaluating 65
example 239
managing 18
monitoring 875
planning 65
server options affecting 67
tailoring definitions 755
using another IBM Tivoli Storage Manager server 722
using disk devices 69
using the storage hierarchy 266
server-free data movement
suggested usage 9
server-prompted scheduling 569
server-to-server communications, establishing
enterprise configuration 682
enterprise event logging 682
virtual volumes 690
server-to-server virtual volumes
deduplication 722
reclaiming 370
using to store data 722
session
canceling 463
negative number 466
server-initiated 427
session *(continued)*
setting the maximum percentage for scheduled operations 571
session, client
canceling 463
DSMC loop 461
held volume 461
managing 461
querying 462, 780
viewing information about 462, 780
sessions, maximum number for scheduled operations 964
SET ACCOUNTING command 793
SET ACTLOGRETENTION command 791
SET AUTHENTICATION command 855
SET CLIENTACTDURATION command 575
SET CONFIGMANAGER command 693, 696
SET CONFIGREFRESH command 709
SET CONTEXTMESSAGING command 798
SET CROSSDEFINE command 685, 688
SET DBREPORTMODE command 661
SET DRMCHECKLABEL command 983
SET DRMCOPYSTGPOOL command 980
SET DRMCOURIERNAME command 983
SET DRMDBBACKUPEXPIREDAYS command 983
SET DRMFILEPROCESS command 983
SET DRMINSTPREFIX command 980
SET DRMNOTMOUNTABLE command 983
SET DRMPLANPOSTFIX command 980
SET DRMPLANPREFIX command 980
SET DRMPRIMSTGPOOL command 980
SET DRMRPFEXPIREDAYS 991
SET DRMVAULTNAME command 983
SET EVENTRETENTION command 567, 616
SET INVALIDPWLIMIT command 853
SET LICENSEAUDITPERIOD command 581
SET MAXCMDRETRIES command 574
SET MAXSCHEDSESSIONS command 571
SET PASSEXP command 850
SET QUERYSCHEDPERIOD command 573
SET RANDOMIZE command 572
SET REGISTRATION command 417
SET RETRYPERIOD command 574
SET SCHEDMODES command 569
SET SERVERHLADDRESS command 685, 688
SET SERVERLLADDRESS command 685, 688
SET SERVERNAME command 604, 684, 685, 688
SET SERVERPASSWORD 684, 685, 688
SET SUBFILE 546
SET SUMMARYRETENTION 788
set up storage agent 828
SETOPT command 605
setting
clients to use subfile backup 547
compression 418
library mode 83
password 850
time interval for checking in volumes 179
setting data deduplication options 314
shared access, nodes 437
shared dynamic serialization, description of 496, 502
shared file system 77
shared SCSI library 110
shared static serialization, description of 496, 502
SHRED DATA command 534
shredding
BACKUP STGPOOL command 535
COPY ACTIVEDATA command 535

shredding *(continued)*
DEFINE STGPOOL command 535
DELETE FILESPACE, command 535
DELETE VOLUME, command 535
deleting empty volumes 411
deleting volumes with data 411
description 533
enforcing 535
EXPIRE INVENTORY command 535
EXPORT NODE command 535, 732
EXPORT SERVER command 535, 732
GENERATE BACKUPSET command 535, 537
MOVE DATA command 398, 535
setting up 534
UPDATE STGPOOL command 535
SHREDDING server option 534
simultaneous-write operations to primary and copy storage pools
drives 349, 350
inheritance model 335
mount points 348
storage pools 351
single drive device type 103
SnapLock
data protection, ensuring 515
event-based retention 514
reclamation 511
retention periods 511
WORM FILE volumes, setting up 515
SnapMirror to Tape 233
snapshot, using in backup 9, 11, 864
using in directory-level backups 232
SNMP
agent 805
communications 805
configuring 809
enabled as a receiver 797, 805
heartbeat monitor 797, 805
manager 805
subagent 805
software support
describing problem for IBM Software Support 1073
determining business impact for IBM Software Support 1073
submitting a problem 1074
Software Support
contacting 1072
Solaris Zones 95
adding devices to 95
configuring 95
global zone 95
listing devices in 97
non-global zone 95
removing devices from 98
Sony WORM media (AIT50 and AIT100) 138
source server 724
space
directories associated with FILE-type device classes 396
space requirements 928
space-managed file 475
special file names 84, 85, 86
SQL 784
SQL activity summary table 788
SQL SELECT * FROM PVUESTIMATE_DETAILS 585
ssl 822, 828
configuration 828
SSL (Secure Sockets Layer)
certificate
adding CA-signed 827
adding to key database 826
communication using 821
digital certificate file protection 869
SSL communication 822
SSLTCPADMINPORT
server option 825
SSLTCPPORT
server option 825
stand-alone mode 594, 595
standard label 19, 20
standard management class, copying 495
standard storage management policies, using 471
start time, randomizing for a schedule 572
starting server
instance user ID 590
starting the server
authorizing root user IDs 591
startup
alternative modes 596
server 588, 596
stand-alone mode 595
startup window, description of 572
static serialization, description of 496, 502
status of a volume in an automated library 51
stopping the server 598
storage agent 54
storage area network (SAN)
client access to devices 54
device changes, detecting 127
LAN-free data movement 54
NDMP operations 55, 199
policy for clients using LAN-free data movement 519
sharing a library among servers 52, 110
storage agent role 54
storage devices 101, 174, 175
storage hierarchy 21
copying active backup data 237
defining in reverse order 245, 257
establishing 256
example 239
for LAN-free data movement 257
how the server stores files in 258
next storage pool
definition 257
deleting 410
migration to 267, 388
restrictions 257
staging data on disk for tape storage 266
storage management policies
description of 26, 476
managing 469
tailoring 490
using standard 471
storage occupancy, querying 393
storage pool
3592, special considerations for 180
active-data pool 237
amount of space used 394
auditing a volume 875
comparing primary and copy types 408
copy 237
creating a hierarchy 256
data format 202, 241, 245
defining 241

storage pool *(continued)*
defining a copy storage pool 406
defining for disk, example 245, 257
defining for NDMP operations 213
defining for tape, example 245, 257
deleting 409
description of 236
destination in copy group 496, 502
determining access mode 241, 406
determining maximum file size 241
determining whether to use collocation 241, 353, 406
disk 22
duplicate, using to restore 901
enabling cache for disk 241, 280
estimating space for archived files on disk 377
estimating space for backed up files on disk 376
estimating space for disk 376
estimating space for sequential 377
estimating space in multiple 256
incomplete, using to restore 901
increasing sizes 22
LTO Ultrium, special considerations for 188
managing 235
monitoring 378
moving files 397
moving files between 398
multiple, using to restore 901
next storage pool
definition 257
deleting 410
migration to 267, 388
overview 48
policy use 496, 502
primary 236
protecting 871, 873, 874, 875, 886
querying 378
renaming 405
restoring 895, 896, 900, 901, 904
search-and-selection order for active files 239
simultaneous-write function 327
updating 241
updating for disk, example 247, 258
using cache on disk 241, 280
validation of data 878
viewing information about 378
volumes, restoring 897, 898, 899
storage privilege class
description 835
granting 839
reducing 839
revoking 840
storage volume
auditing 875
contents 384
formatting random access 76, 251
information about 381
labeling sequential access 129, 252
monitoring use 380
overview 48
preparing sequential access 129, 252
StorageTek devices 192
stub file 475
subfile backups
deleting 548
description of 546
example of 546
expiring 548
subfile backups *(continued)*
managing 547
restoring 548
subordinate storage pool 257
subscriber, deleting 714
subscription
defining 707, 709
deleting 710
scenario 708
subset node replication 932
substitution variables, using 621
support contract 1073
support information 1069
support subscription 1073
supported devices 40
swapping volumes in automated library 137
system catalog tables 784
system privilege class
revoking 840

T

table of contents 229
generating for a backup set 543
managing 205, 230
tape
capacity 196
exporting data 745
finding for client node 387
label prefix 19, 20
monitoring life 383
number of times mounted 383
planning for exporting data 744
recording format 177
reuse in storage pools 141
rotation 60, 141
scratch, determining use 241, 253, 406
setting mount retention period 179
volumes
initializing 19, 20
labeling 19, 20
tape labels
overwriting 94
target replication server
configuring 954
target server 724
TCP keepalive
enabling 207
overview 206
specifying connection idle time 207
TCP/IP 426
connect server to database 636
IPv4 426
IPv6 426
TECUTF8EVENT option 804
temporary disk space 644, 645
temporary space 644, 645
test replication 965
text editor
to work with client 422
threshold
migration, for storage pool
random access 269
sequential access 276
reclamation 241, 363, 407
throughput capability 965
THROUGHPUTDATATHRESHOLD server option 464

THROUGHPUTTIMETHRESHOLD server option 464
tiered data deduplication 313
tiering 313
time interval, setting for checking in volumes 179
timeout
  client session 464
Tivoli Directory Server
  configure for TLS 830
  configure for TLS on the CLI 831
Tivoli Enterprise Console 801
  setting up as a receiver 804
Tivoli event console 797, 801
Tivoli Storage Manager
  server network 677
Tivoli Storage Manager definitions 779
Tivoli Storage Manager device drivers 83
Tivoli Storage Manager for Space Management 495
  description 475
  files, destination for 495
  migration of client files
    description 475
    eligibility 490
  policy for, setting 490, 495
  premigration 475
  recall of migrated files 476
  reconciliation between client and server 476
  selective migration 475
  setting policy for 495
  simultaneous-write function, version support for 329
  space-managed file, definition 475
  stub file 475
TLS (Transport Layer Security)
  specifying communication ports 825
TLS communication 822
transactions, database 633, 675
transparent recall 476
Transport Layer Security (TLS) 823
  specifying communication ports 825
troubleshooting
  errors in database with external media manager 123
tsmdlist 86
tsmdlist utility 86
TXNBYTELIMIT client option 259
TXNGROUPMAX server option 259
type, device
  3590 175
  4MM 174, 175
  8MM 174, 175
  CENTERA 46
  DISK 174
  DLT 174, 175
  ECARTRIDGE 175
  FILE 174
  GENERICTAPE 174, 175
  LTO 176, 188
  multiple in a single library 57
  REMOVABLEFILE 174
  SERVER 174, 175, 724, 726
  VOLSAFE 192
  WORM 174, 175
typographic conventions xv

U

ulimits
  setting
    before server startup 589
Ultrium, LTO device type
  device class, defining and updating 188
  encryption 155, 190, 530
  WORM 138, 192
unavailable access mode
  description 255
  marked with PERMANENT parameter 149
uncertain, schedule status 566, 616
Unicode
  automatically renaming file space 450
  client platforms supported 447
  clients and existing backup sets 456
  deciding which clients need enabled file spaces 448
  description of 447
  displaying Unicode-enabled file spaces 456
  example of migration process 454
  file space identifier (FSID) 456
  how clients are affected by migration 453
  how file spaces are automatically renamed 451
  migrating client file spaces 449
  options for automatically renaming file spaces 450
Unicode versions
  planning for 452
UNIQUETDPTECEVENTS option 802
UNIQUETECEVENTS option 802
UNLOCK ADMIN command 842
UNLOCK NODE command 434
UNLOCK PROFILE command 701, 702
unplanned shutdown 598
unreadable files 884, 885
UPDATE ADMIN command 840
UPDATE ARCHIVE command 550
UPDATE BACKUPSET command 544
UPDATE CLIENTOPT command 461
UPDATE CLOPTSET command 461
UPDATE COPYGROUP command 496, 502
UPDATE DEVCLASS command 175
UPDATE DOMAIN command 494
UPDATE DRIVE command 154
UPDATE LIBRARY command 152
UPDATE LIBVOLUME command 50, 145
UPDATE MGMTCLASS command 495
UPDATE NODE command 428, 454, 459
UPDATE POLICYSET command 494
UPDATE RECOVERYMEDIA command 989
UPDATE SCHEDULE command 611
UPDATE SCRIPT command 623, 624
UPDATE SERVER command 690, 691
UPDATE VOLUME command 252
upgrading 91
URL for client node 416
user exit 797
user exit declarations 813, 1063
user ID, administrative
  creating automatically 442
  description of 416
  preventing automatic creation of 442
user limits
  setting
    before server startup 589
user-exit program 815, 1065

V

validate
  node data 530
VALIDATE LANFREE command 126

VALIDATE POLICYSET command 504
validating data
during a client session 529
for storage pool volumes 878
for virtual volumes 722
logical block protection 156
performance considerations for nodes 530
performance considerations for storage pools 881
variable, accounting log 793
VARY command 78
varying volumes on or off line 78
VERDELETED parameter 471, 498
VEREXISTS parameter 471, 498
versions data deleted, description of 471, 498
versions data exists, description of 471, 498
virtual file space mapping, command 231
virtual tape libraries 114, 116
configuring 114
managing 114
virtual tape library 42, 115, 116
configuring 116
storage capacity 115
virtual volume
performance expectations 725
virtual volumes, server-to-server
deduplication 722
reclaiming 370
using to store data 722
VIRTUALMOUNTPOINT client option 446
Vital Cartridge Records (VCR), corrupted condition 67
VMware host environment
hardware scans 444, 587
VOLSAFE device class 192
volume capacity 176
volume history 866
deleting information from 606
volume history file 79, 866
volume reuse 79
volumes
access preemption 603
access, controlling 140
allocating space for disk 76, 251
assigning to storage pool 251
auditing 147, 875
auditing considerations 875
automated library inventory 51
capacity, compression effect 197
checking in new volumes to library 133
checking out 145
contents, querying 384
defining to storage pools 252
delaying reuse 374, 875
deleting 411, 412, 606
detailed report 386
determining which are mounted 149, 744
disk storage 252
disk storage pool, auditing 882
dismounting 149
errors, read and write 382
estimated capacity 382
finding for client node 387
help in dsmc loop session 461
inventory maintenance 140
location 383
logical volume, raw 251
managing 144
monitoring life 383
volumes *(continued)*
monitoring movement of data 401
monitoring use 380
mount retention time 179
moving files between 397
number of times mounted 383
off-site, limiting number to be reclaimed 373
offsite, limiting number to be reclaimed 241
overview 50
pending status 383
private 50
querying contents 384
querying for general information 381
random access storage pools 236, 251, 253
raw logical volume 251
raw partition, using 76, 251
reclamation 368
removing from a library 145
reuse delay 374, 875
scratch 50
scratch, using 253
sequential 252
sequential storage pools 129, 252
setting access mode 255
standard report 385
status, in automated library 51
status, information on 382
swapping 137
updating 145, 252
using private 50, 51
varying on and off 78
VTL 115, 116

W

web administrative interface
description 17
Web administrative interface
limitation of browser for script definitions 617
Web backup-archive client
granting authority to 441
remote access overview 439
URL 416, 439
what's new
best practice xx
Windows Active Directory
configuring 832
wizard
client configuration 422
setup 422
workstation, registering 420
WORM devices and media
DLT WORM 138
IBM 3592 138
LTO WORM 138
maintaining volumes in a library 144
Oracle StorageTek T10000B drives 139
Oracle StorageTek T10000C drives 139
Quantum LTO3 138
reclamation of optical media 369
Sony AIT50 and AIT100 138
special considerations for WORM media 138
VolSafe
considerations for media 138
defining VOLSAFE device classes 192
WORM FILE and SnapLock 510
WORM parameter 192

writing data simultaneously to primary and copy storage pools
use during client storage operations 327



Product Number: 5608-E01
5608-E02
5608-E03

Printed in USA