Tivoli Storage Manager for Linux

Version 5.3

Administrator's Guide

GC23-4690-04

Tivoli® Storage Manager for Linux

IBM

Version 5.3

Administrator's Guide

GC23-4690-04

Note!
Before using this information and the product it supports, be sure to read the general information under Appendix D, "Notices," on page 765.

Fifth Edition (October 2005)

This edition applies to Version 5.3 of the IBM Tivoli Storage Manager for Linux (program numbers 5608-HSM, 5608-ISM, 5608-ISX, 5608-SAN, 5608-ARM) and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

Changes since the December 2004 edition are marked with a vertical bar (|) in the left margin. Ensure that you are using the correct edition for the level of the product.

Order publications through your sales representative or the branch office serving your locality.

Your feedback is important in helping to provide the most accurate and high-quality information. If you have comments about this book or any other IBM Tivoli Storage Manager documentation, please see "Support Information" on page xv.

© Copyright International Business Machines Corporation 1993, 2005. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this Guide xiii
IBM Tivoli Storage Manager Publications xiii
IBM Tivoli Storage Manager Library xiii
Accessing Publications Online xiv
Ordering Publications xv
Tivoli Technical Training xv
Support Information xv
Searching knowledge bases xv
Obtaining fixes xvi
Contacting IBM Software Support xvi
Conventions Used in This Guide xviii

Changes for Tivoli Storage Manager Version 5 xix
Technical Changes for Version 5 Release 3 — October 2005. xix
Technical Changes for Version 5 Release 3 — December 2004 xix
Technical Changes for Version 5 Release 2 — January 2004 xxiii
Technical Changes for Version 5 Release 2 — June 2003 xxv

Part 1. IBM Tivoli Storage Manager Basics 1

Chapter 1. Introducing IBM Tivoli Storage Manager 3
How IBM Tivoli Storage Manager Stores Client Data 5
Options for Data Protection 8
How Data Moves to Server Storage 14
Consolidating Backed-up Data for Clients . . . 15
How the Server Manages Storage 15
IBM Tivoli Storage Manager Device Support . . 15
Migrating Data through the Storage Hierarchy . 16
Removing Expired Data 16

Chapter 2. Getting Started with Tivoli Storage Manager 19
Interfaces to IBM Tivoli Storage Manager 19
Customizing the Server with the Server Options File 20
Configuring and Managing Server Storage 20
Using Hard Disk Devices with IBM Tivoli Storage Manager 20
Using Removable Media Devices with IBM Tivoli Storage Manager 21
Using Tape Volumes 22
Migrating Data from Disk to Tape 22
Managing Storage Pools and Volumes 23
Managing Client Operations. 24
Managing Client Nodes 24
Managing Security 25
Implementing Policies for Client Data 26
Scheduling Client Operations 28
Maintaining the Server 31
Managing Server Operations 31
Automating Server Operations 31
Managing the Database and Recovery Log . . . 32
Monitoring the IBM Tivoli Storage Manager Server 33
Working with a Network of IBM Tivoli Storage Manager Servers. 33
Exporting and Importing Data 34
Protecting IBM Tivoli Storage Manager and Client Data. 35
Protecting the Server 35
Managing Servers with the New Administration Center 36
Using the Administration Center 36
Quick paths to creating Tivoli Storage Manager objects 39
Quick paths to performing tasks 49
Functions and features in the Administration Center 55

Part 2. Configuring and Managing Server Storage 59

Chapter 3. Introducing Storage Devices 61
How to Use the Server Storage Chapters 61
IBM Tivoli Storage Manager Storage Devices . . . 62
IBM Tivoli Storage Manager Storage Objects . . . 62
Libraries 62
Drives 64
Device Class 64
Library, Drive, and Device Class 66
Storage Pool and Storage Pool Volume 66
Data Movers 68
Path 68
Server 68
IBM Tivoli Storage Manager Volumes. 68
The Volume Inventory for an Automated Library 69
Planning for Server Storage 70
Selecting a Device Configuration 71
Devices on a Local Area Network 71
Devices on a Storage Area Network 71
LAN-Free Data Movement 73
Network-Attached Storage 74
How IBM Tivoli Storage Manager Mounts and Dismounts Removable Media 76
How IBM Tivoli Storage Manager Uses and Reuses Removable Media 77
Configuring Devices 80
Mapping Devices to Device Classes 80
Mapping Storage Pools to Device Classes and Devices. 81

© Copyright IBM Corp. 1993, 2005

Chapter 4. Using Magnetic Disk Devices 83

Configuring Disk Devices 83
Requirements for Disk Subsystems, File Systems and Remote File Access 83
Comparing Random Access and Sequential Access Disk Devices 84
Configuring Random Access Volumes on Disk Devices. 87
Configuring FILE Sequential Volumes on Disk Devices. 88
Varying Disk Volumes Online or Offline. 89
Using Cache 89
Freeing Space on Disk. 89
Specifying Scratch FILE Volumes 90
Using FILE Volumes for Database Backups and Export Operations 90

Chapter 5. Attaching Devices to the Server System 91

Devices Supported by Tivoli Storage Manager. . . 91
Attaching a Manual Drive 91
Attaching an Automated Library Device. 92
Setting the Library Mode. 93
Installing and Configuring Device Drivers 93
Drivers for IBM Devices 93
Drivers for non-IBM Devices 93
Determining Device Special File Names 94
Installing Device Drivers for IBM SCSI Tape Devices. 95
Installing Device Drivers for IBM 349X Libraries 95
Obtaining Information About Your System's SCSI Devices. 95
Configuring Tivoli Storage Manager Passthru Drivers for Tape Drives and Libraries. 96
Installing and Configuring zSeries Linux Fibre Channel Adapter (zfcp) Device Drivers on the IBM zSeries System. 97

Chapter 6. Configuring Storage Devices 99

Device Configuration Overview 100
Mixing Device Types in Libraries. 100
Mixing Different Media Generations in Libraries 101
Mixed Media and Storage Pools 101
Server Options that Affect Storage Operations . . 102
Configuring SCSI Libraries used by One Server 102
Set up the Devices on the Server System . . . 102
Define the Devices to IBM Tivoli Storage Manager 103
Check in and Label Library Volumes 106
Using the Devices to Store Client Data 107
Configuring SCSI Libraries Shared Among Servers on a SAN 107
Setting up Server Communications 108
Set up the Device on the Server Systems and the SAN 108
Setting up the Library Manager Server 108
Setting up the Library Client Servers 110
Using the Devices to Store Client Data 110
Migrating DEFINE DRIVE and DEFINE PATH Definitions to the Library Manager Server . . . 110
Configuring IBM 3494 Libraries 111
Categories in an IBM 3494 Library 111
Configuring an IBM 3494 Library for Use by One Server 112
Set up the Device on the Server System . . . 112
Define the Devices to IBM Tivoli Storage Manager 112
Check in and Label Library Volumes 116
Using the Devices to Store Client Data 116
Sharing an IBM 3494 Library Among Servers. . . 117
Setting up Server Communications 117
Set up the Device on the Server System and the SAN 117
Setting up the Library Manager Server 118
Setting up the Library Client Servers 118
Using the Devices to Store Client Data 119
Migrating an IBM 3494 Library to Control by a Library Manager 119
Sharing an IBM 3494 Library by Static Partitioning of Drives 120
Set up the Device on the Servers 121
Define the Devices to IBM Tivoli Storage Manager ASTRO 121
Define the Devices to Tivoli Storage Manager JUDY 122
Configuring ACSLS-Managed Libraries. 123
Set up the Device on the Server System . . . 124
Define the Devices to IBM Tivoli Storage Manager 124
Setting up the Library Manager Server 127
Setting up the Library Client Servers 128
Check In and Label Library Volumes 128
Using the Devices to Store Client Data 129
Configuring Removable File Devices 129
Example of Removable File Support. 130
Labeling Requirements for Removable Files Devices 131
Configuring Libraries Controlled by Media Manager Programs 131
Setting up Tivoli Storage Manager to Work with an External Media Manager 131
Managing Externally Controlled IBM Tivoli Storage Manager Media 132
Using the Devices to Store Client Data 133
Configuring Manually Mounted Devices 133
Set up the Device on the Server System . . . 133
Define the Device to IBM Tivoli Storage Manager 133
Label Volumes 134
Using the Devices to Store Client Data 135
Configuring IBM Tivoli Storage Manager for LAN-free Data Movement 135
Validating your LAN-free Configuration . . . 136
Configuring IBM Tivoli Storage Manager for NDMP Operations. 136
Defining Devices and Paths 137
Defining Libraries. 137
Defining Drives 138
Defining Data Movers 139

Defining Paths 140
Recovering from Device Changes on the SAN . . 142

Chapter 7. Using NDMP for Operations with NAS File Servers . . 143

Requirements 143
Interfaces Used for NDMP Operations 145
Data Formats for Backup Operations Using NDMP 145
Planning for NDMP Operations 146
Planning for Tape Libraries and Drives used in NDMP Operations. 146
Determining Where to Attach the Tape Library Robotics 146
Determining How to Use the Drives in the Library 151
Planning for File-Level Restore 152
Directory-Level Backup and Restore 154
Configuring Tivoli Storage Manager for NDMP Operations 154
Step 1. Setting Up Tape Libraries for NDMP Operations 155
Step 2. Configuring Tivoli Storage Manager Policy for NDMP Operations 157
Step 3. Registering NAS Nodes with the Tivoli Storage Manager Server 158
Step 4. Defining a Data Mover for the NAS File Server 158
Step 5. Defining a Path to a Library 158
Step 6. Defining Tape Drives and Paths for NDMP Operations. 159
Step 7. Labeling Tapes and Checking Tapes into the Library 160
Step 8. Scheduling NDMP Operations 160
Step 9. Defining Virtual File Spaces 161
Backing Up and Restoring NAS File Servers Using NDMP 161
Performing File-Level Restore 162
Performing Directory-Level Backup and Restore 162
Managing NDMP Operations 163
Managing NAS File Server Nodes 164
Managing Data Movers Used in NDMP Operations 164
Dedicating a Tivoli Storage Manager Drive to NDMP Operations. 165
Managing Storage Pools for NDMP Operations 165
Managing Table of Contents 165

Chapter 8. Managing Removable Media Operations 167

Preparing Removable Media 167
Labeling Removable Media Volumes 168
Checking New Volumes into a Library 171
Special Considerations for WORM Tape Media 174
Managing the Volume Inventory 176
Controlling Access to Volumes 176
Reusing Tapes in Storage Pools 176
Setting Up a Tape Rotation 177
Reusing Volumes Used for Database Backups and Export Operations 178
Maintaining a Supply of Scratch Volumes . . . 178
Managing Volumes in Automated Libraries . . . 179
Changing the Status of a Volume 179
Removing Volumes from a Library 180
Returning Volumes to a Library 180
Managing a Full Library. 180
Auditing a Library's Volume Inventory. . . . 181
Maintaining a Supply of Scratch Volumes in an Automated Library 182
Performing Operations with Shared Libraries 182
Managing Server Requests for Media 183
Using the Administrative Client for Mount Messages 184
Mount Operations for Manual Libraries . . . 184
Handling Messages for Automated Libraries 184
Requesting Information about Pending Operator Requests 184
Replying to Operator Requests 184
Canceling an Operator Request 185
Responding to Requests for Volume Check-In 185
Determining Which Volumes Are Mounted . . 186
Dismounting an Idle Volume 186
Managing Libraries 186
Requesting Information About Libraries . . . 186
Updating Libraries 186
Deleting Libraries 187
Managing Drives 188
Requesting Information about Drives 188
Updating Drives 188
Cleaning Drives 189
Deleting Drives. 193
Managing Paths 193
Requesting Information About Paths 193
Updating Paths. 193
Deleting Paths 194
Managing Data Movers 194
Requesting Information About Data Movers . . 194
Updating Data Movers 194
Deleting Data Movers 194
Handling Tape Alert Messages 195

Chapter 9. Defining Device Classes 197

Defining and Updating Device Classes for Sequential Media 197
Defining and Updating Tape Device Classes . . 198
Defining and Updating REMOVABLEFILE Device Classes 203
Defining and Updating FILE Device Classes . . 204
Defining and Updating LTO Device Classes . . 207
Defining and Updating SERVER Device Classes 208
Defining and Updating VOLSAFE Device Classes 210
Requesting Information about a Device Class. . . 211
Deleting a Device Class 212
How Tivoli Storage Manager Fills Volumes . . . 213
Using Data Compression 213
Tape Volume Capacity and Data Compression 214

Chapter 10. Managing Storage Pools and Volumes. 217

Overview: Storage Pools. 218

Primary Storage Pool 218
Copy Storage Pool. 219
An Example of Server Storage. 219
Defining or Updating Primary Storage Pools 220
Task Tips for Storage Pools 225
Overview: Volumes in Storage Pools 226
Random Access Storage Pool Volumes 226
Sequential Access Storage Pool Volumes . . . 227
Preparing Volumes for Random Access Storage Pools 228
Preparing Volumes for Sequential Access Storage Pools 229
Updating Storage Pool Volumes 230
Access Modes for Storage Pool Volumes . . . 231
Overview: The Storage Pool Hierarchy 232
Setting Up a Storage Pool Hierarchy. 233
How the Server Groups Files before Storing . . 234
Where the Files Are Stored 235
How the Server Stores Files in a Storage Hierarchy 236
Using Copy Storage Pools to Back Up a Storage Hierarchy 237
Using the Hierarchy to Stage Client Data from Disk to Tape. 237
Migration of Files in a Storage Pool Hierarchy . . 238
Migration for Disk Storage Pools 239
Migration for Sequential Access Storage Pools 244
Migration and Copy Storage Pools 248
Using Cache on Disk Storage Pools 248
How the Server Removes Cached Files 249
Effect of Caching on Storage Pool Statistics . . 249
Using the Simultaneous Write Function to Protect Your Data 249
Function Overview 249
How Simultaneous Write Works 251
Operational Considerations. 255
Example: Simultaneous Write as Part of a Backup Strategy 257
Keeping a Client's Files Together: Collocation . . 258
The Effects of Collocation on Operations . . . 260
How the Server Selects Volumes with Collocation Enabled 261
How the Server Selects Volumes with Collocation Disabled 263
Turning Collocation On or Off. 264
Collocation on Copy Storage Pools 264
Planning for and Enabling Collocation 265
Reclaiming Space in Sequential Access Storage Pools 267
How IBM Tivoli Storage Manager Reclamation Works 267
Choosing a Reclamation Threshold 269
Lowering the Reclamation Threshold 270
Starting Reclamation Manually or in a Schedule 270
Specifying the Number of Reclamation Processes 270
Reclaiming Volumes in a Storage Pool with One Drive 271
Reclamation of Tape Volumes with High Capacity 272
Reclamation of Volumes with the Device Type of SERVER 272
Reclamation for Copy Storage Pools 273
How Collocation Affects Reclamation 276
Estimating Space Needs for Storage Pools 277
Estimating Space Needs in Random Access Storage Pools 277
Estimating Space Needs in Sequential Access Storage Pools 278
Monitoring Storage Pools and Volumes. 279
Monitoring Space Available in a Storage Pool 279
Monitoring the Use of Storage Pool Volumes 281
Monitoring Migration Processes 287
Monitoring the Use of Cache Space on Disk Storage 289
Requesting Information on the Use of Storage Space 291
Moving Files from One Volume to Another Volume 294
Moving Data to Other Volumes in the Same Storage Pool. 294
Moving Data to Another Storage Pool 295
Moving Data from an Offsite Volume in a Copy Storage Pool. 295
Procedure for Moving Data. 295
Moving Data for a Client Node 298
Moving Data for All File Spaces for One or More Nodes 298
Moving Data for Selected File Spaces for One Node 298
Requesting Information about the Data Movement Process 299
Preventing Incomplete Data Movement Operations 300
Renaming a Storage Pool 300
Defining a Copy Storage Pool 301
Example: Defining a Copy Storage Pool . . . 302
Comparing Primary and Copy Storage Pools 302
Deleting a Storage Pool 303
Deleting Storage Pool Volumes 304
Deleting an Empty Storage Pool Volume . . . 305
Deleting a Storage Pool Volume with Data . . 305

Part 3. Managing Client Operations 307

Chapter 11. Adding Client Nodes . . . 309

Overview of Clients and Servers as Nodes . . . 310
Installing Client Node Software 310
Registering Nodes with the Server 310
Accepting Default Closed Registration or Enabling Open Registration 310
Registering Nodes with Client Options Sets . . 312
Registering a Network-attached Storage File Server as a Node 312
Registering a Source Server as a Node on a Target Server 312
Registering an Application Programming Interface to the Server 313
Connecting Nodes with the Server 313
Required Client Options. 314

Non–Required Client Options 314
UNIX and Linux Client Options 314
Methods for Creating or Updating a Client Options File 314
Using a Text Editor 315
Using the Client Configuration Wizard 315
Comparing Network-Attached Nodes to Local Nodes 315
Adding Clients from the Administrative Command Line Client 316
Enabling Open Registration 316
Configuring the Client Options File to Connect with the Server 316
Example: Register Three Client Nodes Using the Administrative Command Line 316

Chapter 12. Managing Client Nodes 317

Managing Client Node Registration Techniques . . 317
Managing Nodes 318
Managing Client Nodes across a Firewall . . . 318
Updating Client Node Information 320
Renaming Client Nodes 320
Locking and Unlocking Client Nodes 320
Deleting Client Nodes 321
Consolidating Multiple Clients under a Single Client Node Name 321
Displaying Information about Client Nodes . . 323
Overview of Remote Access to Web Backup-Archive Clients 325
Managing Client Access Authority Levels . . . 327
Managing File Spaces 329
Client Nodes and File Spaces 329
Supporting Unicode-Enabled Clients 330
Displaying Information about File Spaces . . . 338
Moving Data for a Client Node 339
Deleting File Spaces 339
Managing Client Option Files 340
Creating Client Option Sets on the Server . . . 340
Managing Client Option Sets 342
Managing IBM Tivoli Storage Manager Sessions 343
Displaying Information about IBM Tivoli Storage Manager Sessions 343
Canceling an IBM Tivoli Storage Manager Session 344
When a Client Session is Automatically Canceled 345
Disabling or Enabling Access to the Server . . 346
Managing Client Restartable Restore Sessions 346
Managing IBM Tivoli Storage Manager Security 348
The Server Console 348
Administrative Authority and Privilege Classes 348
Managing Access to the Server and Clients . . 350
Managing IBM Tivoli Storage Manager Administrators 351
Managing Levels of Administrative Authority 353
Managing Passwords and Login Procedures . . 354

Chapter 13. Implementing Policies for Client Data 357

Basic Policy Planning 358
The Standard Policy 359
Getting Users Started 360
Changing Policy 360
File Expiration and Expiration Processing . . . 361
Client Operations Controlled by Policy 362
Backup and Restore 362
Archive and Retrieve 362
Client Migration and Recall 363
The Parts of a Policy 364
Relationships among Clients, Storage, and Policy 365
More on Management Classes 367
Contents of a Management Class 367
Default Management Classes 368
The Include-Exclude List 368
How Files and Directories Are Associated with a Management Class 370
How IBM Tivoli Storage Manager Selects Files for Policy Operations 372
Incremental Backup 372
Selective Backup 374
Logical Volume Backup 374
Archive 375
Automatic Migration from a Client Node . . . 375
How Client Migration Works with Backup and Archive 376
Creating Your Own Policies 376
Example: Sample Policy Objects 377
Defining and Updating a Policy Domain . . . 378
Defining and Updating a Policy Set 379
Defining and Updating a Management Class 380
Defining and Updating a Backup Copy Group 381
Defining and Updating an Archive Copy Group 387
Assigning a Default Management Class . . . 389
Validating and Activating a Policy Set 389
Assigning Client Nodes to a Policy Domain . . . 391
Running Expiration Processing to Delete Expired Files 391
Running Expiration Processing Automatically 392
Using Commands and Scheduling to Control Expiration Processing. 392
Additional Expiration Processing with Disaster Recovery Manager 392
Protection and Expiration of Archive Data 393
Data Retention Protection 393
Deletion Hold 394
Protecting Data Using the SnapLock Feature . . . 394
Reclamation and the SnapLock Feature 395
Set up SnapLock Volumes as Tivoli Storage Manager WORM FILE Volumes 398
Policy Configuration Scenarios 399
Configuring Policy for Direct-to-Tape Backups 399
Configuring Policy for Tivoli Storage Manager Application Clients 400
Policy for Logical Volume Backups 400
Configuring Policy for NDMP Operations . . . 401
Configuring Policy for LAN-free Data Movement 402
Policy for IBM Tivoli Storage Manager Servers as Clients. 404

Setting Policy to Enable Point-in-Time Restore for Clients 404
Distributing Policy Using Enterprise Configuration 405
Querying Policy 405
Querying Copy Groups 406
Querying Management Classes 406
Querying Policy Sets 407
Querying Policy Domains 407
Deleting Policy 408
Deleting Copy Groups 408
Deleting Management Classes 408
Deleting Policy Sets 409
Deleting Policy Domains 409

Chapter 14. Managing Data for Client Nodes 411
Validating a Node's Data 411
Performance Considerations for Data Validation 412
Validating a Node's Data During a Client Session 412
Creating and Using Client Backup Sets 412
Generating Client Backup Sets on the Server 413
Restoring Backup Sets from a Backup-Archive Client 414
Moving Backup Sets to Other Servers 415
Managing Client Backup Sets 415
Enabling Clients to Use Subfile Backup 418
Example of Subfile Backups 418
Setting Up Clients to Use Subfile Backup . . . 419
Managing Subfile Backups 419
Optimizing Restore Operations for Clients 420
Environment Considerations 421
Restoring Entire File Systems 421
Restoring Parts of File Systems 421
Restoring Databases for Applications 422
Restoring Files to a Point in Time 423
Concepts for Client Restore Operations 423
Managing Archive Data 426
Archive Operations Overview 426
Managing Performance for Archive Operations 427
Managing Storage Usage for Archives 429
Auditing Archive Descriptions 431
Getting Information about Archives 432

Chapter 15. Scheduling Operations for Client Nodes 433
Prerequisites to Scheduling Operations 433
Scheduling a Client Operation 434
Defining Client Schedules 434
Associating Client Nodes with Schedules . . . 435
Starting the Scheduler on the Clients 435
Displaying Schedule Information 436
Creating Schedules for Running Command Files 437
Updating the Client Options File to Automatically Generate a New Password 438
Comparing IBM Tivoli Storage Manager Scheduling Across Operating Systems and Components 438
Commands for Scheduling Client Operations . . . 439

Chapter 16. Managing Schedules for Client Nodes 441
Managing IBM Tivoli Storage Manager Schedules 441
Verifying that the Schedule Ran 441
Adding New Schedules 442
Copying Existing Schedules 442
Modifying Schedules 442
Deleting Schedules 442
Displaying Information about Schedules . . . 443
Managing Node Associations with Schedules . . . 443
Adding New Nodes to Existing Schedules . . . 444
Moving Nodes from One Schedule to Another 444
Displaying Nodes Associated with Schedules 444
Removing Nodes from Schedules 444
Managing Event Records 444
Displaying Information about Scheduled Events 445
Managing Event Records in the Server Database 446
Managing the Throughput of Scheduled Operations 446
Modifying the Default Scheduling Mode . . . 447
Specifying the Schedule Period for Incremental Backup Operations 449
Balancing the Scheduled Workload for the Server 449
Controlling How Often Client Nodes Contact the Server 451
Specifying One-Time Actions for Client Nodes . . 452
Determining How Long the One-Time Schedule Remains Active 453

Part 4. Maintaining the Server . . . 455

Chapter 17. Managing Server Operations 457
Licensing IBM Tivoli Storage Manager 457
Registering Licensed Features 458
Saving Your Licenses 459
Monitoring Licenses 459
Starting and Halting the Server 459
Starting the Server 460
Halting the Server 462
Moving the IBM Tivoli Storage Manager Server 463
Changing the Date and Time on the Server . . . 464
Managing Server Processes 465
Requesting Information about Server Processes 466
Canceling Server Processes 466
Preemption of Client or Server Operations . . 467
Setting the Server Name 468
Adding or Updating Server Options 468
Adding or Updating a Server Option without Restarting the Server 469
Automatic Tuning of Server Options 469
Getting Help on Commands and Error Messages 469

Chapter 18. Automating Server Operations 471
Automating a Basic Administrative Command Schedule 471
Defining the Schedule 472

Verifying the Schedule 472
Tailoring Schedules 473
Example: Defining and Updating an Administrative Command Schedule 475
Copying Schedules 476
Deleting Schedules 476
Managing Scheduled Event Records 476
Querying Events 476
Removing Event Records from the Database . . 477
IBM Tivoli Storage Manager Server Scripts . . . 477
Defining a Server Script 478
Managing Server Scripts. 481
Running a Server Script 484
Using Macros 484
Writing Commands in a Macro 485
Writing Comments in a Macro. 485
Using Continuation Characters 486
Using Substitution Variables in a Macro . . . 486
Running a Macro 487
Controlling Command Processing in a Macro 487

Chapter 19. Managing the Database and Recovery Log 489

How IBM Tivoli Storage Manager Processes Transactions 490
Performance Considerations: Transferring Files as a Group between Client and Server 490
How IBM Tivoli Storage Manager Manages Space 492
Available Space. 492
Assigned Capacity. 493
Utilization 493
Estimating and Monitoring Database and Recovery Log Space Requirements. 494
Monitoring the Database and Recovery Log . . 495
Increasing the Size of the Database or Recovery Log. 496
Automating the Increase of the Database or Recovery Log 497
Recovering When the Recovery Log Runs Out of Space 498
Manually Increasing the Database or Recovery Log. 499
Decreasing the Size of the Database or Recovery Log. 501
Step 1: Determining If Volumes Can Be Deleted 501
Step 2: Reducing the Capacity of the Database or Recovery Log 502
Step 3: Deleting a Volume from the Database or Recovery Log 502
Optimizing Database and Recovery Log Performance. 503
Optimal Database Volume Allocation 503
Utilizing Hardware with Non-Volatile Write Cache 504
Adjusting the Database Buffer Pool Size . . . 504
Manually Adjusting the Database Buffer Pool Size 504
Adjusting the Recovery Log Buffer Pool Size 505
Estimating Database Reorganization 506
Reorganizing the Database 508

Chapter 20. Monitoring the IBM Tivoli Storage Manager Server. 513

Using IBM Tivoli Storage Manager Queries to Display Information 513
Requesting Information about IBM Tivoli Storage Manager Definitions 513
Requesting Information about Client Sessions 514
Requesting Information about Server Processes 515
Requesting Information about Server Settings 516
Querying Server Options 516
Querying the System 517
Using SQL to Query the IBM Tivoli Storage Manager Database. 518
Using the ODBC Driver 518
Issuing SELECT Commands 518
Using SELECT Commands in IBM Tivoli Storage Manager Scripts. 521
Canceling a SELECT Command 522
Controlling the Format of SELECT Results . . 522
Querying the SQL Activity Summary Table . . 522
Creating Output for Use by Another Application 523
Using the IBM Tivoli Storage Manager Activity Log 523
Requesting Information from the Activity Log 524
Setting a Retention Period for the Activity Log 525
Setting a Size Limit for the Activity Log . . . 525
Logging IBM Tivoli Storage Manager Events to Receivers 526
Controlling Event Logging 527
Logging Events to the IBM Tivoli Storage Manager Server Console and Activity Log. . . 528
Logging Events to a File Exit and a User Exit 528
Logging Events to the Tivoli Enterprise Console 529
Logging Events to an SNMP Manager 533
Enterprise Event Logging: Logging Events to Another Server 538
Querying Event Logging 540
Using Tivoli Decision Support. 540
Scheduling the Decision Support Loader with IBM Tivoli Storage Manager 541
Monitoring IBM Tivoli Storage Manager Accounting Records 541
Daily Monitoring Scenario 543
Using IBM Tivoli Storage Manager Operational Reporting 543
Operational Reporting Concepts 543
Installing the IBM Tivoli Storage Manager Operational Reporting Stand-alone Package . . 546
Generating a Comprehensive Report at Regular Intervals 547
Generating a Comprehensive Monitor at Regular Intervals 551
Completing the Set Up 553
Viewing Reports and Other Features from the Management Console. 555
Generating Reports and Monitors from the Command Line. 558

Chapter 21. Working with a Network of IBM Tivoli Storage Manager Servers 563

Concepts for Working with a Network of Servers 563
Configuring and Managing Servers: Enterprise Configuration 564
Performing Tasks on Multiple Servers 565
Central Monitoring 565
Storing Data on Another Server 566
Example Scenarios. 566
Planning for Enterprise Administration. 567
Setting Up Communications Among Servers . . . 568
Setting Up Communications for Enterprise Configuration and Enterprise Event Logging . . 568
Setting Up Communications for Command Routing 571
Updating and Deleting Servers 574
Setting Up an Enterprise Configuration. 575
Enterprise Configuration Scenario 576
Creating the Default Profile on a Configuration Manager 579
Creating and Changing Configuration Profiles 580
Getting Information about Profiles 587
Subscribing to a Profile 589
Refreshing Configuration Information 593
Returning Managed Objects to Local Control 594
Setting Up Administrators for the Servers . . . 594
Handling Problems with Synchronization of Profiles 595
Switching a Managed Server to a Different Configuration Manager 595
Deleting Subscribers from a Configuration Manager 596
Renaming a Managed Server 596
Performing Tasks on Multiple Servers 596
Working with Multiple Servers Using the Administration Center 596
Routing Commands 597
Setting Up Server Groups 599
Querying Server Availability 601
Using Virtual Volumes to Store Data on Another Server 601
Setting Up Source and Target Servers for Virtual Volumes 602
Performing Operations at the Source Server . . 604
Reconciling Virtual Volumes and Archive Files 606

Chapter 22. Exporting and Importing Data 609

Data That Can Be Exported and Imported. . . . 609
Exporting Restrictions 610
Deciding What Information to Export 610
Deciding When to Export 611
Exporting Data Directly to Another Server. . . . 612
Options to Consider Before Exporting 612
Preparing to Export to Another Server for Immediate Import 614
Monitoring the Server-to-Server Export Process 615
Exporting Administrator Information to Another Server 615
Exporting Client Node Information to Another Server 615
Exporting Policy Information to Another Server 616
Exporting Server Data to Another Server . . . 616
Exporting and Importing Data Using Sequential Media Volumes. 617
Preparing to Export or Import Data (Sequential Media) 617
Exporting Tasks 619
Importing Data from Sequential Media Volumes 621
Monitoring Export and Import Processes . . . 631
Exporting and Importing Data from Virtual Volumes 634

Part 5. Protecting the Server . . . 637

Chapter 23. Protecting and Recovering Your Server 639

Levels of Protection 640
Storage Pool Protection: An Overview 640
How Restore Processing Works 640
How the Destroyed Volume Access Mode Works 641
Database and Recovery Log Protection: An Overview. 641
Mirroring. 642
Database and Recovery Log Protection 642
Snapshot Database Protection 644
Mirroring the Database and Recovery Log. . . . 644
Separating Disk Volume Copies On Separate Physical Disks When Mirroring the Database and Recovery Log 645
Defining Database or Recovery Log Mirrored Volume Copies 645
Specifying Mirroring and Database Page Shadowing Server Options 646
Requesting Information about Mirrored Volumes 647
Backing Up Storage Pools 648
Scheduling Storage Pool Backups. 649
Example: Simple Hierarchy with One Copy Storage Pool. 650
Using Simultaneous Write to Copy Storage Pools 650
Using Multiple Copy Storage Pools 651
Delaying Reuse of Volumes for Recovery Purposes 651
Backing Up the Database 652
Defining Device Classes for Backups 653
Setting the Recovery Log Mode 653
Estimating the Size of the Recovery Log . . . 653
Scheduling Database Backups 654
Automating Database Backups 655
Saving the Volume History File 656
Saving the Device Configuration File 658
Saving the Server Options 660
Saving the Database and Recovery Log Information 660
Doing Full and Incremental Backups 660
Doing Snapshot Database Backups 661
Recovering Your Server Using Database and Storage Pool Backups. 661
Restoring a Database to a Point-in-Time . . . 662
Restoring a Database to its Most Current State 666
Restoring Storage Pools 666

Restoring Your Server Using Mirrored Volumes 669
Restoring Storage Pool Volumes 669
What Happens When a Volume Is Restored . . 670
When a Volume Restoration Is Incomplete. . . 670
Auditing a Storage Pool Volume 671
What Happens When You Audit Storage Pool Volumes 671
Data Validation During Audit Volume Processing 673
Auditing a Volume in a Disk Storage Pool. . . 676
Auditing Multiple Volumes in a Sequential Access Storage Pool 677
Auditing a Single Volume in a Sequential Access Storage Pool. 677
Auditing Volumes by Date Written 677
Auditing Volumes in a Specific Storage Pool . . 678
Defining a Schedule to Audit Volumes on a Regular Basis 678
Correcting Damaged Files 678
Maintaining the Integrity of Files. 678
Restoring Damaged Files 679
Backup and Recovery Scenarios 680
Protecting Your Database and Storage Pools . . 680
Recovering to a Point-in-Time from a Disaster 681
Recovering a Lost or Damaged Storage Pool Volume 683
Restoring a Library Manager Database 684
Restoring a Library Client Database 685

Chapter 24. Using Disaster Recovery Manager 687
Querying Defaults for the Disaster Recovery Plan File. 688
Specifying Defaults for the Disaster Recovery Plan File 688
Specifying Defaults for Offsite Recovery Media Management 690
Specifying Recovery Instructions for Your Site . . 692
Specifying Information About Your Server and Client Node Machines 693
Specifying Recovery Media for Client Machines 695
Creating and Storing the Disaster Recovery Plan 696
Storing the Disaster Recovery Plan Locally . . 697
Storing the Disaster Recovery Plan on a Target Server 697
Managing Disaster Recovery Plan Files Stored on Target Servers 697
Displaying Information about Recovery Plan Files 697
Displaying the Contents of a Recovery Plan File 698
Restoring a Recovery Plan File 698
Expiring Recovery Plan Files Automatically . . 699
Deleting Recovery Plan Files Manually 699
Moving Backup Media 699
Moving Backup Volumes Offsite 701
Moving Backup Volumes Onsite 702
Summary of Disaster Recovery Manager Daily Tasks 704
Staying Prepared for a Disaster 705
Recovering From a Disaster 706
Server Recovery Scenario 707
Client Recovery Scenario 709
Recovering When Using Different Hardware at the Recovery Site 712
Automated SCSI Library at the Original Site and a Manual SCSI Library at the Recovery Site . . 712
Automated SCSI Library at the Original and Recovery Sites 713
Managing Copy Storage Pool Volumes at the Recovery Site 714
Disaster Recovery Manager Checklist 714
The Disaster Recovery Plan File 717
Breaking Out a Disaster Recovery Plan File . . 717
Structure of the Disaster Recovery Plan File . . 717
Example Disaster Recovery Plan File 720

Appendix A. External Media Management Interface Description . . 741
CreateProcess Call. 741
Processing during Server Initialization 742
Processing for Mount Requests 742
Processing for Release Requests 742
Processing for Batch Requests 743
Error Handling. 743
Begin Batch Request 744
End Batch Request 744
Volume Query Request 744
Initialization Requests 745
Volume Eject Request. 746
Volume Release Request. 747
Volume Mount Request 747
Volume Dismount Request 750

Appendix B. User Exit and File Exit Receivers 751
Sample User Exit Declarations. 752
Sample User Exit Program 755
Readable Text File Exit (FILETEXTEXIT) Format 756

Appendix C. Setting up the Secure Sockets Layer (SSL) 759
Setting up SSL for Integrated Solutions Console 759
Setting up the Lightweight Directory Access Protocol (LDAP) over SSL 762

Appendix D. Notices 765
Programming Interface 766
Trademarks 767

Glossary 769

Index 779

About this Guide

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.

IBM Tivoli Storage Manager Publications

This section lists publications in the IBM Tivoli Storage Manager library and related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

IBM Tivoli Storage Manager Library

The following documents are available in theTivoli Storage Manager library:

Publication Title	Order Number
IBM Tivoli Storage Management Products License Information	GH09-4572
IBM Tivoli Storage Manager Messages	SC32-9090
IBM Tivoli Storage Manager for Linux Administrator's Guide	GC23-4690
IBM Tivoli Storage Manager for Linux Administrator's Reference	GC23-4691
IBM Tivoli Storage Manager for Linux Installation Guide	GC32-1599
IBM Tivoli Storage Manager Basic Concepts Poster	SC32-9464

The following table lists Tivoli Storage Manager storage agent publications.

Publication Title	Order Number
IBM Tivoli Storage Manager for SAN for AIX Storage Agent User's Guide	GC32-0771
IBM Tivoli Storage Manager for SAN for HP-UX Storage Agent User's Guide	GC32-0727
IBM Tivoli Storage Manager for SAN for Linux Storage Agent User's Guide	GC23-4693
IBM Tivoli Storage Manager for SAN for Sun Solaris Storage Agent User's Guide	GC32-0781
IBM Tivoli Storage Manager for SAN for Windows Storage Agent User's Guide	GC32-0785

The following table lists Tivoli Storage Manager client publications.

Publication Title	Order Number
IBM Tivoli Storage Manager for Space Management for UNIX and Linux: User's Guide	GC32-0794
IBM Tivoli Storage Manager for Macintosh: Backup-Archive Clients Installation and User's Guide	GC32-0787

© Copyright IBM Corp. 1993, 2005

Publication Title	Order Number
IBM Tivoli Storage Manager for NetWare: Backup-Archive Clients Installation and User's Guide	GC32-0786
IBM Tivoli Storage Manager for UNIX and Linux: Backup-Archive Clients Installation and User's Guide	GC32-0789
IBM Tivoli Storage Manager for Windows: Backup-Archive Clients Installation and User's Guide	GC32-0788
IBM Tivoli Storage Manager Using the Application Program Interface	GC32-0793

The following table lists publications for application protection products.

Publication Title	Order Number
IBM Tivoli Storage Manager for Application Servers: Data Protection for WebSphere Application Server Installation and User's Guide	SC32-9075
IBM Tivoli Storage Manager for Databases: Data Protection for Microsoft SQL Server Installation and User's Guide	SC32-9059
IBM Tivoli Storage Manager for Databases: Data Protection for Oracle for UNIX and Linux Installation and User's Guide	SC32-9064
IBM Tivoli Storage Manager for Databases: Data Protection for Oracle for Windows Installation and User's Guide	SC32-9065
IBM Tivoli Storage Manager for Databases: Data Protection for Informix Installation and User's Guide	SH26-4095
IBM Tivoli Storage Manager for Enterprise Resource Planning: Data Protection for mySAP Installation and User's Guide for DB2	SC33-6341
IBM Tivoli Storage Manager for Enterprise Resource Planning: Data Protection for mySAP Installation and User's Guide for Oracle	SC33-6340
IBM Tivoli Storage Manager for Hardware: Data Protection for EMC Symmetrix for R/3 Installation and User's Guide	SC33-6386
IBM Tivoli Storage Manager for Hardware: Data Protection for Enterprise Storage Server for DB2 UDB Installation and User's Guide	SC32-9060
IBM Tivoli Storage Manager for Hardware: Data Protection for FlashCopy Devices for Oracle Installation and User's Guide	GC32-1772
IBM Tivoli Storage Manager for Hardware: Data Protection for FlashCopy Devices for mySAP Installation and User's Guide for DB2 UDB	SC33-8208
IBM Tivoli Storage Manager for Hardware: Data Protection for FlashCopy Devices for mySAP Installation and User's Guide for Oracle	SC33-8207
IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for UNIX, Linux, and OS/400 Installation and User's Guide	SC32-9056
IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for Windows Installation and User's Guide	SC32-9057
IBM Tivoli Storage Manager for Mail: z/OS Edition Licensed Program Specifications	GC32-0838
IBM Tivoli Storage Manager for Mail: Data Protection for Microsoft Exchange Server Installation and User's Guide	SC32-9058

Accessing Publications Online

The product CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both. To access the publications using

a Web browser, open the infocenter.html file. The file is in the appropriate publications directory on the product CD.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center Web site. Access the Tivoli software information center by first going to the Tivoli software library at the following Web address:

http://www.ibm.com/software/tivoli/library/

Scroll down and click the **Product manuals** link. In the Tivoli Technical Product Documents Alphabetical Listing window, click the <**Tivoli Storage Manager**> link to access the product library at the Tivoli software information center.

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File → Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering Publications

You can order many Tivoli publications online at the following Web site:

www.elink.ibmlink.ibm.com/public/
applications/publications/cgibin/pbi.cgi

You can also order publications in the United States by calling 800-879-2755.

Tivoli Technical Training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site:

http://www.ibm.com/software/tivoli/education

Support Information

This section describes the following options for obtaining support for IBM products:

- "Searching knowledge bases"
- "Obtaining fixes" on page xvi
- "Contacting IBM Software Support" on page xvi

Searching knowledge bases

If you have a problem with your IBM software, begin by searching the available knowledge bases to determine whether the resolution to your problem is already documented.

Search the information center on your local system or network

IBM Tivoli Storage Manager provides an information center on a CD. You can run the information center from the CD, or copy it to your local computer or an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, reference information, and support documents.

Search the Internet

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem. To search multiple Internet resources for your product, go to the support web site for the product (www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html) and look for the section to search the support knowledge base. From this section, you can search a variety of resources including:

- IBM technotes
- IBM downloads
- IBM Redbooks
- Forums and newsgroups

Obtaining fixes

A product fix might be available to resolve your problem. You can determine what fixes are available for your IBM software product by checking the product support Web site:

1. Go to the IBM Software Support Web site (http://www.ibm.com/software/support).
2. Under **Products A - Z**, select your product name. This opens a product-specific support site.
3. Under **Self help**, follow the link to **All Updates**, where you will find a list of fixes, fix packs, and other service updates for your product. For tips on refining your search, click **Search tips**.
4. Click the name of a fix to read the description and optionally download the fix.

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

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

For more information about types of fixes, see the *Software Support Handbook* (http://techsupport.services.ibm.com/guides/handbook.html).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus, and Rational products, as well as DB2 and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage in one of the following ways:
  - **Online**: Go to the Passport Advantage Web page (www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home) and click **How to Enroll**
  - **By phone**: For the phone number to call in your country, go to the IBM Software Support Web site (http://techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region.
- For IBM eServer software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web page (http://www.ibm.com/servers/eserver/techsupport.html).

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the contacts page of the IBM Software Support Handbook on the Web (http://techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region for phone numbers of people who provide support for your location.

Follow the steps in this topic to contact IBM Software Support:

1. Determine the business impact of your problem.
2. Describe your problem and gather background information.
3. Submit your problem to IBM Software Support.

Determine the business impact of your problem

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

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.

Describe your problem and gather background information

When explaining a problem to IBM, 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 currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.

Submit your problem to IBM Software Support

You can submit your problem in one of two ways:

- **Online**: Go to the "Submit and track problems" page on the IBM Software Support site (http://www.ibm.com/software/support/probsub.html). Enter your information into the appropriate problem submission tool.
- **By phone**: For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web (http://techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround for you to implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM product support Web pages daily, so that other users who experience the same problem can benefit from the same resolutions.

For more information about problem resolution, see Searching knowledge bases and Obtaining fixes.

Conventions Used in This Guide

- Command to be entered on the Linux command line:

  ```
  > dsmadmc
  ```

- Command to be entered on the command line of an administrative client:

  ```
  query devclass
  ```

Changes for Tivoli Storage Manager Version 5

This section summarizes changes that have been made to the Tivoli Storage Manager product and this publication.

Technical Changes for Version 5 Release 3 — October 2005

The following changes have been made to the product for this edition:

ACSLS Library Support

Library support has been enhanced to allow the sharing of ACSLS libraries. This implementation will provide the basic support to share ACSLS libraries across Tivoli Storage Manager servers in the same way that SCSI libraries are shared. Support for LAN-Free data movement using libraries that are controlled by StorageTek's ACSLS interface has also been provided. As a result, the use of a third party product is now optional.

See Chapter 6, "Configuring Storage Devices," on page 99 and the *Administrator's Reference* for more information.

Changes to the Administration Center

The Administration Center, introduced in Version 5.3 as a replacement for the administrative Web interface, now supports disaster recovery manager functions. The Tivoli Storage Manager server must be licensed for IBM Tivoli Storage Manager Extended Edition to take advantage of these functions.

Version 5.3 of the Administration Center required you to click on the Go button after making a selection from the table action list. The Go button has been eliminated reducing the number clicks required to perform an action.

New WORM Support

Because they permit write operations only, WORM (write once, read many) media lets you protect critical data from accidental or deliberate deletion. Tivoli Storage Manager now supports DLT WORM and LTO WORM drives and tape media. WORM technology for DLT devices is currently available for SDLT600 tape drives only. For more information, see "Special Considerations for WORM Tape Media" on page 174.

Support for Automatic Volume Labeling in Non-SCSI Libraries

Automatic volume labeling support has been enhanced. This implementation will enable the use of the AUTOLABEL parameter in place of DSMLABEL or LABEL LIBVOLUME for non-SCSI libraries. Using the AUTOLABEL parameter allows volumes to be labeled as they are needed for backup operations and eliminates having to label all volumes before they are checked in to Tivoli Storage Manager inventory. See the *Administrator's Reference* for more information.

Technical Changes for Version 5 Release 3 — December 2004

The following changes have been made to the product for this edition:

Accurate SAN Device Mapping

Device IDs within a SAN environment change when a reset or other environmental changes occur. With accurate SAN device mapping, Tivoli

© Copyright IBM Corp. 1993, 2005

Storage Manager can now detect SAN changes and automatically make the appropriate processing changes to the server definitions. QUERY SAN can now be issued to display all devices on the SAN, along with information such as the path, device-type, and serial number.

See Chapter 6, "Configuring Storage Devices," on page 99 for more information.

Activity Log Management

The activity log can now be managed based either on maximum size for the log or retention time. The new option for size-based activity log management gives administrators greater control over the amount of space the activity log occupies. Additional information about the activity log is now displayed when the server status is queried.

See Chapter 20, "Monitoring the IBM Tivoli Storage Manager Server," on page 513 for more information.

Check-In and Check-Out Enhancements

The amount of user intervention required for check in and check out functions has been reduced.

See Chapter 8, "Managing Removable Media Operations," on page 167 and Chapter 24, "Using Disaster Recovery Manager," on page 687 for more information.

Collocation by Group

Collocation by group is now supported. Groups of nodes can be defined, and the server can then collocate data based on these groups. Collocation by group can yield the following benefits:

- Reduce unused tape capacity by allowing more collocated data on individual tapes
- Minimize mounts of target volumes
- Minimize database scanning and reduce tape passes for sequential-to-sequential transfer

For newly defined storage pools, the default storage pool collocation setting is now GROUP.

Note: During collocation processing the message ANR1142I will be replaced with ANR1176I.

See Chapter 10, "Managing Storage Pools and Volumes," on page 217 for more information.

Communications Options

There is now a shared memory communications option between the Tivoli Storage Manager server for Linux® and the backup-archive client for Linux. It can be used to perform backups, archives, restores, and retrieves. You can also enable the shared memory communication protocol in the Linux storage agent for communication with the Linux backup-archive client.

Note: While using shared memory, if a message queue must be created, but the system limit for the maximum number of message queues (MSGMNI) will be exceeded if a message queue is created, the following message will be issued:

```
ANR9999D shmcomm.c(1598): ThreadId<39>
Error from msgget (2), errno = 28
```

To find the maximum number of message queues allowed on your system, issue the following command:

```
cat /proc/sys/kernel/msgmni
```

To increase the maximum number of message queues on your system, issue the following command:

```
sysctl -w kernel.msgmni=n
```

where *n* is the maximum number of message queues (MSGMNI) you want to be allowed by the system.

See the *Administrator's Reference* for more information.

Database Reorganization

Periodically reorganizing the server's database can improve performance. A new command has been added that allows administrators to determine the estimated space saved by reorganization of the database. The process can be queried to monitor the command's progress and, when finished, the results can be viewed by querying the database in detailed format.

See Chapter 19, "Managing the Database and Recovery Log," on page 489 for more information.

Disk-only Backup

Disk-only backup has been enhanced in order to take advantage of the inexpensive disk storage currently available in the market. These improvements to sequential-access FILE device type and random-access DISK device class storage include:

- The ability to create large, sequential-access FILE-type storage pools using a single FILE device-class definition that specifies two or more directories.
- The ability to create and format FILE- or DISK device-type volumes in a single step.
- The ability to use enhanced space trigger functionality to automatically allocate space for private volumes in sequential-access FILE device type and random-access DISK device class storage pools. This will reduce the potential for disk fragmentation and maintenance overhead.

See the following for more information:

- Chapter 3, "Introducing Storage Devices," on page 61
- Chapter 4, "Using Magnetic Disk Devices," on page 83
- Chapter 6, "Configuring Storage Devices," on page 99
- Chapter 9, "Defining Device Classes," on page 197
- Chapter 10, "Managing Storage Pools and Volumes," on page 217
- Chapter 19, "Managing the Database and Recovery Log," on page 489

Enhancements for Server Migration and Reclamation Processes

Administrators can now control and schedule routine server operations by scheduling the migration or reclamation command to run during convenient server activity times. The number of processes for migration and reclamation is also enhanced to allow multiple processes for the operations. These new features allow for better utilization of available tape drives and FILE volumes.

See Chapter 10, "Managing Storage Pools and Volumes," on page 217 for more information.

IBM 3592 WORM Support

Tivoli Storage Manager now supports the 3592 WORM tape device.

See Chapter 9, "Defining Device Classes," on page 197 and Chapter 8, "Managing Removable Media Operations," on page 167 for more information.

Improved Defaults

Defaults for parameters on some commands and server options have been improved to better match conditions in current user environments. If you have not specified these options in the options file, new defaults will take effect immediately. If you have specified some of these options, the value you have already specified will be used and the default changes will have no effect.

See Chapter 9, "Defining Device Classes," on page 197, Chapter 12, "Managing Client Nodes," on page 317, and the *Administrator's Reference* for more information.

LAN-free Environment Configuration

Enhancements have been made to allow you to quickly determine if your LAN-free environment has been configured correctly. You can request validation for one client node and one storage agent. When validation is requested, a detailed report is generated explaining why the storage pool is or is not LAN-free capable. As a result, you can determine if there is a setting or configuration issue on the server preventing LAN-free data movement.

See "Validating your LAN-free Configuration" on page 136 for more information.

NDMP Operations

NDMP operations for backup of NAS file servers have been enhanced to support the following functions:

- Directory-level backup of NAS data, which enables the division of a file system backup operation among several NDMP backup operations as well as several tape drives. This enhancement will reduce backup and restore times.
- NDMP Directory Level Backup will enable Tivoli Storage Manager to backup user created snapshots that are stored as sub-directories, specifically Network Appliance snapshots.

See Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143 for more information.

New Interface to Manage Servers: Administration Center

The Administration Center is a Web-based interface that can be used to centrally configure and manage IBM Tivoli Storage Manager Version 5.3 servers. This new interface replaces the administrative Web interface. The Administration Center is installed as an IBM Integrated Solutions Console component. The Integrated Solutions Console allows you to install components provided by multiple IBM applications, and access them from a single interface. See the Installation Guide for installation information. See "Managing Servers with the New Administration Center" on page 36 for more information.

SCSI Library Enhancements

You can now define a 3592 drive to an SCSI library.

See the *Administrator's Reference* for more information.

Server Processing Control in Scripts

Administrators can automate routine server operations. In server scripts, commands can now be processed in serial and parallel. This enables multiple processes to run concurrently, which reduces data transfer time.

See Chapter 18, "Automating Server Operations," on page 471 for more information.

Simultaneous Write Inheritance Improvements

The simultaneous write function lets you write data simultaneously to primary storage pools and up to three copy storage pools, increasing your level of data protection and reducing the time required for storage pool backup. In version 5.3, when a data storage operation switches from a primary storage pool to the next primary storage pool in the storage pool hierarchy, the next storage pool inherits the list of copy storage pools and the value of the COPYCONTINUE parameter from the primary storage pool. One benefit of this design is that it reduces the likelihood that mount points have to be released and reacquired when the server switches from one primary storage pool to another.

See Chapter 10, "Managing Storage Pools and Volumes," on page 217 for more information.

Space Triggers for Mirrored Volumes

Space triggers for mirrored volumes have been enhanced, allowing administers to specify an expansion prefix for new mirrored database and log volumes.

See Chapter 23, "Protecting and Recovering Your Server," on page 639 for more information.

Support for Multiple Tivoli Storage Manager Client Nodes

Backup from multiple nodes to be stored as a single target node in the server database is now allowed. With the consolidation of data under a single target node on the server, directories and files can be easily found when restore operations are necessary. It is also no longer necessary for physical machines to share password files, because password management and Tivoli Storage Manager server authentication is now handled by clients (independent of relationships with other nodes).

See Chapter 12, "Managing Client Nodes," on page 317 for more information.

Tivoli Storage Manager Scheduling Flexibility

Tivoli Storage Manager schedule capability now allows for more useful calendar-type administrative and client scheduling. There is now the flexibility to schedule most repetitive items and even some holidays, such as:

- run on the last Friday of every month
- run on the first Sunday of every quarter
- run on every day in the last week of year

See Chapter 16, "Managing Schedules for Client Nodes," on page 441 and Chapter 18, "Automating Server Operations," on page 471 for more information.

Technical Changes for Version 5 Release 2 — January 2004

The following changes have been made to the product for this edition:

NDMP Support

Tivoli Storage Manager now supports backup using NDMP for NAS file servers that comply with NDMP standards and are neither EMC Celerra nor Network Appliance file servers. NAS vendors can now use a certification process in order to ensure NAS file servers are compatible with Tivoli Storage Manager.

See Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143 for more information.

Disaster Recovery Manager

With this function you can prepare a disaster recovery plan to help you recover your Tivoli Storage Manager servers, clients, and data in the case of a disaster. You can also use the function to manage your offsite recovery media and automate some recovery procedures.

See Chapter 24, "Using Disaster Recovery Manager," on page 687 for more information.

IBM Tivoli Storage Manager for Data Retention

IBM Tivoli Storage Manager for Data Retention provides enhancements to Tivoli Storage Manager Extended Edition to prevent critical data from being erased or rewritten. This new product helps meet additional requirements defined by regulatory agencies for retention and disposition of data. These enhancements include new functionality and new device support and are delivered in these key areas:

Data retention protection

This feature prevents deliberate or accidental deletion of data until its specified retention criterion is met.

See "Data Retention Protection" on page 393 for more information.

Event-based retention policy

In some cases, retention must be based on an external event such as closing a brokerage account. IBM Tivoli Storage Manager for Data Retention supports event-based retention policy to allow data retention to be based on an event other than the storage of the data.

See "Defining and Updating an Archive Copy Group" on page 387 for more information.

Deletion hold

In order to ensure that records are not deleted when a regulatory retention period has lapsed but other legal requirements mandate that the records continue to be maintained, IBM Tivoli Storage Manager for Data Retention includes deletion hold. Using this feature will prevent stored data from being deleted until the hold is released.

See "Deletion Hold" on page 394 for more information.

Operational Reporting

The Operational Reporting feature automates daily monitoring tasks and generates reports indicating whether a server requires attention. This feature reduces the amount of time required to administer Tivoli Storage Manager.

See "Using IBM Tivoli Storage Manager Operational Reporting" on page 543 for more information.

IBM 3592 Support

Tivoli Storage Manager now supports IBM 3592 devices. The 3592 device has fast access capability that gives users the option of scaling down tape capacities in order to get improved data access response times.

See "Scale Capacity" on page 202 and "Setting Up a Storage Pool Hierarchy" on page 233 for more information.

Sony AIT50 and AIT100 WORM Media Support

Tivoli Storage Manager now supports the Sony AIT50 and AIT100 WORM (write once, read many) media technology. To enable this support, the DEFINE DEVCLASS (8MM) command has a new parameter. For more information about WORM media, including Sony AIT50 and AIT100, see "Special Considerations for WORM Tape Media" on page 174.

Technical Changes for Version 5 Release 2 — June 2003

The following changes have been made to the product for this edition:

Security: Firewall Support

Tivoli Storage Manager has enhanced support for environments with firewalls in which communication originating from outside the firewall is to be restricted. Clients normally contact the server, but with the new firewall support, you can choose to restrict session initiation to the server. Scheduled, backup-archive client operations can be restricted to server-initiated sessions.

See Chapter 11, "Adding Client Nodes," on page 309 and the *Quick Start* for more information.

Support for SCSI Libraries with Multiple Drive Types

Tivoli Storage Manager now supports libraries that are configured with more than a single drive and media type. Partitioning the library to segregate the device types is not required, but each device type requires a separate device class and storage pool. This is limited to certain models which are denoted as such in our supported devices web page.

See "Mixing Device Types in Libraries" on page 100 for more information.

Linux for pSeries®

The Tivoli Storage Manager server for Linux is now supported on pSeries hardware.

See Chapter 5, "Attaching Devices to the Server System," on page 91 and *Quick Start* for more information.

Linux for zSeries®

The Tivoli Storage Manager server for Linux is now supported on zSeries hardware.

See Chapter 5, "Attaching Devices to the Server System," on page 91 and *Quick Start* for more information.

NDMP Operations

IBM 3494 Library Support

NDMP support to the library type IBM 3494 Tape Library DataServer is now provided.

See Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143 for more information.

File Level Restore

Tivoli Storage Manager currently provides backup and recovery support for network-attached storage (NAS) file servers and utilizes Network Data Management Protocol (NDMP) to communicate with and provide backup and recovery services for NAS file servers.

Support for file level restore includes enhancements to allow tracking of individual files within a backed-up file system image. This enhancement makes it possible to display the contents of an image backup, and individual files within that image can be selected for restore. Implementation is achieved by generating a table of contents (TOC) during backup which is stored on the server.

See Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143 for more information.

EMC Celerra NAS Device Support

Backup and restore operations for EMC Celerra file servers via NDMP is now supported. This support includes all base NDMP functions provided for Network Appliance file servers as well as the file-level restore function.

See Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143 and Chapter 10, "Managing Storage Pools and Volumes," on page 217 for more information.

Accurate SAN Device Mapping

Device IDs within a SAN environment change when a reset or other environmental changes occur. With accurate SAN device mapping, Tivoli Storage Manager can now detect SAN changes and report that a reconfiguration is required.

See "Recovering from Device Changes on the SAN" on page 142 for more information.

Macintosh OS X Unicode Support for Backup-Archive Client

Unicode file spaces are now supported on the Macintosh client. By supporting a Unicode-enabled client, the Tivoli Storage Manager server can store file spaces with Unicode file space names, directory names, and file names. The client can successfully process a 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 from the server.

See Chapter 12, "Managing Client Nodes," on page 317 for more information.

TapeAlert Device Support

TapeAlert is an application that provides detailed diagnostic information about tape and library device hardware errors. It captures the log page from the drive or library when tapes are dismounted and issues the appropriate ANR error messages, allowing you to recognize problems as early as possible.

See "Handling Tape Alert Messages" on page 195 for more information.

Increased Archive Retention Limits

Tivoli Storage Manager now supports increased retention times for archives and backup sets. These new retention values will allow data archives to be kept longer.

See the *Administrator's Reference* for more information.

Tape Autolabeling

Tivoli Storage Manager now provides the option to have tape volumes automatically labeled by the server. This option is available for SCSI library types. The server will label both blank and incorrectly labeled tapes when they are initially mounted. This eliminates the need to pre-label a set of tapes.

For more information, see the readme file.

StorageTek VolSafe Support

Tivoli Storage Manager now supports StorageTek's VolSafe media technology.

See "Defining Device Classes" or the *Administrator's Reference* for more information.

Server Performance Tuning

The maximum value of the server option TXNGROUPMAX has been increased. When transferring multiple small files, increasing the TXNGROUPMAX option can improve throughput for operations to tape. It is now possible to set the TXNGROUPMAX option for individual clients.

See the *Administrator's Reference* for more information.

Licensing Changes

The application client for the WebSphere® server is now licensed.

See Chapter 17, "Managing Server Operations," on page 457 for more information.

Product Packaging and Name Changes

The following table lists changes to product packaging and names for IBM Tivoli Storage Manager. See www.ibm.com/software/tivoli/solutions/storage/ for complete details.

Table 1. Product Packaging and Name Changes

Former name	Current name or term	Notes
Tivoli Disaster Recovery Manager	disaster recovery manager (DRM)	This product is now part of IBM Tivoli Storage Manager Extended Edition.
Tivoli Data Protection for NDMP	operations that use NDMP	This product is now part of IBM Tivoli Storage Manager Extended Edition.
Tivoli Storage Manager Managed System for SAN	IBM Tivoli Storage Manager for Storage Area Networks	This product includes LAN-free data movement and library sharing on SANs. Tivoli SANergy® is a separate product, licensed to users through this product.
Tivoli Space Manager	IBM Tivoli Storage Manager for Space Management	The client is called *space manager* or *HSM client*.

Table 1. Product Packaging and Name Changes (continued)

Former name	Current name or term	Notes
Tivoli Data Protection products	One of the following: IBM Tivoli Storage Manager for Application Servers IBM Tivoli Storage Manager for Databases IBM Tivoli Storage Manager for Enterprise Resource Planning IBM Tivoli Storage Manager for Hardware IBM Tivoli Storage Manager for Mail	See the Web site for details. The clients are frequently called *application clients* in the product information.

Part 1. IBM Tivoli Storage Manager Basics

© Copyright IBM Corp. 1993, 2005

Chapter 1. Introducing IBM Tivoli Storage Manager

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

Server

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

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

Administrative interface
The administrative interface allows administrators to control and monitor server activities, define management policies for clients, and set up schedules to provide services to clients at regular intervals. Administrative interfaces available include a command-line administrative client and a Web browser interface called the Administration Center. Tivoli Storage Manager allows you to manage and control multiple servers from a single interface that runs in a Web browser.

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

Server storage
The server can write data to hard disk drives, disk arrays and subsystems, stand-alone tape drives, tape libraries, and other forms of random- and sequential-access storage. The media that the server uses are grouped into *storage pools*. The storage devices can be connected directly to the server, or connected via local area network (LAN) or storage area network (SAN).

Client Nodes
A client node can be a workstation, a personal computer, a file server, a network-attached storage (NAS) file server, or even another Tivoli Storage Manager server. The client node has IBM Tivoli Storage Manager client software installed (except for NAS file servers using NDMP). A client node is registered with the server.

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

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

© Copyright IBM Corp. 1993, 2005

Network-attached storage file server (using NDMP)
The server can use the Network Data Management Protocol (NDMP) to back up and restore file systems stored on a network-attached storage (NAS) file server. The data on the NAS file server is backed up to a tape library. No Tivoli Storage Manager software needs to be installed on the NAS file server. See Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143 for more information, including supported NAS file servers.

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

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

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

Also available is Tivoli Storage Manager for Hardware, which works with the backup-archive client and the API to help eliminate backup-related performance effects.

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

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

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

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

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

For information about supported operating systems for clients, see the IBM Tivoli Storage Manager Web site at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html

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

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

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

Concepts:
"How IBM Tivoli Storage Manager Stores Client Data"
"How the Server Manages Storage" on page 15
Chapter 2, "Getting Started with Tivoli Storage Manager," on page 19

How IBM Tivoli Storage Manager Stores Client Data

Tivoli Storage Manager policies are rules that determine how the client data is stored and managed. The rules include where the data is initially stored, how many backup versions are kept, how long archive copies are kept, and so on. You can have multiple policies and assign the different policies as needed to specific clients, or even to specific files.

Policy assigns a location in server storage where data is initially stored. Server storage is divided into storage pools that are groups of storage volumes. Server storage can include hard 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 "The Standard Policy" on page 359. You can modify this policy and define additional policies.

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

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

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

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

Instant archive and rapid recovery

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

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

Migration and recall

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

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

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

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

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

The steps in the process are as follows:

1 A client initiates a backup, archive, or migration operation. The file involved in the operation is bound to a management class. The management class is either the default or one specified for the file in client options (the client's include-exclude list).

2 If the file is a candidate for backup, archive, or migration based on information in the management class, the client sends the file and file information to the server.

3 The server checks the management class that is bound to the file to determine the *destination*, the name of the Tivoli Storage Manager storage pool where the server initially stores the file. For backed-up and archived files, destinations are assigned in the backup and archive copy groups, which are within management classes. For space-managed files, destinations are assigned in the management class itself.

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

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

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

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

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

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

For information on assigning storage destinations in copy groups and management classes, and on binding management classes to client files, see Chapter 13, "Implementing Policies for Client Data," on page 357.

For information on managing the database, see Chapter 19, "Managing the Database and Recovery Log," on page 489.

For information about storage pools and storage pool volumes, see Chapter 10, "Managing Storage Pools and Volumes," on page 217.

For information about event-based policy, deletion hold, and data retention protection, see Chapter 13, "Implementing Policies for Client Data," on page 357.

Options for Data Protection

IBM Tivoli Storage Manager provides a variety of backup and archive operations, allowing you to select the right protection for the situation. Table 2 shows some examples of the protection options.

Table 2. Examples of Meeting Your Goals with IBM Tivoli Storage Manager

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

Table 2. Examples of Meeting Your Goals with IBM Tivoli Storage Manager (continued)

For this goal...	Do this...
Create an archive for a backup-archive client, from data that is already stored for backup.	Use the backup-archive client to perform incremental backups, and then generate a backup set by using the Tivoli Storage Manager server. This is also called *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 404 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 412 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.
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 Tivoli Storage Manager for Hardware product, such as IBM System Storage Archive Manager for IBM Enterprise Storage Server® for DB2.
Make backups transparent to end users.	Use the backup-archive client with centrally scheduled backups that run during off-shift hours. Monitor the schedule results.

Table 2. Examples of Meeting Your Goals with IBM Tivoli Storage Manager (continued)

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

Schedule the backups of client data to help enforce the data management policy that you establish. If you schedule the backups, rather than rely on the clients to perform the backups, the policy that you establish is followed more consistently. See Chapter 15, "Scheduling Operations for Client Nodes," on page 433.

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 Backup Compared with Other Backup Types" on page 14.

Table 3 on page 11 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 3. 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 372 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 374 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 418 and the *Backup-Archive Clients Installation and User's Guide*.

Table 3. Summary of Client Operations (continued)

Type of operation	Description	Usage	Restore options	For more information
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 Windows NT® and Windows 2000 systems.	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*.
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 400 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 7, "Using NDMP for Operations with NAS File Servers," on page 143.
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 Manager for Hardware 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 IBM Tivoli Storage Manager for hardware components.
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 3. 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 375 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 412.

Progressive Incremental Backup Compared with Other Backup Types

IBM Tivoli Storage Manager has a unique, efficient method for its standard backups, as well as a number of other methods that are summarized in Table 3 on page 11. The standard method that Tivoli Storage Manager uses is *progressive incremental backup*.

The terms *differential* and *incremental* are often used to describe backups. The terms usually 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.

Additional Protection: Storage Pool and Server Database Backups

Built into the server are additional levels of protection for client data:

- You can back up storage pools. The data is backed up to copy storage pools, which the server can automatically access if needed to retrieve a file. See "Storage Pool Protection: An Overview" on page 640.
- You can back up the server's database. The database is key to the server's ability to track client data in server storage. See "Database and Recovery Log Protection: An Overview" on page 641.

These backups can become part of a disaster recovery plan, created automatically by disaster recovery manager. See Chapter 24, "Using Disaster Recovery Manager," on page 687.

How Data Moves to Server Storage

The Tivoli Storage Manager client traditionally sends its data to the server over the LAN. The server then transfers the data to a device that is attached to the server. With the advent of SAN and network-attached storage, however, Tivoli Storage Manager offers options that enable you to minimize use of the LAN and the use of the computing resources of both the client and the server.

LAN-free data movement allows storage agents that are installed on client nodes to move data without sending the data over the LAN to the server. See “LAN-Free Data Movement” on page 73.

For network-attached storage, use NDMP operations to avoid data movement over the LAN. See “NDMP Backup Operations” on page 75.

Consolidating Backed-up Data for Clients

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 or by file space per client. Your selection depends on the size of the file spaces being stored and the restore requirements.

See “Keeping a Client’s Files Together: Collocation” on page 258.

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 412.

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 for a Client Node” on page 298.

How the Server Manages Storage

Through the server, you manage its storage — 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.

IBM Tivoli Storage Manager Device Support

Tivoli Storage Manager supports the use of a variety of devices for server storage. Tivoli Storage Manager can use direct-attached storage as well as network-attached storage. See the current list on the IBM Tivoli Storage Manager Web site at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html

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

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

Drive
Each drive represents a drive mechanism in a tape device.

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

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

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

Storage pools and volumes
A storage pool is a named collection of volumes that have the same media type. A storage pool is associated with a device class. For example, an LTO tape storage pool contains only LTO tape volumes. A storage pool volume is associated with a specific storage pool.

For details about device concepts, see Chapter 3, “Introducing Storage Devices,” on page 61.

Migrating Data 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 “Overview: The Storage Pool Hierarchy” on page 232.

Removing Expired Data

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 391.

Reclaiming Media for Reuse

As server policies automatically expire data, the media where the data is stored accumulates unused space. Other products might require you to implement a tape rotation scheme that allows you to reuse the media only when all data on the media has expired. The Tivoli Storage Manager server implements a different process, called reclamation, that 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. Getting Started with Tivoli Storage Manager

The server comes with many defaults set so that clients can begin using its services immediately. The amount and importance of the data protected by Tivoli Storage Manager, your business process requirements, and other factors make it likely that you need to adjust and customize the server's behavior. Your changing storage needs and client requirements can mean on-going configuration changes and monitoring.

The server's capabilities are extensively described in this guide. To get an introduction to the tasks available to an administrator of Tivoli Storage Manager, read the following sections:

Administrative Tasks:
"Interfaces to IBM Tivoli Storage Manager"
"Configuring and Managing Server Storage" on page 20
"Managing Client Operations" on page 24
"Maintaining the Server" on page 31
"Protecting the Server" on page 35
"Managing Servers with the New Administration Center" on page 36

Interfaces to IBM Tivoli Storage Manager

Tivoli Storage Manager has the following types of interfaces:

- 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 Administration Center allows you to access Tivoli Storage Manager server functions from any workstation using a supported Web browser. The interface also allows Web access to the command line. See the "Managing Servers with the New Administration Center" on page 36 for more information.

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

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

  For more information, see the *IBM Tivoli Storage Manager Using the Application Program Interface*.
- Access to information in the server's database via standard SQL SELECT statements. For reporting purposes, the Tivoli Storage Manager product also

© Copyright IBM Corp. 1993, 2005

provides an ODBC driver. The driver allows you to use a spreadsheet or database program to query the server database for information. See "Using SQL to Query the IBM Tivoli Storage Manager Database" on page 518.

Customizing the Server with the Server Options File

Server options allow you to customize the server and its operations. Some examples of what these options affect are:

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

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

Configuring and Managing Server Storage

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

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

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

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

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

For an introduction to key storage concepts, see Chapter 3, "Introducing Storage Devices," on page 61.

Using Hard Disk Devices with IBM Tivoli Storage Manager

Hard disk devices can be used with Tivoli Storage Manager for two purposes:

- Storage of the database and recovery log

- Storage of client data that is backed up, archived, or migrated from client nodes

The server can store data on hard 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 217.

Using Removable Media Devices with IBM Tivoli Storage Manager

Removable media devices can be used with Tivoli Storage Manager for the following purposes:

- Storage of client data that is backed up, archived, or migrated from client nodes
- Storage of database backups
- The exporting of data, to move the data to another server

Attaching and Configuring Devices

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

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

See Chapter 5, “Attaching Devices to the Server System,” on page 91 for more information.

Managing Removable Media Operations

Routine removable media operations including the following:

- Preparing media for use

- Controlling media reuse
- Ensuring that sufficient media are available
- Mounting volumes in response to server requests, for manually operated drives
- Managing libraries and drives

For information about removable media operations, see Chapter 8, "Managing Removable Media Operations," on page 167.

Using Tape Volumes

All tape volumes must be labeled before they can be used. If you want a volume to be used only when it is requested by name, you must also define it to IBM Tivoli Storage Manager. 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*. A storage pool can request available scratch volumes up to the number specified for that storage pool.

Preparing Tape Volumes with a Manual Drive

If you are using only one tape drive, consider labeling several tapes at this time.

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

To label tapes with an automated library, remove any tapes that you do not want to use with IBM Tivoli Storage Manager, and load the tapes to be labeled.

1. 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:

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

2. Define any volumes that are to be used as private volumes:

   ```
   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. Migrating to tape from a disk storage pool allows you to verify that tape devices are properly set up. Migration requires tape mounts. The mount messages are directed to the console message queue and to any administrative client that has been started with either the mount mode or console mode option. To have the server migrate data from BACKUPPOOL to AUTOPOOL and from ARCHIVEPOOL to TAPEPOOL do the following:

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

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

```
migrate stgpool backuppool
```

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

See "Migration for Disk Storage Pools" on page 239 and the *Administrator's Reference* for more information.

Managing Storage Pools and Volumes

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

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

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

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

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

Storage pool backup
The data on primary storage pools can be backed up to copy storage pools for disaster recovery purposes. Backup to copy storage pools can occur simultaneously as client data is written to the primary storage pool.

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

You 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.

Increasing Disk Storage Pool Sizes

At server installation, these storage pool volumes are created:

Note that these volumes may not be adequate for your storage needs. You should consider how much data clients will be storing. Disk storage pool volumes should be at least large enough to hold one day's worth of client backups. As you add clients, you may need to increase your disk storage pool size. You can format and define additional storage pool volumes. For example, to create a 100MB volume named *sbkup01.dsm* in BACKUPPOOL, do the following:

```
define volume backuppool sbkup01.dsm formatsize=100
```

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 217.

Managing 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

Managing Client Nodes

A very basic administrative task is adding client nodes, giving the systems that the nodes represent access to the services and resources of the Tivoli Storage Manager server. The Tivoli Storage Manager server supports a variety of client nodes. You can register the following types of clients and servers as client nodes:

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

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

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

Adding Administrators

If you have installed any additional administrative clients, you should register them and grant an authority level to each. For details about authority levels, see Chapter 12, “Managing Client Nodes,” on page 317.

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

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

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

See “Managing IBM Tivoli Storage Manager Administrators” on page 351 for more information.

Adding Backup-Archive Clients

If you have installed additional backup-archive clients you should register them. For example, register a node named MERCEDES with the password MONTANA:

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

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

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 309.

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 411.

Allowing Subfile Backups

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

For more information on managing client nodes, see Chapter 11, “Adding Client Nodes,” on page 309, Chapter 12, “Managing Client Nodes,” on page 317, Chapter 14, “Managing Data for Client Nodes,” on page 411 and the *Backup-Archive Clients Installation and User’s Guide*.

Managing Security

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 310.

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

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

See "Managing Passwords and Login Procedures" on page 354.

You can control the authority of administrators. An organization may name a single administrator or may distribute the workload among a number of administrators and grant them different levels of authority. For details, see "Managing Levels of Administrative Authority" on page 353.

For better security when clients connect across a firewall, you can control whether clients can initiate contact with the server for scheduled operations. See the "Managing Client Nodes across a Firewall" on page 318 for details.

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

For additional ways to manage security, see "Managing IBM Tivoli Storage Manager Security" on page 348.

Implementing Policies for Client Data

As the administrator, you define the rules for client backup, archive, and migration operations, based on user or business requirements. The rules are called *policies*. Policies identify:

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

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

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

Creating a New Policy

The Tivoli Storage Manager default policy specifies how Tivoli Storage Manager manages client files. For example, it specifies that Tivoli Storage Manager retains up to two backup versions of any file that exists on the client (see Chapter 13,

"Implementing Policies for Client Data," on page 357 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.

Attention: 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. Note, however, that if you store directly to tape, the number of available tape drives limits the number of client nodes that can store data at the same time.

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

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

   ```
   copy domain standard newdomain
   ```

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

   For our 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
   ```

   Note: 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**, do either of the following:

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

Tivoli Storage Manager will now keep up to ten versions of backed up files for client node **mercedes** because it is assigned to the NEWDOMAIN policy domain.

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

For more information about the default policy and establishing and managing new policies, see Chapter 13, "Implementing Policies for Client Data," on page 357.

Scheduling 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 340 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.

Creating a Client Include-Exclude List

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 317 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 368.

Automating Client Operations

You can schedule most client operations, such as backup, to begin automatically. You can schedule the following on most clients:

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

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

1. Schedule an incremental backup and associate the schedule with the clients. The schedule, named DAILY_INCR, is for the Tivoli Storage Manager default policy domain, named STANDARD. The default specifies backup to the disk storage pool BACKUPPOOL. This schedule calls for a schedule window that:
   - Begins on the date the schedule is defined (the default) at 11:00 p.m.
   - Lasts for 1 hour (the default)
   - Is repeated daily (the default)
   - Stays in effect indefinitely (the default)

   ```
   define schedule standard daily_incr action=incremental -
     starttime=23:00

   define association standard daily_incr bill,mark,mercedes
   ```

2. Start the client scheduler. For the schedules to become active for a workstation, a user must start the scheduler from the node.

   ```
   dsmc schedule
   ```

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

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

3. Because the DAILY_INCR schedule is to run daily, you can verify that it is working as it should on the day after you define the schedule and associate it with clients. If the schedule has run successfully, the status will be *Completed*.

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

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

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

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

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

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 360.
- For automating client operations, see Chapter 15, "Scheduling Operations for Client Nodes," on page 433.
- 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 357.

Maintaining the Server

To keep the server running well, you have access to 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

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.

Managing Server Operations

There are a variety of tasks associated with managing server operations:

- 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 457. For suggestions about the day-to-day tasks required to administer the server, see Chapter 17, "Managing Server Operations," on page 457.

Automating Server Operations

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

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

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

Automating Administrative Commands

You can automate Tivoli Storage Manager administrative tasks by scheduling administrative commands. For example, you can automate commands that need to be run regularly or that require significant processing or other system resources. In this section, you define a schedule to run expiration processing.

1. Define a schedule named EXPPROC that invokes expiration processing every night at 2:00 a.m.

   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)

   ```
   define schedule expproc type=administrative -
     cmd='expire inventory' active=yes starttime=02:00
   ```

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. If the schedule has run successfully, the status will be *Completed*.

   ```
   query event expproc type=administrative begindate=today-1
   ```

For more information about automating Tivoli Storage Manager operations, see Chapter 18, "Automating Server Operations," on page 471.

Managing the Database and Recovery Log

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

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 start-up operations.

You can tune database and recovery log performance automatically or manually. You can set up triggers so that additional space is automatically added to the database and recovery log as needed.

Increasing Database and Recovery Log Sizes

The first-time installation procedure creates a default 17MB database volume (*db.dsm*) and a default 9MB recovery log volume (*log.dsm*). Databases and recovery logs are allocated in 4MB increments with an additional 1MB for overhead.

The database size is largely determined by the number of client files being stored on server storage. As you add clients, you may need to increase your database size. The recovery log size may need to be increased depending on the current utilization.

To increase the size of the database or recovery log, do the following:

Format and define a 500MB database volume and a 25MB recovery log volume and extend the database and recovery log, by entering:

```
define dbvolume db2.dsm formatsize=500
define logvolume log2.dsm formatsize=25

extend db 500
extend log 25
```

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 489.

Monitoring the IBM Tivoli Storage Manager Server

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

The Administration Center includes a health monitor, which presents a view of the overall status of multiple servers and their storage devices. From the health monitor, you can link to details for a server, including a summary of the results of client schedules and a summary of the availability of storage devices. See “Managing Servers with the New Administration Center” on page 36.

You can use Tivoli Storage Manager queries and SQL queries to get information about the server. An ODBC interface is available.

You can set up automatic logging of information about Tivoli Storage Manager clients and server events.

The Operational Reporting feature automates your daily monitoring tasks and generates reports that can tell you if a server requires attention. It reduces the amount of time and effort to administer Tivoli Storage Manager.

See the following sections for more information about these tasks.

- Chapter 20, “Monitoring the IBM Tivoli Storage Manager Server,” on page 513
- “Using SQL to Query the IBM Tivoli Storage Manager Database” on page 518
- “Logging IBM Tivoli Storage Manager Events to Receivers” on page 526
- “Daily Monitoring Scenario” on page 543
- “Using IBM Tivoli Storage Manager Operational Reporting” on page 543

Working with a Network of IBM Tivoli Storage Manager Servers

You may have a number of Tivoli Storage Manager servers in your network, at the same or different locations. Some examples of different configurations are:

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

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

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

- Designate one server as a configuration manager that distributes configuration information such as policy to other servers. See "Setting Up an Enterprise Configuration" on page 575.
- Route commands to multiple servers while logged on to one server. See "Routing Commands" on page 597.
- 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 538.
- 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 601 for details.
- Share an automated library among Tivoli Storage Manager servers. See "Devices on a Storage Area Network" on page 71.
- 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 24, "Using Disaster Recovery Manager," on page 687 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 601 for details.

Exporting and Importing Data

As conditions change, you can move data from one server to another by using export and import processes. For example, you may need to balance workload among servers by moving client nodes from one server to another. The following methods are available:

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

For more information about moving data between servers, see Chapter 22, "Exporting and Importing Data," on page 609.

Protecting IBM Tivoli Storage Manager and Client Data

The database, recovery log, and storage pools are critical to the operation of the server. If the database or recovery log is unusable, the entire server is unavailable. If a database is lost and cannot be recovered, the backup, archive, and space-managed data for that server is lost. If a storage pool volume is lost and cannot be recovered, the data on the volume is also lost.

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. See Chapter 24, "Using Disaster Recovery Manager," on page 687 for details.

Protecting the Server

Because the server is protecting client data, it is important to protect the server itself.

Tivoli Storage Manager provides a number of ways to protect and recover your server from media failure or from the loss of the Tivoli Storage Manager database or storage pools. Recovery is based on the following preventive measures:

- Mirroring, by which the server maintains one or more copies of the database or the recovery log, allowing the system to continue when one of the mirrored disks fails
- Periodic backup of the database
- Periodic backup of the storage pools
- Audit of storage pools for damaged files, and recovery of damaged files when necessary
- Backup of the device configuration and volume history files
- Validation of the data in storage pools, using cyclic redundancy checking

For information about protecting the server with these measures, see Chapter 23, "Protecting and Recovering Your Server," on page 639.

You can also create a maintenance script to perform database and storage pool backups through the Server Maintenance work item in the Administration Center. See "Managing Servers with the New Administration Center" on page 36 for details.

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

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

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

For more information about protecting your server and for details about recovering from a disaster, see Chapter 23, “Protecting and Recovering Your Server,” on page 639.

Managing Servers with the New Administration Center

The Administration Center is a Web-based interface that can be used to centrally configure and manage IBM Tivoli Storage Manager Version 5.3 servers. The Administration Center provides wizards to help guide you through common configuration tasks. Properties notebooks allow you to modify settings and perform advanced management tasks.

The Administration Center is installed as an IBM Integrated Solutions Console component. The Integrated Solutions Console allows you to install components provided by multiple IBM applications, and access them from a single interface. For installation information, see *Installation Guide*.

Using the Administration Center

The Administration Center is installed as a component of the IBM Integrated Solutions Console. You can use the Administration Center to centrally configure and manage your IBM Tivoli Storage Manager environment.

Basic work items (for example, server maintenance, storage devices, and so on) are listed on the left side of the Integrated Solutions Console. When you click on a work item, a work page containing one or more portlets (for example, the Servers portlet) is displayed in the work area on the right side of the interface. You use portlets to perform individual tasks, such as creating storage pools.

Each time you click a work item, a new work page is opened. This allows you to open the same work item for more than one server. To navigate among open work items, use the page bar at the top of the work area.

Many portlets contain tables. These tables display objects like servers, policy domains, or reports. There are two ways to work with table objects. For any table object, you can do the following:

1. Click its radio button or check box in the **Select** column.
2. Click **Select Action** to display the table action list.
3. Select an action to perform that action.

For some table objects, you can also click the object name to open a portlet or work page pertaining to it. In most cases, a properties notebook portlet is opened. This provides a fast way to work with table objects. You can always open the same portlet or work page using the table action list.

Fields marked with an asterisk and highlighted in yellow require an entry or selection. However, if you have the Google search bar installed in your browser,

some fields can display bright yellow, whether they are required or not. To get help at any time, click the context sensitive help button in the title bar of a portlet, properties notebook, and so on.

If you want more space in the work area, you can hide the navigation tree by clicking

Do not use the **Back**, **Forward** and **Refresh** buttons in your browser. Doing so can cause unexpected results. Using your keyboard's **Enter** key can also cause unexpected results. Use the controls in the Administration Center interface instead.

The following simple task will help familiarize you with Administration Center controls. Suppose you want to create a new client node and add it to the STANDARD policy domain associated with a particular server.

1. If you have not already done so, access the Administration Center by entering the following address in a supported Web browser: http://*machine_name*:8421/ibm/console. The machine name is the network name or IP address of the machine on which you installed the Administration Center. To get started, log in using the Integrated Solutions Console user ID and password that you created during the installation.
2. If the work item list is not displayed, click **Work Items** on the Integrated Solutions Console. Click **Tivoli Storage Manager**, and then click **Policy Domains and Client Nodes** in the Work Items list. The Policy Domains work page is displayed with a table that lists the servers that are accessible from the Administration Center. The table also lists the policy domains defined for each server:

3. In the **Server Name** column of the **Policy Domains** table, click the name of the server with the STANDARD domain to which you want to add a client node. A portlet is displayed with a table that lists the policy domains created for that server:

ID_mseg Policy Domains

Policy domains help you to apply consistent rules for data management to groups of clien

--- Select Action ---

Select	Domain Name	Description	Backup Retention
○	JOANNE	test	30 days
○	MYPOOLDOMAIN	Installed default policy domain.	30 days
○	NAS5DOMAIN	Policy domain for jnnas5 NAS data mover.	30 days
○	STANDARD	Installed default policy domain.	30 days
○	TEST		30 days

Page 1 of 1 Total: 5 Filtered: 5 Displayed: 5 Selected: 0

4. In the **Domain Name** column of the server's **Policy Domains** table, click the STANDARD policy domain. The STANDARD Properties portlet is displayed:

STANDARD Properties

- Domain Properties
- Client Nodes
- Option Sets
- Schedules Nodes
- Management Classes

5. In the domain's properties portlet, click **Client Nodes**. A table is displayed listing all the nodes assigned to the STANDARD policy domain:

--- Select Action ---

Select	Name	Type	Platform
○	MSEG	Client	WinNT
○	NODE1	Client	(?)
○	NODE2	Client	(?)

Page 1 of 1 Total: 3 Filtered: 3 Displayed: 3 Selected: 0

6. In the client nodes table, click **Select Action**, and then select **Create a Node**. The Create Node wizard is displayed:

7. Follow the instructions in the wizard. After you complete the wizard, the name of the new client node is displayed in the **Client Node** table for the STANDARD policy domain.

Quick paths to creating Tivoli Storage Manager objects

Use this quick path reference to create and modify objects.

"Administrator" on page 40
"Backup set" on page 40
"Client node (including its file spaces)" on page 41
"Data mover" on page 41
"Database space trigger" on page 41
"Database volume" on page 42
"Device class" on page 42
"Disaster recovery plan" on page 42
"Drive" on page 42
"Library" on page 43
"Management class (including copy groups)" on page 43
"Option set" on page 43
"Path for a drive" on page 44
"Path for a library" on page 44
"Policy domain" on page 44
"The role of policy sets" on page 45
"Profile (for a configuration manager server)" on page 45
"Recovery log space trigger" on page 45
"Recovery log volume" on page 46
"Schedule for client nodes" on page 46

"Schedule for a server (administrative schedule)" on page 46
"Server (other Tivoli Storage Manager servers)" on page 47
"Server group" on page 47
"Server script" on page 47
"Storage pool" on page 48
"Subscription to a profile (for a managed server)" on page 48
"Volumes in a library" on page 48
"Volumes in a storage pool" on page 48

Note: The Add a Storage Device wizard helps you to create the objects that the server needs to use a new device, including a library, drives, paths, a device class, and storage pools.

Administrator

Task	Path
Create	1. Click **Enterprise Management** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **Server Properties**. 4. In the server's properties notebook, click the **Administrators** tab. 5. In the table, click **Select Action**, then select **Add Administrator**.
View and modify	1. Click **Enterprise Management** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **Server Properties**. 4. In the server's properties notebook, click the **Administrators** tab. 5. In the table, select an administrator. 6. Click **Select Action**, then select **Modify Administrator**.

Backup set

Task	Path
Create	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's policy domains table, click a name of a domain. 4. In the domain's properties portlet, click **Client Nodes**. 5. In the **Client Nodes** table, click the name of a node. 6. In the node's properties notebook, click the **Backup Sets** tab. 7. In the **Backup Sets** table, click **Select Action**, then select **Generate backup set**.
View and modify	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's policy domains table, click a name of a domain. 4. In the domain's properties portlet, click **Client Nodes** . 5. In the **Client Nodes** table, click the name of a node. 6. In the node's properties notebook, click the **Backup Sets** tab. 7. In the **Backup Sets** table, select a backup set. 8. Click **Select Action**, then select **Modify Backup set**.

Client node (including its file spaces)

Task	Path
Create	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's policy domains notebook, click a domain name. 4. In the domain's properties portlet, click **Client Nodes** . 5. In the table, click **Select Action**, then select **Create a Client Node**.
View and modify	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's policy domains notebook, click a domain name. 4. In the domain's properties portlet, click **Client Nodes**. 5. In the table, select a node name. 6. In the table, click **Select Action**, then select **Modify Client Node**.

Data mover

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **View Data Mover**. 4. In the server's data movers portlet, click **Select Action**, then select **Create NAS Data Mover**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **View Data Mover**. 4. In the server's data movers portlet, select a data mover. 5. click **Select Action**, then select **Modify Data Mover**.

Database space trigger

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click the name of a server. 3. In the server's properties notebook, click the **Database and Log** tab. 4. In the **Database** table, click **Select Action**, then select **Create Space Trigger**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties notebook, click the **Database and Log** tab. 4. In the **Database** table, select a database volume. 5. Click **Select Action**, then select **Modify Space Trigger**.

Database volume

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties portlet, click the **Database and Log** tab. 4. In the **Database** table, click **Select Action**, then select **Add Volume**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties portlet, click the **Database and Log** tab. 4. In the **Database** table, select a database volume. 5. Click **Select Action**, then select **Extend or Reduce**.

Device class

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **View Device Class**. 4. In the server's device classes portlet, click **Select Action**, then select **Create a Device Class**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **View Device Class**. 4. In the server's device classes portlet, select a device class. 5. Click **Select Action**, then select **Modify Device Class**.

Disaster recovery plan

Task	Path
Create	1. Click **Disaster Recovery Management** in the Work Items list. 2. In the **Servers** table, click a server name. 3. Click **Select Action**, then select **Create Recovery Plan**
View and modify	1. Click **Disaster Recovery Management** in the Work Items list. 2. In the **Servers** table, click a server name. 3. Click **Select Action**, then select **View Disaster Recovery Management Properties**

Drive

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click on the name of a library. 3. In the library's properties portlet, click the **Drives** tab. 4. In the table, click **Select Action**, then select **Add a Drive**.

Task	Path
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click on the name of a library. 3. In the library's properties portlet, click the **Drives** tab. 4. In the table, select a drive. 5. Click **Select Action**, then select **Modify Drive**.

Library

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click **Select Action**, then select **Create a Library**. 3. In the **Select a Server** table, select a server and click **Create a Library**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, select a library. 3. Click **Select Action**, then select **Modify Library**.

Management class (including copy groups)

Task	Path
Create	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click on the name of a server. 3. In the server's policy domains notebook, click on the name of a domain. 4. In the domain's properties portlet, click **Management Classes**. 5. In the table, click **Select Action**, then select **Create Management Class**.
View and modify	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click on the name of a server. 3. In the server's policy domains notebook, click on the name of a domain. 4. In the domain's properties portlet, click **Management Classes**. 5. In the table, select a management class. 6. Click **Select Action**, then select **Modify Management Class**.

Option set

Task	Path
Create	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click on the name of a server. 3. In the server's policy domains notebook, click on the name of a domain. 4. In the domain's properties portlet, click **Option Sets**. 5. In the table, click **Select Action**, then select **Create an Option Set**.

Task	Path
View and modify	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click on the name of a server. 3. In the server's policy domains notebook, click on the name of a domain. 4. In the domain's properties portlet, click **Option Sets**. 5. In the table, select an option set. 6. Click **Select Action**, then select **Modify Option Set**.

Path for a drive

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click a library name. 3. In the library's properties portlet, click the **Drive Paths** tab. 4. In the table, click **Select Action**, then select **Add Path**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click a library name. 3. In the library's properties portlet, click the **Drive Paths** tab. 4. In the table, select a drive path. 5. Click **Select Action**, then select **Modify Path**.

Path for a library

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click a library name. 3. In the library's properties portlet, click the **Library Paths** tab. 4. In the table, click **Select Action**, then select **Add Path**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click a library name. 3. In the library's properties portlet, click the **Library Paths** tab. 4. In the table, select a library path. 5. Click **Select Action**, then select **Modify Path**.

Policy domain

Task	Path
Create	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click on the name of a server. 3. In the server's policy domains notebook, click **Select Action**, then select **Create a Policy Domain**.

Task	Path
View and modify	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click on the name of a server. 3. In the server's policy domains notebook, select the name of a domain. 4. Click **Select Action**, then select **Modify Policy Domain**.

The role of policy sets

Experienced administrators of Tivoli Storage Manager are probably familiar with policy sets, which contain the management classes within a policy domain. The Administration Center eliminates the need to validate or activate a policy set when you make changes to management classes.

In the Administration Center, you work only with the active policy set. Any changes that you make to management classes are immediately activated, without additional effort on your part.

Profile (for a configuration manager server)

Task	Path
Create	1. Click **Enterprise Management** in the Work Items list. 2. In the **Servers** table, click the name of a server that is a configuration manager. 3. In the server's profiles table, click **Select Action**, then select **Create Profile**.
View and modify	1. Click **Enterprise Management** in the Work Items list. 2. In the **Servers** table, click on the name of a server that is a configuration manager. 3. In the server's profiles table, select a profile. 4. Click **Select Action**, then select **Modify Profile**.

Recovery log space trigger

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click the name of a server. 3. In the server's properties notebook, click the **Database and Log** tab. 4. In the **Recovery Log** table, click **Select Action**, then select **Create Space Trigger**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click the name of a server. 3. In the server's properties portlet, click the **Database and Log** tab. 4. In the **Recovery Log** table, select a recovery log. 5. Click **Select Action**, then select **Modify Space Trigger**.

Recovery log volume

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties portlet, click the **Database and Log** tab. 4. In the **Recovery Log** table, click **Select Action**, then select **Add Volume**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties portlet, click the **Database and Log** tab. 4. In the **Recovery Log** table, select a recovery log volume. 5. Click **Select Action**, then select **Extend or Reduce**.

Schedule for client nodes

Task	Path
Create	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click on the name of a server. 3. In the server's policy domains notebook, click on the name of a domain. 4. In the domain's properties portlet, click **Node Schedules**. 5. In the table, click **Select Action**, then select **Create a Schedule**.
View and modify	1. Click **Policy Domains and Client Nodes** in the Work Items list. 2. In the **Servers** table, click on the name of a server. 3. In the server's policy domains notebook, click on the name of a domain. 4. In the domain's properties portlet, click **Node Schedules**. 5. In the table, select a schedule. 6. Click **Select Action**, then select **Create a Schedule**.

Schedule for a server (administrative schedule)

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties portlet, click the **Administrative Schedules** tab. 4. In the table, click **Select Action**, then select **Create a Schedule**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties portlet, click the **Administrative Schedules** tab. 4. In the table, select a script. 5. Click **Select Action**, then select **Modify Schedule**.

Server (other Tivoli Storage Manager servers)

Task	Path
Create	1. Click **Enterprise Management** in the Work Items list. 2. In the table, click **Select Action**, then select **Add Server Connection**.
View and modify	1. Click **Enterprise Management** in the Work Items list. 2. In the table, select a server. 3. Click **Select Action**, then select **Modify Server Connection**.

Server group

Task	Path
Create	1. Click **Enterprise Management** in the Work Items list. 2. In the **Servers** table, click the name of a server. 3. In the servers and server groups portlet, click **Server Groups**. 4. In the **Server Groups** table, click **Select Action**, then select **Create a Server Group**.
View and modify	1. Click **Enterprise Management** in the Work Items list. 2. In the **Servers** table, click the name of a server. 3. In the servers and server groups portlet, click **Server Groups** . 4. In the **Server Groups** table, select a server group. 5. Click **Select Action**, then select **Modify a Server Group**.

Server script

Task	Path
Create	To create a script: 1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties portlet, click the **Scripts** tab. 4. In the table, click **Select Action**, then select **Create Script**. To get assistance in creating a script that performs key maintenance tasks: 1. Click **Server Maintenance** in the Work Items list. 2. In the table, click **Select Action**, then select **Create a Maintenance Script**.
View and modify	To view and modify a script: 1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, click a server name. 3. In the server's properties portlet, click the **Scripts** tab. 4. In the table, select a script. 5. Click **Select Action**, then select **Modify Script**. To modify a maintenance script: 1. Click **Server Maintenance** in the Work Items list. 2. In the table, select a maintenance script. 3. Click **Select Action**, then select **Modify Maintenance Script**.

Storage pool

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the Servers table, select a server. 3. Click **Select Action**, then select **View Storage Pools**. 4. In the server's storage pools portlet, click **Select Action**, then select **Create a Storage Pool**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **View Storage Pools**. 4. In the server's storage pools portlet, select a storage pool. 5. Click **Select Action**, then select **Modify Storage Pool**.

Subscription to a profile (for a managed server)

Task	Path
Create	1. Click **Enterprise Management** in the Work Items list. 2. In the **Servers** table, click on the name of a managed server. 3. In the server's Subscriptions table, click **Select Action**, then select **Create Subscription**.

Volumes in a library

AIX HP-UX Linux Solaris Windows

Task	Path
Create (check in, or check in and label)	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click a library name. 3. In the library's properties portlet, click the **Volumes** tab. 4. In the table, click **Select Action**, then select **Add Volumes**.
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Libraries for all Servers** table, click a library name. 3. In the library's properties portlet, click the **Volumes** tab. 4. In the table, select a volume. 5. Click **Select Action**, then select **Modify Volume**.

Volumes in a storage pool

Task	Path
Create	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **View Storage Pools**. 4. In the server's storage pools portlet, click a storage pool name. 5. In the storage pool's properties notebook, click the **Volumes** tab. 6. In the volumes table, click **Select Action**, then select **Add Volume**.

Task	Path
View and modify	1. Click **Storage Devices** in the Work Items list. 2. In the **Servers** table, select a server. 3. Click **Select Action**, then select **View Storage Pools**. 4. In the server's storage pools portlet, click a storage pool name. 5. In the storage pool's properties notebook, click the **Volumes** tab. 6. In the table, select a volume. 7. Click **Select Action**, then select **Modify Volume** or **View Contents**.

Quick paths to performing tasks

Use this quick path reference to learn how to perform some common tasks. Details about other tasks are available in the help for the Administration Center (for example, setting up library sharing).

Add a new storage device for the server to use

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **Add a Storage Device**.

   The wizard helps you to create a library, drives, paths, a device class, and storage pools for the device.

Add or remove an administrator

To add an administrator:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the server's properties notebook, click the **Administrators** tab.
4. In the **Administrators** table, click **Select Action**, then select **Add Administrator**.

To remove an administrator:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the servers properties notebook, click the **Administrators** tab.
4. In the **Administrators** table, select an administrator.
5. Click **Select Action**, then select **Remove Administrator**.

Check in and label volumes for a library

AIX HP-UX Linux Solaris Windows

1. Click **Storage Devices** in the Work Items list.
2. In the **Libraries for all Servers** table, click a library name.
3. In the library's properties notebook, click the **Volumes** tab.
4. In the **Volumes** table, click **Select Action**, then select **Add Volumes**.

Check out volumes from a library

AIX HP-UX Linux Solaris Windows

1. Click **Storage Devices** in the Work Items list.
2. In the **Libraries for all Servers** table, click a library name.
3. In the library's properties notebook, click the **Volumes** tab.

4. In the **Volumes** table, click **Select Action**, then select **Check Out Volumes**.

Create a disaster recovery plan

1. Click **Disaster Recovery Management** in the Work Items list.
2. In the **Servers** table, click a server name.
3. Click **Select Action**, then select **Create Recovery Plan**

Create or modify a schedule for the server (administrative schedule)

To create a schedule:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the server's properties notebook, click the **Administrative Schedules** tab.
4. In the **Schedules** table, click **Select Action**, then select **Create a Schedule**.

To modify a schedule:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the server's properties notebook, click the **Administrative Schedules** tab.
4. In the **Schedules** table, select a schedule.
5. Click **Select Action**, then select **Modify Schedule**.

Create or modify a script

To create a server script:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the server's properties notebook, click the **Scripts** tab.
4. In the **Scripts** table, click **Select Action**, then select **Create Script**,.

To modify a server script:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the server's properties notebook, click the **Scripts** tab.
4. In the **Scripts** table, select a script.
5. Click **Select Action**, then select **Modify Script**.

To create a script that performs key maintenance tasks:

1. Click **Server Maintenance** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **Create a Maintenance Script**.

Create a server group

1. Click **Enterprise Management** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the servers and server groups portlet, click **Server Groups**.
4. In the **Server Groups** table, click **Select Action**, then select **Create a Server Group**.

Display information about a storage pool volume

1. Click **Storage Devices** in the Work Items list.

2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **View Storage Pools**.
4. In the server's **Storage Pools** table, click the name of a storage pool.
5. In the storage pool's properties notebook, click the **Volumes** tab.

Display, modify or back up volume history

To display volume history:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **View Volume History**.

To modify volume history:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **View Volume History**.
4. In the **Volumes** table, select a volume.
5. Click **Select Action**, then select **Modify Volume History**.

To backup volume history:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **View Volume History**.
4. In the **Volumes** table, select a volume.
5. Click **Select Action**, then select **Backup Volume History**.

Enable LAN-free data movement

1. Click **Policy Domains and Client Nodes** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's policy domains table, click the name of a domain.
4. In the domain's properties portlet, click **Client Nodes**.
5. In the **Client Nodes** table, select a client node.
6. Click **Select Action**, then select **Enable LAN-free Data Movement**.

Expire inventory

1. Click **Enterprise Management** in the Work Items list.
2. In the **Servers** table, click a server name.
3. Click **Select Action**, then select **Expire Inventory**

z/OS

Format LAN-free volumes

To have the server format volumes automatically when they are requested by a storage agent:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the server's properties notebook, click the **Advanced Settings** tab.

To format volumes yourself and have them available in the storage pool before LAN-free operations begin:

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server that has a storage pool enabled for LAN-free operations.

3. Click **Select Action**, then select **View Storage Pools**.
4. In the server's **Storage Pools** table, select a LAN-free enabled storage pool.
5. Click **Select Action**, then select **Format LAN-free Volumes**.

Granting an administrator access to the Administration Center

1. Click the **Settings** tab above the Work Items list.
2. In the **Settings** list, click **User and Group Management**.
3. In the **Root** table, click "**all portal user groups**".
4. In the "**all portal user groups**" table, click **TSM_AdminCenter**.
5. In the **TSM_AdminCenter** portlet, click **New user**.

Issue a command to a server from the command line

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **Use Command Line**.

The command line is also available from the **Servers** tables that are displayed by clicking the following items in the Work Items list:

- Health Monitor
- Enterprise Management
- Policy Domains and Client Nodes
- Server Maintenance

Lock and unlock an administrator

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the server's properties notebook, click the **Administrators** tab.
4. In the **Administrators** table, select an administrator.
5. Click **Select Action**, then select **Modify Administrator**.

Lock or unlock a client node

1. Click **Policy Domains and Client Nodes** in the Work Items list.
2. In the **Policy Domains** table, select a server.
3. Click **Select Action**, then select **Search for Client Node**.
4. Click **Update Table**
5. In the **Client Nodes** table, click the name of a client node.
6. In the node's properties notebook, click the **Security** tab.

Move disaster recovery media

1. Click **Disaster Recovery Management** in the Work Items list.
2. In the **Servers** table, click a server name.
3. Click **Select Action**, then select **Move Disaster Recovery Media**

Access the Web client interface

Ensure that the Web client agent has been installed, configured, and started on the client node machine. See the *Backup-Archive Clients Installation and User's Guide* for instructions.

1. Click **Policy Domains and Client Nodes** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's policy domains table, click the name of a domain.

4. In the domain's properties portlet, click **Client Nodes**.
5. In the **Client Nodes** table, select a client node.
6. Click **Select Action**, then select **Launch Backup-archive Client**.

Note: You can use the **Communications** tab of a client node's properties notebook to change the Web client address.

Protect a network-attached storage (NAS) file server

AIX HP-UX Linux Solaris Windows

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **View Data Movers**.
4. In the server's data movers table, click **Select Action**, then select **Create NAS Data Mover**.

Query the activity log for all messages about a particular client node

1. Click **Health Monitor** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's health information portlet, click **Activity Log**.
4. In the **Filter** list, click **No filter** and select **Client Node Name**.
5. In the **Search string** field, type the name of the client node.
6. Click **Update Table**.

The activity log can also be displayed from the server's properties notebook.

Query the status of the database or database volumes

To display the overall status of the database:

1. Click **Health Monitor** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's health information portlet, click **Database and Recovery Log Information**.

To display details about database volumes, including the status of mirror volumes:

1. Click **Health Monitor** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **Server Properties**.
4. In the server's properties notebook, click the **Database and Log** tab.
5. In the **Database** table, click the name of a volume.

Query the status of devices

1. Click **Health Monitor** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's health information portlet, click **Storage Device Status**.

Rename an administrator

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's properties notebook, click the **Administrators** tab.
4. In the **Administrators** table, select an administrator.
5. Click **Select Action**, then select **Modify Administrator**.

Rename a server

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **Remove Server Connection**.
4. Click **Select Action**, then select **Add Server Connection**.

Rename a storage pool

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **View Storage Pools**.
4. In the server's **Storage Pools** table, select a storage pool.
5. Click **Select Action**, then select **Modify Storage Pool**.

Run a script

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's properties notebook, click the **Scripts** tab.
4. In the **Scripts** table, select a script.
5. Click **Select Action**, then select **Run Script**.

Schedule NDMP backups for NAS file servers

AIX HP-UX Linux Solaris Windows

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click a server name.
3. In the server's properties notebook, click the **Administrative Schedules** tab.
4. Click **Select Action**, then select **Create a Schedule**.

Set up one server as the configuration manager for other, managed servers

1. Click **Enterprise Management** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **Set Up Enterprise Configuration**.

Set up server-to-server communication

Part A:

1. Click **Enterprise Management** in the Work Items list.
2. In the **Servers** table, select a server.
3. Click **Select Action**, then select **Server-to-Server Communication Settings**.

Part B:

1. Click **Enterprise Management** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the servers and server groups portlet, click **Servers**.
4. In the **Servers** table, click **Select Action**, then select **Define Server**.

Specify the amount of time between automatic expiration processes

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's properties notebook, click the **General** tab.

Specify the maximum life of a restartable restore session

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's properties notebook, click the **General** tab.

Specify the size of the database buffer pool

1. Click **Storage Devices** in the Work Items list.
2. In the **Servers** table, click the name of a server.
3. In the server's properties notebook, click the **Database and Log** tab.
4. In the **Database** table, select a database volume.
5. Click **Select Action**, then select **Database Properties**.

Vary disk drives online and offline

1. Click **Storage Devices** in the Work Items list.
2. In the **Libraries** table, click a library name.
3. In the library's properties notebook, click the **Drives** tab.
4. In the **Drives** table, select a drive.
5. Click **Select Action**, then select **Modify Drive**.

Functions and features in the Administration Center

The Administration Center offers the functions of most administrative commands, as well as unique functions such as the health monitor and wizards to help you perform complex tasks. However, some Tivoli Storage Manager features and functions are limited or not supported in the Administration Center in Tivoli Storage Manager Version 5.3.2.

The following table shows commands that are supported with some restrictions or not yet supported in the Administration Center. Use the command line if the command or command parameter that you need is not available in the Administration Center.

Command	**Supported in the Administration Center**
AUDIT LIBRARY	No
CLEAN DRIVE	No Setting up automatic cleaning *is* supported.
COPY DOMAIN	No

Command	Supported in the Administration Center
DEFINE COPYGROUP TYPE=ARCHIVE	Supported except for these parameters: • RETINIT • RETMIN AIX Windows Solaris HP-UX Linux These parameters are needed only to support IBM System Storage Archive Manager. z/OS These parameters are needed only to support Tivoli Storage Manager for Data Retention Protection.
DEFINE DATAMOVER TYPE=SCSI (Windows servers only)	No This command is needed only to support configuration for server-free data movement on Windows servers.
DEFINE SPACETRIGGER	Supported for databases and recovery logs, but not for storage pools.
DEFINE STGPOOL	Supported except for the RECLAMATIONTYPE parameter This parameter is needed only for support of EMC Centera devices.
DELETE DATAMOVER	No
Event logging commands (BEGIN EVENTLOGGING, END EVENTLOGGING, ENABLE EVENTS, DISABLE EVENTS)	No Some SNMP options can be viewed in the interface, in a server's properties notebook.
ESTIMATE DBREORG	No
MOVE MEDIA	No
RENAME FILESPACE	No
RESET commands (RESET BUFPOOL, RESET DBMAXUTIL, RESET LOGCONSUMPTION, RESET LOGMAXUTIL)	No
RESTORE VOLUME	Yes, except use the command line to restore random-access storage pool volumes.
SET ACTLOGRETENTION	No

Command	Supported in the Administration Center
SET ARCHIVERETENTIONPROTECTION	No
SET EVENTRETENTION	No
SET QUERYSCHEDPERIOD	No
SET RETRYPERIOD	No
SET SCHEDMODES	No
SET SERVERNAME	No
SET SUMMARYRETENTION	No
SET TAPEALERTMSG	No
SET TOCLOADRETENTION	No
SETOPT	Only the following server options can be modified using the Administration Center: • BUFPOOLSIZE • EXPINTERVAL • RESTOREINTERVAL
VALIDATE LANFREE	Use the Enable LAN-free Data Movement wizard to get this function.

Part 2. Configuring and Managing Server Storage

© Copyright IBM Corp. 1993, 2005

Chapter 3. Introducing Storage Devices

This chapter introduces key concepts that you must be familiar with to work with Tivoli Storage Manager storage devices. It also describes what you will find in the storage device chapters.

"IBM Tivoli Storage Manager Storage Devices" on page 62
"IBM Tivoli Storage Manager Storage Objects" on page 62
"IBM Tivoli Storage Manager Volumes" on page 68
"Planning for Server Storage" on page 70
"Selecting a Device Configuration" on page 71
"How IBM Tivoli Storage Manager Mounts and Dismounts Removable Media" on page 76
"How IBM Tivoli Storage Manager Uses and Reuses Removable Media" on page 77
"Configuring Devices" on page 80

How to Use the Server Storage Chapters

If you are new to Tivoli Storage Manager, you should begin by familiarizing yourself with the concepts presented in this chapter. The other chapters in this part of the book will help you to do the following:

Goal	Chapter
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, "Using Magnetic Disk Devices," on page 83
Physically attach storage devices to your system and to install and configure the required device drivers.	Chapter 5, "Attaching Devices to the Server System," on page 91
Configure devices to use with Tivoli Storage Manager, and to see detailed scenarios of representative device configurations.	Chapter 6, "Configuring Storage Devices," on page 99
Plan, configure, and manage an environment for NDMP operations	Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143
Perform routine operations such as labeling volumes, checking volumes into automated libraries, and maintaining storage volumes and devices.	Chapter 8, "Managing Removable Media Operations," on page 167
Define and manage device classes.	Chapter 9, "Defining Device Classes," on page 197
Understand storage pool and storage volume concepts, and to define and manage storage pools and storage volumes.	Chapter 10, "Managing Storage Pools and Volumes," on page 217

© Copyright IBM Corp. 1993, 2005

IBM Tivoli Storage Manager Storage Devices

Tivoli Storage Manager devices may be real physical devices, such as disk drives or tape drives, or logical devices, such as files on a disk or storage on another server. See the following sections for details:

- "Libraries"
- "Drives" on page 64
- "Files on Disk as Sequential Volumes (FILE)" on page 65
- "Sequential Volumes on Another IBM Tivoli Storage Manager Server (SERVER)" on page 65
- "Disk Devices" on page 65
- "Data Movers" on page 68

For a summary, see Table 5 on page 80. For details about specific devices that are supported, visit the IBM Tivoli Storage Manager Web site at this address www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

IBM Tivoli Storage Manager Storage Objects

The Tivoli Storage Manager devices and media are represented by objects that you define and that are stored in the database. The objects contain information about the devices and media. You can query, update, and delete the objects. The storage objects are:

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

The following sections describe these objects.

Libraries

A physical library is a collection of one or more drives that share similar media mounting requirements. That is, the drive may 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 3494 library for private, scratch volumes, and scratch WORM (write once, read many) volumes).

Tivoli Storage Manager supports a variety of library types described in the following sections.

Shared Libraries

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

ACSLS Libraries

An ACSLS library is a type of external library that is controlled by the StorageTek software, Automated Cartridge System Library Software (ACSLS). The server can act as a client application to the ACSLS software to use the drives.

The ACSLS library is a collection of drives managed by the StorageTek ACSLS media management software. The StorageTek software performs the following functions

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

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

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

Manual Libraries

In a manual library, an operator mounts the volumes. You cannot combine drives of different types or formats, such as Digital Linear Tape (DLT) and 8 mm, in a single manual library. A separate manual library would have to be created for each device type.

When the server determines that a volume must be mounted on a drive in a manual library, the server issues mount request messages that prompt an operator to mount the volume. The server sends these messages to the server console and to administrative clients that were started by using the special *mount mode* or *console mode* parameter.

Manual libraries are also used as logical entities for sharing sequential FILE volumes with other servers.

For help on configuring a manual library, see Chapter 6, “Configuring Storage Devices,” on page 99. For information about how to monitor mount messages for a manual library, see “Mount Operations for Manual Libraries” on page 184.

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 may be of different types. A SCSI library may contain drives of mixed technologies, for example LTO Ultrium and DLT drives.

Some examples of this library type are:

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

Note: The IBM 3494 Tape Library Dataserver, although it has a SCSI interface, is defined as a 349X library type.

For help on configuring a SCSI library, see Chapter 6, “Configuring Storage Devices,” on page 99.

349X Libraries

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

The drives in a 3494 library must be of one type only (either IBM 3490, 3590, or 3592). For help on configuring a 349X library, see Chapter 6, "Configuring Storage Devices," on page 99.

External Libraries

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

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

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

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

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

For a definition of the interface that Tivoli Storage Manager provides to the external media management system, see Appendix A, "External Media Management Interface Description," on page 741.

Drives

Each drive mechanism within a device that uses removable media is represented by a drive object. For devices with multiple drives, including automated libraries, each drive is separately defined and must be associated with a library. Drive definitions can include such information as the element address (for drives in SCSI libraries), how often the drive is cleaned (for tape drives), and whether or not the drive is online.

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

Device Class

Each device defined to Tivoli Storage Manager is associated with one device class. That device class specifies a device type and media management information, such as recording format, estimated capacity, and labeling prefixes. A device class for a tape drive must also specify a library.

A device type identifies a device as a member of a group of devices that share similar media characteristics. For example, the 8MM device type applies to 8-mm tape drives. Device types include a variety of removable media types as well as FILE and SERVER.

Disk Devices

Tivoli Storage Manager lets you define DISK volumes using a single command. You can also use space triggers to automatically create preassigned private volumes when predetermined space utilization thresholds have been exceeded.

For important disk-related information, see "Requirements for Disk Subsystems, File Systems and Remote File Access" on page 83.

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. See Chapter 9, "Defining Device Classes," on page 197 and the *Administrator's Reference* for more information about supported removable media device types.

Files on Disk as Sequential Volumes (FILE)

This device type allows you to create sequential volumes by creating files on disk storage. To the server, these files have the characteristics of a tape volume. Unless sharing with storage agents is specified, the FILE device type does not require you to define library or drive objects; only a device class is required. 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:

- 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,

For important disk-related information, see "Requirements for Disk Subsystems, File Systems and Remote File Access" on page 83.

Sequential Volumes on Another IBM Tivoli Storage Manager Server (SERVER)

This device type allows you to 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. You must define a device class and a target server. No library or drive definition is required.

Virtual volumes can be used 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 601.

Library, Drive, and Device Class

These three objects taken together represent a physical storage entity as 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 "Defining Drives" on page 138 and "Managing Drives" on page 188.
- For more information about the library object, see "Defining Libraries" on page 137 and "Managing Libraries" on page 186.
- For more information about the device class object, see Chapter 9, "Defining Device Classes," on page 197.

Storage Pool and Storage Pool Volume

A storage pool is a collection of storage pool 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 8mm tape volumes contains only 8-mm tape volumes. You can control the characteristics of storage pools, such as whether scratch volumes are used. For details about defining storage pools, see Chapter 10, "Managing Storage Pools and Volumes," on page 217.

Tivoli Storage Manager supplies default disk storage pools. For more information, see "Configuring Random Access Volumes on Disk Devices" on page 87.

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 228 and "Preparing Volumes for Sequential Access Storage Pools" on page 229.

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 shows these relationships.

Figure 4. Relationships between Storage and Device Objects

For more information about the storage pool and volume objects, see Chapter 10, "Managing Storage Pools and Volumes," on page 217.

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 Tivoli Storage Manager server or client resources
- Without using significant 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.

Path

Paths allow access to drives 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 140 and "Managing Paths" on page 193.

Server

You need to define a server object for the following purposes:

- To use a library that is on a SAN and that is managed by another Tivoli Storage Manager server. You must define that server and then specify it as the library manager when you define the library. For more information, "Setting up the Library Client Servers" on page 110.
- To use LAN-free data movement. You define the storage agent as a server. For more information, see *IBM Tivoli Storage Manager Storage Agent User's Guide*.
- To store client data in the storage of another Tivoli Storage Manager server. For more information, see "Using Virtual Volumes to Store Data on Another Server" on page 601.

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

IBM Tivoli Storage Manager Volumes

Tivoli Storage Manager volumes are classified according to status: private, scratch, and scratch WORM (write once, read many). Scratch WORM status applies to 349X libraries only and only if the volumes are IBM 3592 WORM volumes.

- 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 229. For information about changing the status of a volume (for example, from private to scratch) in an automated library, see "Changing the Status of a Volume" on page 179.

- 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.

Note: 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.

The 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 "Checking New Volumes into a Library" on page 171.

Planning for Server Storage

This section discusses how to evaluate your environment to determine the device classes and storage pools for your server storage.

1. Determine which drives and libraries are supported by the server. For more information on device support, see "Devices Supported by Tivoli Storage Manager" on page 91.
2. Determine which storage devices may be selected for use by the server. For example, determine how many tape drives you have that you will allow the server to use. For more information on selecting a device configuration, see "Selecting a Device Configuration" on page 71.

   The servers can share devices in libraries that are attached through a SAN. If the devices are not on a SAN, the server expects to have exclusive use of the drives defined to it. If another application (including another Tivoli Storage Manager server) tries to use a drive while the server to which the drive is defined is running, some server functions may fail. See www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html for more information about specific drives and libraries.
3. Determine the device driver that supports the devices. For more information on device driver support, see "Installing and Configuring Device Drivers" on page 93.
4. Determine how to attach the devices to the server. For more information on attaching devices, see "Attaching an Automated Library Device" on page 92.
5. Determine whether to back up client data directly to tape or to a storage hierarchy.
6. Determine which client data is backed up to which device, if you have multiple device types.
7. Determine the device type and device class for each of the available devices. Group together similar devices and identify their device classes. For example, create separate categories for 4-mm and 8-mm devices.

   Note: For sequential access devices, you can categorize the type of removable media based on their capacity. For example, standard length cartridge tapes and longer length cartridge tapes require different device classes.
8. Determine how the mounting of volumes is accomplished for the devices:
   - Devices that require operators to load volumes must be part of a defined MANUAL library.
   - Devices that are automatically loaded must be part of a defined SCSI or 349X. Each automated library device is a separate library.
   - Devices that are controlled by 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 using the storage of another Tivoli Storage Manager server, consider network bandwidth and network traffic. If your network resources constrain your environment, you may have problems using the SERVER device type efficiently.

   Also consider the storage resources available on the target server. Ensure that the target server has enough storage space and drives to handle the load from the source server.

10. Determine the storage pools to set up, based on the devices you have and on user requirements. Gather users' requirements for data availability. Determine which data needs quick access and which does not.
11. Be prepared to label removable media. You may want to create a new labeling convention for media so that you can distinguish them from media used for other purposes.

Selecting a Device Configuration

The following sections describe ways that you can configure your storage devices to work with Tivoli Storage Manager:

- "Devices on a Local Area Network"
- "Devices on a Storage Area Network"
- "LAN-Free Data Movement" on page 73
- "Network-Attached Storage" on page 74

For information about supported devices and Fibre Channel hardware and configurations, see the following Web site at this address www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html

Devices on a Local Area Network

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. See "Sharing an IBM 3494 Library by Static Partitioning of Drives" on page 120 and "Sharing an IBM 3494 Library Among Servers" on page 117 for details.

For information on the categories of libraries supported by Tivoli Storage Manager, see "Libraries" on page 62.

Devices on a Storage Area Network

A storage area network (SAN) is a dedicated storage network that can improve system performance. On a SAN you can consolidate storage and relieve the distance, scalability, and bandwidth limitations of LANs and wide area networks (WANs). Using Tivoli Storage Manager in a SAN allows the following functions:

- Sharing storage devices among multiple Tivoli Storage Manager servers. For more information on sharing storage devices, see "Configuring SCSI Libraries Shared Among Servers on a SAN" on page 107.
- Allowing Tivoli Storage Manager clients, through a storage agent on the client machine, to move data directly to storage devices (LAN-free data movement).

In a SAN you can share storage devices that are supported by the Tivoli Storage Manager device driver. This includes most SCSI devices. See Chapter 5, "Attaching Devices to the Server System," on page 91 for device driver setup information.

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

Figure 5. Library Sharing in a Storage Area Network (SAN) Configuration. The servers communicate over the LAN. The library manager controls the library over the SAN. The library client stores data to the library devices over the SAN.

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

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

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

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

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

LAN-Free Data Movement

Tivoli Storage Manager allows a client, through a storage agent, to directly back up and restore data to a tape library on a SAN. Figure 6 shows a SAN configuration in which a client directly accesses a tape or FILE library to read or write data.

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

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

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

1. The client begins a backup operation. The client and the server exchange policy information over the LAN to determine the destination of the backed up data.

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

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

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

Note: See the IBM Tivoli Storage Manager home page at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html for the latest information on clients that support the feature.

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 third-party software. Instead, they interact with programs like Tivoli Storage Manager through industry-standard network protocols, such as NDMP. Tivoli Storage Manager uses the NDMP protocol to communicate with and direct backup and restore operations for NAS file servers.

Using NDMP, Tivoli Storage Manager can back up and restore images of complete file systems. NDMP allows the Tivoli Storage Manager server to control the backup of a NAS file server. The file server transfers the backup data to a drive in a SCSI-attached tape library. The NAS file server can be distant from the Tivoli Storage Manager server.

Tivoli Storage Manager tracks file system image backups on tape, and has the capability to perform NDMP file-level restores. For more information regarding NDMP file-level restores, see “NDMP File-Level Restore” on page 75.

Figure 7. Network- Attached Storage (NAS) Configuration

Tivoli Storage Manager and Other NAS Backup Methods

When Tivoli Storage Manager uses NDMP to protect NAS file servers, the Tivoli Storage Manager server controls operations while the NAS file server transfers the data. To use a backup-archive client to back up a NAS file server, mount the NAS file server file system on the client machine (with either an NFS mount or a CIFS map) and back up as usual. The following table compares the two methods:

Table 4. Comparing NDMP Operations and Tivoli Storage Manager Backup-Archive Client Operations

NDMP	Tivoli Storage Manager Backup-Archive Client
Network data traffic is less because the Tivoli Storage Manager server controls operations remotely, but the NAS file server moves the data locally.	Network data traffic is greater because all backup data goes across the LAN from the NAS file server to the client and then to the Tivoli Storage Manager server.
Less file server processing is required to back up a file system 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.
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.

NDMP Backup Operations

In backup images produced by NDMP operations for a NAS file server, Tivoli Storage Manager creates NAS file system 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. 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.

The following operations are not supported for data that has been backed up by using NDMP:

- Storage pool migration
- Storage pool backup and restore
- Reclamation
- Move data operations
- Export and import operations
- Backup set generation

NDMP File-Level Restore

Tivoli Storage Manager provides an option whereby file-level restores can be performed based on backup images produced by NDMP operations. Your choices can be summarized as follows:

- If you enable the file-level restore option, the Tivoli Storage Manager server collects and stores file level information when backing up file system images by using NDMP operations. This requires additional processing and network

resources. However, you will be able to use the Web client to query and present file-level information to a user who can then select files and directories to restore.

- If you do not enable the file-level restore option, the Tivoli Storage Manager server backs up file system images by using NDMP operations without gathering file-level information. You will not be able to list the files on the client. You will be able to restore them if you already know what they are. This is the default setting.

If you choose to enable the file-level restore option, the Tivoli Storage Manager server constructs a table of contents (TOC) of file-level information for a single backup image produced by NDMP operations. The TOC is stored in the server storage of the Tivoli Storage Manager server. The server can then retrieve the TOC so that information can be queried by the client or server.

The TOC is created when backing up using backup images produced by NDMP operations using the:

- BACKUP NAS client command, with include.fs.nas specified in the client options file or specified in the client options set
- BACKUP NODE server command

Directory-Level Backup and Restore

If you have a large NAS file system, initiating a backup on 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.

How IBM Tivoli Storage Manager Mounts and Dismounts Removable Media

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 IBM Tivoli Storage Manager Uses and Reuses Removable Media

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

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

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

Figure 8 on page 78 shows a typical life cycle for removable media. The numbers (such as **1**) refer to numbers in the figure.

Figure 8. Simplified View of the Life Cycle of a Tape

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

   See "Labeling Removable Media Volumes" on page 168.
2. If you plan to define volumes to a storage pool associated with a device, you should check in the volume with its status specified as private. Use of scratch volumes is more convenient in most cases.
3. A client sends data to the server for backup, archive, or space management. The server stores the client data on the volume. Which volume the server selects **3** depends on:
   - The policy domain to which the client is assigned.
   - The management class for the data (either the default management class for the policy set, or the class specified by the client in the client's include/exclude list or file).
   - The storage pool specified as the destination in either the management class (for space-managed data) or copy group (for backup or archive data). The storage pool is associated with a device class, which determines which device and which type of media is used.
   - Whether the maximum number of scratch volumes that a server can request from the storage pool has been reached when the scratch volumes are selected.

- Whether collocation is enabled for that storage pool. When collocation is enabled, the server attempts to place data for different client nodes, groups of client nodes, or client file spaces on separate volumes. For more information, see "Keeping a Client's Files Together: Collocation" on page 258.

Figure 9 shows more detail about the policies and storage pool specifications which govern the volume selection described in step 3.

Figure 9. 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 358.
   - Movement and deletion of file spaces by an administrator.
   - Automatic reclamation of media **5**

     The amount of data on the volume and the reclamation threshold set for the storage pool affects when the volume is reclaimed. When the volume is reclaimed, any valid, unexpired data is moved to other volumes or possibly to another storage pool (for storage pools with single-drive libraries).
   - Collocation, by which Tivoli Storage Manager attempts to keep data belonging to a single client node, group of client nodes, or client file space on a minimal number of removable media in a storage pool.

   If the volume becomes empty because all valid data either expires or is moved to another volume, the volume is available for reuse (unless a time delay has been specified for the storage pool). The empty volume becomes a scratch volume if it was initially a scratch volume. The volume starts again at step 3 on page 78.
5. You determine when the media has reached its end of life.

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

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

Configuring Devices

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

Table 5 summarizes the definitions that are required for different device types.

Table 5. Required Definitions for Storage Devices

		Required Definitions			
Device	**Device Types**	**Library**	**Drive**	**Path**	**Device Class**
Magnetic disk	DISK	—	—	—	Yes [1]
	FILE [2]	—	—	—	Yes
Tape	3590 3592 4MM 8MM DLT ECARTRIDGE [3] LTO NAS QIC VOLSAFE	Yes	Yes	Yes	Yes
Removable media (file system)	REMOVABLEFILE	Yes	Yes	Yes	Yes
Virtual volumes	SERVER	—	—	—	Yes

1 The DISK device class exists at installation and cannot be changed.

2 FILE libraries, drives and paths are required for sharing with storage agents.

3 The ECARTRIDGE device type is for StorageTek's cartridge tape drives such as the SD-3, 9480, 9890, and 9940 drives.

Mapping Devices to Device Classes

As an example of mapping devices to device classes, assume that you have the following devices to use for server storage:

- Internal disk drives
- An automated tape library with 8-mm drives
- A manual DLT tape drive

You can map storage devices to device classes as shown in Table 6.

Table 6. Mapping Storage Devices to Device Classes

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

You must define any device classes that you need for your removable media devices such as tape drives. See Chapter 9, "Defining Device Classes," on page 197 for information on defining device classes to support your physical storage environment.

Mapping Storage Pools to Device Classes and Devices

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

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

- Immediate access to certain backed-up files, such as accounts receivable and payroll accounts.

  These files should be stored on disk. However, you need to ensure that data is moved from the disk to prevent it from becoming full. You can set up a storage hierarchy so that files can migrate automatically from disk to the automated tape library.
- Periodic access to some archived files, such as monthly sales and inventory reports.

  These files can be stored on 8-mm tapes, using the automated library.
- Occasional access to backed-up or archived files that are rarely modified, such as yearly revenue reports.

  These files can be stored using the DLT drive.

To match user requirements to storage devices, you define storage pools, device classes, and, for device types that require them, libraries and drives. For example, to set up the storage hierarchy so that data migrates from the BACKUPPOOL to 8-mm tapes, you specify BACKTAPE1 as the next storage pool for BACKUPPOOL. See Table 7.

Table 7. 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

Table 7. Mapping Storage Pools to Device Classes, Libraries, and Drives (continued)

Storage Pool	Device Class	Library (Hardware)	Drives	Volume Type	Storage Destination
BACKTAPE1	8MM_CLASS	AUTO_8MM (Exabyte EXB-210)	DRIVE01, DRIVE02	8mm 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 default disk storage pools named BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL. For more information, see "Configuring Random Access Volumes on Disk Devices" on page 87.

Chapter 4. Using Magnetic Disk Devices

Tivoli Storage Manager uses magnetic disk devices to do the following:

- Store the database and the recovery log. For details, see Chapter 19, "Managing the Database and Recovery Log," on page 489.
- Store client data that has been backed up, archived, or migrated from client nodes. The client data is stored in storage pools. Procedures for configuring disk storage of client data are described in this chapter.
- Store backups of the database and export and import data. See "Using FILE Volumes for Database Backups and Export Operations" on page 90.

See the following sections:

Tasks:
"Configuring Random Access Volumes on Disk Devices" on page 87
"Configuring FILE Sequential Volumes on Disk Devices" on page 88
"Varying Disk Volumes Online or Offline" on page 89
"Using Cache" on page 89
"Freeing Space on Disk" on page 89
"Specifying Scratch FILE Volumes" on page 90
"Using FILE Volumes for Database Backups and Export Operations" on page 90

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Note: Some of the tasks described in this chapter require an understanding of Tivoli Storage Manager storage objects. For an introduction to these storage objects, see "IBM Tivoli Storage Manager Storage Objects" on page 62.

Configuring Disk Devices

Tivoli Storage Manager stores data on magnetic disks in two ways:

- In random access volumes, as data is normally stored on disk. See "Configuring Random Access Volumes on Disk Devices" on page 87.
- In files on the disk that are treated as sequential access volumes. See "Configuring FILE Sequential Volumes on Disk Devices" on page 88.

Requirements for Disk Subsystems, File Systems and Remote File Access

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

© Copyright IBM Corp. 1993, 2005

operations when they can fail after a successful write report to Tivoli Storage Manager. A write failure after a successful notification constitutes a data integrity problem because the data that was reported as successfully written is unavailable for retrieval. In this situation, all data subsequently written is also at risk due to positioning mismatches within the target file. To avoid these problems, ensure that disk subsystems and file systems, whatever implementation you use, are always able to return data when the data is requested.

Remote file systems in particular may 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.

Comparing 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. Table 8 provides some general information about the characteristics of DISK devices (random access) and FILE devices (sequential access) and the benefits of each. The particular advantages provided by either device type will depend on the operating system on which your Tivoli Storage Manager server is running.

Table 8. Comparing Random Access and Sequential Access Disk Devices

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
Storage space allocation and tracking	Disk blocks.	Volumes.	Space allocation and tracking by blocks incurs higher overhead (more database storage space, and more processing power) than space allocation and tracking by volume.
Concurrent volume access	A volume can be accessed concurrently by different operations.	Only one operation can access the volume at a time.	Concurrent volume access means that two or more different operations can access the same volume at the same time.
Client restore operations	One session per restore.	Multiple concurrent sessions accessing different volumes simultaneously on both the server and the storage agent.	Multi-session restore enables backup-archive clients to perform multiple restore sessions for no-query restore operations, increasing the speed of restores. For more information, see “Concepts for Client Restore Operations” on page 423.

Table 8. Comparing Random Access and Sequential Access Disk Devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
Available for use in LAN-free backup	Not available.	Available for LAN-free backup using Tivoli SANergy, a separate product, licensed to users through the Tivoli Storage Manager product. Tivoli SANergy is included with some versions of Tivoli Storage Manager.	Using LAN-free backup, data moves over a dedicated storage area network (SAN) to the sequential-access storage device, freeing up bandwidth on the LAN. For more information, see "LAN-Free Data Movement" on page 73.
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 87. For more information about volumes on FILE devices, see "Configuring FILE Sequential Volumes on Disk Devices" on page 88.
Tivoli Storage Manager server caching (after files have been migrated to the next storage pool in the storage pool hierarchy)	Server caching is available, but overhead is incurred in freeing the cached space. For example, as part of a backup operation, the server must erase cached files to make room for storing new files.	Server caching is not necessary because access times are comparable to random access (DISK) access times.	Caching can improve how quickly the Tivoli Storage Manager server retrieves files during client restore or retrieve operations. For more information, see "Using Cache on Disk Storage Pools" on page 248.
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 267.
Aggregate reconstruction	Not available; the result is wasted space.	Aggregate reconstruction occurs as part of the reclamation process. It is also available using the RECONSTRUCT parameter on the MOVE DATA and MOVE NODEDATA commands.	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. For more information, see "How IBM Tivoli Storage Manager Reclamation Works" on page 267.

Table 8. Comparing Random Access and Sequential Access Disk Devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
Available for use as copy storage pools	Not available.	Available.	A copy storage pool provides an additional level of protection for client data. For more information, see "Backing Up Storage Pools" on page 648.
File location	Volume location is limited by the trigger prefix or by manual specification.	FILE volumes use directories. A list of directories may be specified. If directories correspond with file systems, performance is optimized.	
Restoring the database to an earlier level	See comments.	Use the REUSEDELAY parameter to retain volumes in a pending state; volumes are not rewritten until the specified number of days have elapsed. During database restoration, if the data is physically present, it can be accessed after 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 a Storage Pool Volume" on page 671. For more information about reuse delay, see "Delaying Reuse of Volumes for Recovery Purposes" on page 651. For command syntax, refer to the *Administrator's Reference*.
Migration	Performed by node and filespace. Migration from random-access pools can use multiple processes.	Performed by volume. Migration from sequential-access pools is limited to a single process per pool.	For more information, see "Migration for Disk Storage Pools" on page 239.
Storage pool backup	Performed by node and filespace. Every storage pool backup operation must check every file in the primary pool to determine whether the file must be backed up.	Performed by volume. For a primary pool, there is no need to scan every object in the primary pool every time the pool is backed up to a copy pool.	For more information, see "Overview: Storage Pools" on page 218.

Table 8. Comparing Random Access and Sequential Access Disk Devices (continued)

Function	Random Access (DISK)	Sequential Access (FILE)	Comment
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 668 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 a Client's Files Together: Collocation" on page 258.

Configuring Random Access Volumes on Disk Devices

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

Note: Define storage pool volumes on disk drives that reside on the server machine, not on remotely mounted file systems. Network attached drives can compromise the integrity of the data that you are writing. See "Disk Devices" on page 65 for more information on requirements for local and remote file systems.

Do the following 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 223 for details.
2. Prepare a volume for use in a random access storage pool by defining the volume. For example, you want to define a 21MB volume for the ENGBACK1 storage pool. You want the volume to be located in the path */opt/tivoli/tsm/server/bin* and named stgvol.002. Enter the following command:

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

   If you do not specify a full path name, the command uses the current path. See "Defining Storage Pool Volumes" on page 229 for details.

   The DSMFMT utility is available from the operating system command line. See the *Administrator's Reference* for details.
3. Do one of the following:
   - Specify the new storage pool as the destination for client files that are backed up, archived, or migrated, by modifying existing policy or creating new policy. See Chapter 13, "Implementing Policies for Client Data," on page 357 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 225.

Configuring FILE Sequential Volumes on Disk Devices

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

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

1. Define a device class with device type FILE.

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

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

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

   See "Defining and Updating FILE Device Classes" on page 204.

   To store database backups or exports on FILE volumes, this step is all you need to do to prepare the volumes. You can use FILE sequential volumes to transfer data for purposes such as electronic vaulting. For example, you can send the results of an export operation or a database backup operation to another location. At the receiving site, the files can be placed on tape or disk. You can define a device class with a device type of FILE. For more information, see "Defining Device Classes for Backups" on page 653 and "Planning for Sequential Media Used to Export Data" on page 618.

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

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

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

   This command defines storage pool ENGBACK2 with device class FILECLASS.

   See "Defining or Updating Primary Storage Pools" on page 220 for details.

   To allow Tivoli Storage Manager to use scratch volumes for this device class, specify a value greater than zero for the number of maximum scratch volumes when you define the device class. If you do set MAXSCRATCH=0 to not allow scratch volumes, you must define each volume to be used in this device class. See "Preparing Volumes for Sequential Access Storage Pools" on page 229 for details.

3. Do one of the following:

   - Specify the new storage pool as the destination for client files that are backed up, archived, or migrated, by modifying existing policy or creating new policy. See Chapter 13, "Implementing Policies for Client Data," on page 357 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 225.

   You can also set up pre-defined sequential volumes with the DEFINE VOLUME command using:

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

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

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

Varying Disk Volumes Online or Offline

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

To perform maintenance on a disk volume or to upgrade disk hardware, you can vary a disk volume offline. For example, to vary the disk volume named */storage/pool001* offline, enter:

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

If Tivoli Storage Manager encounters a problem with a disk volume, the server automatically varies the volume offline.

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

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

Using Cache

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

Using cache can improve how fast a frequently accessed file is retrieved. Faster retrieval can be important for clients storing space-managed files. If the file needs to be accessed, the copy in cache can be used rather than the copy on tape. However, using cache can degrade the performance of client backup operations and increase the space needed for the database. For more information, see "Using Cache on Disk Storage Pools" on page 248.

Freeing Space on Disk

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

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

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

- Use the EXPIRE INVENTORY command. See "Running Expiration Processing to Delete Expired Files" on page 391.
- Set the server option for the expiration interval, so that expiration processing runs periodically. See the *Administrator's Reference* for information on how to set the options.

Specifying Scratch FILE Volumes

You can specify a maximum number of scratch volumes for a storage pool that has a FILE device type. When the server needs a new volume, the server automatically creates a file that is a scratch volume, up to the number you specify. 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.

Using FILE Volumes for Database Backups and Export Operations

When you back up the database or export server information, Tivoli Storage Manager records information about the volumes used for these operations in the *volume history*. Tivoli Storage Manager will not allow you to reuse these volumes until you delete the volume information from the volume history. To reuse volumes that have previously been used for database backup or export, use the DELETE VOLHISTORY command. For information about the volume history and volume history files, see "Saving the Volume History File" on page 656.

Note: If your server is licensed for the disaster recovery manager (DRM) function, the volume information is automatically deleted during MOVE DRMEDIA command processing. For additional information about DRM, see Chapter 24, "Using Disaster Recovery Manager," on page 687.

Chapter 5. Attaching Devices to the Server System

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

Tasks:
"Attaching a Manual Drive"
"Attaching an Automated Library Device" on page 92
"Installing and Configuring Device Drivers" on page 93

Devices Supported by Tivoli Storage Manager

A list of supported storage devices is available on the Tivoli Storage Manager Products Technical Support web site at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html

Tivoli Storage Manager supports a wide range of manual devices and automated library devices:

Tape Devices
Tivoli Storage Manager supports the manual and automated tape devices listed at the Tivoli Storage Manager Web site.

Storage Area Network (SAN) Devices
Tivoli Storage Manager supports devices in a storage area network (SAN) environment, but the devices must be supported by the Tivoli Storage Manager device driver. See the Tivoli Storage Manager Web site for information about supported Fibre Channel hardware and configurations.

Attaching a Manual Drive

Perform the following steps to attach a manual drive:

1. Install the SCSI or FC adapter card in your system and associated drivers of the adapter card, if not already installed.

   Note: The attached devices should be on their own Host Bus Adapter (HBA) and should not share which other devices types (disk, CDROM, etc). The IBM tape drives have some special requirements on HBA and associated drivers. Check the support website for details.
2. Determine the SCSI IDs available on the SCSI or FC adapter card to which you are attaching the device. Find one unused SCSI ID for each drive.
3. Follow the manufacturer's instructions to set the SCSI ID for the drive to the unused SCSI IDs that you found. You may have to set switches on the back of the device or set the IDs on the operator's panel.

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

© Copyright IBM Corp. 1993, 2005

4. Follow the manufacturer's instructions to attach the device to your server system hardware.

   Attention:

   a. Power off your system before attaching a device to prevent damage to the hardware.
   b. Attach a terminator to the last device in the chain of devices connected on one SCSI adapter card.
5. Install the appropriate device drivers for attached tape devices. See "Installing and Configuring Device Drivers" on page 93.

   Note: The IBM device drivers support some specific Linux kernel levels. Check the support web site for details. For installing devices on zSeries systems, please see "Installing and Configuring zSeries Linux Fibre Channel Adapter (zfcp) Device Drivers on the IBM zSeries System" on page 97. Tivoli Storage Manager Version 5.3 supports the Tivoli Storage Manager device driver on all Linux kernel levels.

Attaching an Automated Library Device

Perform the following steps to attach an automated library device:

1. Install the SCSI or FC adapter card in your system and associated drivers of the adapter card, if not already installed.

   Note: The attached devices should be on their own Host Bus Adapter (HBA) and should not share which other devices types (disk, CDROM, etc). The IBM tape drives have some special requirements on HBA and associated drivers. Check the support website for details.
2. Determine the SCSI IDs available on the SCSI adapter card to which you are attaching the device. Find one unused SCSI ID for each drive, and one unused SCSI ID for the library or autochanger controller.

   Note: In some automated libraries, the drives and the autochanger share a single SCSI ID, but have different LUNs. For these libraries, only a single SCSI ID is required. Check the documentation for your device.
3. Follow the manufacturer's instructions to set the SCSI ID for the drives to the unused SCSI IDs that you found. You may have to set switches on the back of the device or set the IDs on the operator's panel.

   Note: Each device that is connected in a chain to a single SCSI bus must be set to a unique SCSI ID. If each device does not have a unique SCSI ID, serious system problems can arise.
4. Follow the manufacturer's instructions to attach the device to your server system hardware.

   Attention:

   a. Power off your system before attaching a device to prevent damage to the hardware.
   b. Attach a terminator to the last device in the chain of devices connected on one SCSI adapter card. Detailed instructions should be in the documentation that came with your hardware.
5. Install the appropriate device drivers for attached medium changer devices. See "Installing and Configuring Device Drivers" on page 93.

Note: The IBM device drivers support some specific Linux kernel levels. Check the support web site for details. For installing devices on zSeries systems, please see "Installing and Configuring zSeries Linux Fibre Channel Adapter (zfcp) Device Drivers on the IBM zSeries System" on page 97. Tivoli Storage Manager Version 5.3 supports the Tivoli Storage Manager device driver on all Linux kernel levels.

Setting the Library Mode

For the Tivoli Storage Manager server to access a SCSI library, set the device for the appropriate mode. This is usually called *random* mode; however, terminology may vary from one device to another. Refer to the documentation for your device to determine how to set it to the appropriate mode.

Notes:

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

Installing and Configuring Device Drivers

To use a tape or medium changer device, you must install the appropriate device driver. IBM device drivers are available for most IBM labeled devices. The Tivoli Storage Manager Passthru driver, which is provided with IBM Tivoli Storage Manager, is available for non-IBM devices.

Drivers for IBM Devices

IBM device drivers are available on the ftp site ftp://ftp.software.ibm.com/storage/devdrvr/. Installation and user's guides can be downloaded from the Doc folder. It is recommended that you install the most current driver available. See "Installing Device Drivers for IBM SCSI Tape Devices" on page 95 and "Installing Device Drivers for IBM 349X Libraries" on page 95 for more information.

- The IBM device driver *IBMtape* should be installed for the following devices:
  IBM 3570
  IBM 3590 and 3592
  IBM LTO Ultrium 3576, 3580, 3581, 3582, 3583, and 3584
- The IBM device driver *ibmatl* should be installed for IBM 3494 and 3495 devices.

Drivers for non-IBM Devices

Tivoli Storage Manager Version 5.3 supports the Passthru driver, which replaces the Tivoli Storage Manager Linux SCSI device driver. The Tivoli Storage Manager Passthru drivers require the Linux SCSI generic (sg) driver along with the Linux operating system to install the kernels. See "Configuring Tivoli Storage Manager Passthru Drivers for Tape Drives and Libraries" on page 96 for more information.

The Tivoli Storage Manager Passthru drivers are available at ftp://ftp.software.ibm.com/storage/tivoli-storage-management/maintenence/server. See the *IBM Tivoli Storage Manager*

for Linux Installation Guide for installation instructions and "Configuring Tivoli Storage Manager Passthru Drivers for Tape Drives and Libraries" on page 96 for more information.

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

Third party vendor device drivers are supported 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 third party vendor device driver is not recommended. Generic device drivers are not supported in WORM device classes. For more information, see the DEFINE DEVCLASS - GENERICTAPE command in the *Administrator's Reference*.

Determining Device Special File Names

To work with tape, medium changer, or removable media devices, Tivoli Storage Manager needs the device's special file name. You specify the device special file name when you issue the DEFINE PATH commands for drives and libraries.

When a device configures successfully, a logical file name is returned. Table 9 specifies the name of the device (or the special file name) that corresponds to the drive or library. In the examples, *x* denotes any number from 0 to 254.

Table 9. Device Examples

Device	Device Special File Name
Tape drives that are supported by the Tivoli Storage Manager Passthru driver	/dev/tsmscsi/mt*x*
Tape drives that are supported by the IBM device driver	/dev/IBMtape*x*
SCSI-attached libraries that are supported by the Tivoli Storage Manager Passthru driver	/dev/tsmscsi/lb*x*
Automatic Cartridge Facility feature of the IBM 3590 B11 as a library	/dev/IBMchanger*x*
IBM 349X libraries	/dev/3494lib
IBM 3581, 3583, 3584 libraries	/dev/IBMchanger*x*
Mount point to use on REMOVABLEFILE device type (CD-ROM) **Note:** Based on different Linux distributions on different architectures, the mount point can be different. Another possible mount point is /mnt/media/cdrom. The driver for the CD-ROM should be provided by the device vendor. The TSM Passthru driver and IBM tape drivers do not support CD-ROM devices.	/mnt/cdrom

Installing Device Drivers for IBM SCSI Tape Devices

For information on how to install device drivers for IBM SCSI devices, see the appropriate document:

- *IBM TotalStorage Tape Device Drivers Installation and User's Guide*
- *IBM Ultrium Device Drivers: Installation and User's Guide*

The guides can be downloaded from the FTP site at ftp://ftp.software.ibm.com/storage/devdrvr/. They are located in the Doc folder.

Depending on the device driver you are installing, a message is issued after completing the procedure in the manual.

- If you are installing the device driver for an IBM 3480 or 3490 tape device, you receive:

  ```
  IBMtapex Available
  ```

  where IBMtape*x* is the logical filename for the tape device.
- If you are installing the device driver for an IBM 3581, 3583, 3584, or 3590 Model B11, you receive:

  ```
  IBMtapex Available
  ```

  or

  ```
  IBMChangerx Available
  ```

  Note the value of *x*, which is assigned automatically by the system. To determine the special file name of your device, use the */dev/IBM* prefix with the name provided by the system. For example:
  - If the message is `IBMtape0 Available`, the special file name for the drive is */dev/IBMtape0*.
  - If the message is `IBMChanger2 Available`, the special file name for the media changer device is */dev/IBMChanger2*.

The file name may have additional characters at the end to indicate different operating characteristics, but these are not needed by Tivoli Storage Manager. Use the base file name in the *Device*= parameter of the DEFINE PATH command to assign a device to a drive (*/dev/IBMtapex*) or a library (*/dev/IBMChangerx*).

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

Installing Device Drivers for IBM 349X Libraries

For an IBM 349X Tape Library Dataserver, refer to *IBM TotalStorage Tape Device Drivers Installation and User's Guide* to install the driver. The driver is the IBM Tape Library Driver, which consists of the *ibmatl* (a daemon) and other components.

To determine the name to use to define the path for the library, check what you entered in the /etc/ibmatl.conf file for the library's symbolic name.

Obtaining Information About Your System's SCSI Devices

Information about the devices seen by your system is available in the file */proc/scsi/scsi.*This file contains a list of *every* detected SCSI device. For example, if a system contains some StorageTek and IBM libraries, a SAN Gateway, and some Quantum DLT drives, the */proc/scsi/scsi* file will look similar to this:

```
Attached devices:
Host: scsi2 Channel: 00 Id: 00 Lun: 00
  Vendor:  STK    Model: 9738     Rev: 2003
  Type:  Medium Changer        ANSI SCSI revision: 02
Host: scsi2 Channel: 00 Id: 01 Lun: 02
  Vendor:  PATHLIGHT Model: SAN Gateway     Rev: 32aC
  Type: Unknown         ANSI SCSI revision: 03
Host: scsi2 Channel: 00 Id: 01 Lun: 02
  Vendor:  QUANTUM  Model: DLT7000     Rev: 2560
  Type:  Sequential-Access         ANSI SCSI revision: 02
Host: scsi2 Channel: 00 Id: 01 Lun: 04
  Vendor: IBM     Model: 7337     Rev: 1.63
  Type:  Medium Changer         ANSI SCSI revision: 02
```

From this listing, the following device information is available: the host number, channel number, SCSI ID, Logical Unit number, vendor, firmware level, type of device, and the SCSI mode.

Configuring Tivoli Storage Manager Passthru Drivers for Tape Drives and Libraries

To use the Tivoli Storage Manager Linux Passthru driver, you must perform the following steps.

1. Verify that the device is connected to your system, and is powered on and active.
2. Verify that the device is correctly detected by your system by issuing this command:

   ```
   cat /proc/scsi/scsi
   ```
3. Ensure that both the Tivoli Storage Manager device driver package (tsmscsi) and the storage server package are installed.
4. There are two driver configuration methods available in the Tivoli Storage Manager device driver package: autoconf and tsmscsi. Both of these methods perform the following tasks:
   - Load the Linux SCSI generic driver (sg) to the kernel.
   - Create necessary special files for the Passthru driver.
   - Create device information files for libraries (*/dev/tsmscsi/libinfo*) and tape devices (*/dev/tsmscsi/mtinfo*).
5. Run the configuration method you prefer (autoconf or tsmscsi) for the Tivoli Storage Manager Passthru driver.
   - To run the autoconf configuration method, issue the following command:

     ```
     autoconf
     ```
   - To run the tsmscsi configuration method, perform the following steps:
     a. Copy the two sample configuration files located in the installation directory from *lb.conf.smp* and *mt.conf.smp* to *lb.conf* and *mt.conf*, respectively.
     b. Edit the lb.conf and mt.conf files. Add one stanza (as shown in the example at the top of the file) for each SCSI target, ID, and LUN combination. Each combination of SCSI target, ID, and LUN entries correspond to a tape drive or library you want configured. Make sure the files meet these requirements:
        - Remove the example located at the top of the files.
        - There must be a new line between each stanza.
        - There must be one new line after the last stanza.
        - There must be no pound sign (#) in either file.

c. Run the tsmscsi script from the device driver installation directory.

Note: For Tivoli Storage Manager Version 5.3 and later, you need to re-execute the autoconf or tsmscsi script to re-configure Tivoli Storage Manager devices following a reboot. In general the Linux SCSI generic driver is pre-loaded to the kernel. To verify that the driver is in the kernel, you can issue the following command:

```
> lsmod | grep sg
```

If the driver is not in the kernel, you can execute the autoconf or tsmscsi to load it.

6. Verify that the device is configured properly by viewing the text files for libraries (*/dev/tsmscsi/libinfo*) and tape devices (*/dev/tsmscsi/mtinfo*).
7. Determine the special file names for the tape drives and libraries:
   - To determine the names for the tape drives, issue the following command:

     ```
     > ls  /dev/tsmscsi/mt*
     ```

   - To determine the names for the libraries issue the following command:

     ```
     > ls  /dev/tsmscsi/lb*
     ```

   This information helps you identify which of the /dev/tsmscsi/mtx and /dev/tsmscsi/lbx special file names to provide the server when you issue a DEFINE PATH command.

Installing and Configuring zSeries Linux Fibre Channel Adapter (zfcp) Device Drivers on the IBM zSeries System

The zSeries Linux Fibre Channel Adapter (zcfp) device driver is a special adapter driver on zSeries system. It provides a standard SCSI interface for up level SCSI device drivers in Linux operating systems on zSeries platforms. It is used with two support modules - qdio for qdio devices and scsi_mod, for general SCSI support. Without the zfcp driver, the system cannot detect any SCSI device attached on the zSeries systems. Tivoli Storage Manager and IBM tape device drivers can run on zSeries platforms with Linux operating systems in both 31–bit and 64–bit environments, and support most Original Equipment Manufacturer (OEM) and IBM tape devices with fibre and SCSI interfaces.

In order to function properly on the zSeries system, IBM tape device drivers and Linux SCSI generic drivers should be loaded to the kernal *after* the two support modules and the zfcp driver have been loaded:

1. Load the two modules in this order:
   a. qdio module
   b. scsi_mod module
2. Perform Fibre Channel Protocol (FCP) mapping prior to installing the driver.
3. Install the zfcp driver.
4. Load the Linux SCSI generic (sg) driver or IBM tape device driver.

For further details, see the IBM Redpaper, *Getting Started with zSeries Fibre Channel Protocol*, which is available at www.ibm.com/redbooks/

There are some differences for the zfcp drivers on Linux kernels 2.4.x and 2.6.x. See the zfcp driver release notes for further details.

Chapter 6. Configuring Storage Devices

This chapter contains concepts and procedures for configuring tape devices and removable file devices.

Use the following table to locate information needed to understand the concepts of Tivoli Storage Manager device support:

Concepts:
“Device Configuration Overview” on page 100
“Mixing Device Types in Libraries” on page 100
“Server Options that Affect Storage Operations” on page 102
“Defining Devices and Paths” on page 137
“Recovering from Device Changes on the SAN” on page 142

Use the following table to locate instructions for specific tasks:

Tasks:
“Configuring SCSI Libraries used by One Server” on page 102
“Configuring SCSI Libraries Shared Among Servers on a SAN” on page 107
“Configuring an IBM 3494 Library for Use by One Server” on page 112
“Sharing an IBM 3494 Library Among Servers” on page 117
“Sharing an IBM 3494 Library by Static Partitioning of Drives” on page 120
“Configuring ACSLS-Managed Libraries” on page 123
“Configuring Removable File Devices” on page 129
“Configuring Libraries Controlled by Media Manager Programs” on page 131
“Configuring Manually Mounted Devices” on page 133
“Configuring IBM Tivoli Storage Manager for LAN-free Data Movement” on page 135
“Validating your LAN-free Configuration” on page 136
“Configuring IBM Tivoli Storage Manager for NDMP Operations” on page 136

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see “Managing Servers with the New Administration Center” on page 36.

Note: Some of the tasks described in this chapter require an understanding of Tivoli Storage Manager storage objects. For an introduction to these storage objects, see “IBM Tivoli Storage Manager Storage Objects” on page 62.

© Copyright IBM Corp. 1993, 2005

Device Configuration Overview

Before Tivoli Storage Manager can use a removable media device, you must typically perform the steps described in this section. You can also configure devices using the device configuration wizard in the Administration Center. See "Managing Servers with the New Administration Center" on page 36 for more details.

1. Plan for the device.

   See "Planning for Server Storage" on page 70.
2. Attach the device to the server system, and ensure that the appropriate device driver is installed and configured. For more information on attaching devices, see Chapter 5, "Attaching Devices to the Server System," on page 91.

   For more information about which device drivers to use, see "Installing and Configuring Device Drivers" on page 93.
3. Define the device to Tivoli Storage Manager.

   Define the library, drive, paths, device class, storage pool, and storage volume objects. For an introduction to these objects, see "IBM Tivoli Storage Manager Storage Objects" on page 62 and "Configuring Devices" on page 80.
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 IBM Tivoli Storage Manager Stores Client Data" on page 5. For a description of the default policy, see "The Standard Policy" on page 359.

   Note: As an alternative to creating or modifying a Tivoli Storage Manager policy, you can place the new storage pool in the storage pool migration hierarchy by updating an already defined storage pool.
5. Prepare storage volumes for use by the device. At a minimum, you must label volumes for the device. For SCSI, 349X, and ACSLS libraries, add the volumes to the device's volume inventory by checking in the volumes. For more information, see Chapter 3, "Introducing Storage Devices," on page 61.

   Note: Each volume used by a server for any purpose must have a unique name. This applies to volumes that reside in different libraries, volumes used for storage pools, and volumes used for operations such as database backup or export.
6. Register clients to the domain associated with the policy that you defined or updated in the preceding step. For more information, see Chapter 13, "Implementing Policies for Client Data," on page 357.

Mixing Device Types in Libraries

Tivoli Storage Manager supports mixing different device types within a single automated library, as long as the library itself can distinguish among the different media for the different device types. Libraries with this capability are those models supplied from the manufacturer already containing mixed drives, or capable of supporting the addition of mixed drives. Check with the manufacturer, and also check the Tivoli Storage Manager Web site for specific libraries that have been tested on Tivoli Storage Manager with mixed device types.

For example, you can have Quantum SuperDLT drives, LTO Ultrium drives, and StorageTek 9940 drives in a single library defined to the Tivoli Storage Manager

server. For examples of how to set this up, see"Configuration with Multiple Drive Device Types" on page 105 and "Configuration with Multiple Drive Device Types" on page 114.

Mixing Different Media Generations in Libraries

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

An exception is that the server does support mixtures of the following types of LTO Ultrium drives and media:

- LTO Ultrium Generation 1 (LTO1) and LTO Ultrium Generation 2 (LTO2)
- LTO Ultrium Generation 2 (LTO2) with LTO Ultrium Generation 3 (LTO3)

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), first delete all existing Ultrium drive definitions and the paths associated with them. Then you can define the new Generation 2 (or Generation 3) drives and paths.

Note: LTO Ultrium Generation 3 drives can only read Generation 1 media. Therefore, if you are mixing Generation 1 and Generation 3 drives and media in a single library, you need to mark the Generation 1 media as read-only and all Generation 1 scratch volumes need to be checked out.

Mixed Media and Storage Pools

You cannot mix media formats in a storage pool. This includes LTO1, LTO2 and LTO3 formats. Each unique media format must be mapped to a separate storage pool through its own device class. Multiple storage pools and their device classes of different types can point to the same library which can support them as explained above.

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 neccessary to keep both LTO1 and LTO2 drives within the same library, separate storage pools for each must be used.

Server Options that Affect Storage Operations

Tivoli Storage Manager provides a number of options that you can specify in the server options file (dsmserv.opt) to configure certain server storage operations. Table 10 provides brief descriptions of these options. See the *Administrator's Reference* for details.

Table 10. Server Storage Options

Option	Description
3494SHARED	Enables sharing of an IBM 3494 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.
DRIVEACQUIRERETRY	Specifies how many times the server retries the acquisition of a drive in a library when there are no drives available after acquiring a mount point.
NOPREEMPT	Specifies whether the server allows certain operations to preempt other operations for access to volumes and devices. See "Preemption of Client or Server Operations" on page 467 for details.
RESOURCETIMEOUT	Specifies how long the server waits for a resource before canceling the pending acquisition of a resource. **Note:** For proper management of shared library resources, consider setting the RESOURCETIMEOUT option at the same time limit for all servers in a shared configuration. In the case of error recovery, Tivoli Storage Manager always defers to the longest time limit.
SEARCHMPQUEUE	Specifies the order in which the server satisfies requests in the mount queue.

Configuring SCSI Libraries used by One Server

In the following examples, automated SCSI libraries containing two drives are attached to the server system. The libraries are not shared with other Tivoli Storage Manager servers or with storage agents. The libraries are typically attached to the server system via SCSI cables.

Set up the Devices on the Server System

You must first set up the device on the server system. This involves the following tasks:

1. Set the appropriate SCSI ID for each drive and for the library or medium-changer.

2. Physically attach the devices to the server hardware.
3. Install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see "Attaching an Automated Library Device" on page 92 and "Installing and Configuring Device Drivers" on page 93.

Define the Devices to IBM Tivoli Storage Manager

There are two configurations described in this section:

- In the first configuration, both drives in the SCSI library are the same device type. You define one device class.
- In the second configuration, the drives are different device types. You define a device class for each drive device type.

  Drives with different device types are supported in a single library if you define a device class for each type of drive. If you are configuring this way, you must include the specific format for the drive's device type by using the FORMAT parameter with a value other than DRIVE.

Configuration with a Single Drive Device Type

In this example, the SCSI library contains two DLT tape drives.

1. Define a SCSI library named AUTODLTLIB. The library type is *SCSI* because the library is a SCSI-controlled automated library. Enter the following command:

   ```
   define library autodltlib libtype=scsi
   ```

   Note: If you have a SCSI library with a barcode reader and you would like to automatically label tapes before they are checked in, you can specify the following:

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

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

   ```
   define path server1 autodltlib srctype=server desttype=library
   device=/dev/tsmscsi/lb3
   ```

   The DEVICE parameter specifies the device driver's name for the library, which is the special file name.

   See "Defining Libraries" on page 137 and "SCSI Libraries" on page 63. For more information about paths, see "Defining Paths" on page 140.

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

   ```
   define drive autodltlib drive01
   define drive autodltlib drive02
   ```

   This example uses the default address for the drive's element address. The server obtains the element address from the drive itself at the time that the path is defined.

   The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can have the server obtain the element address from the drive itself at the time that the path is defined, or you can specify the element address when you define the drive.

   Depending on the capabilities of the library, the server may not be able to automatically detect the element address. In this case you must supply the element address when you define the drive. If you need the element address, check the Tivoli Storage Manager device driver support web site at

www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

See "Defining Drives" on page 138. For more information about paths, see "Defining Paths" on page 140.

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

```
define path server1 drive01 srctype=server desttype=drive
library=autodltlib device=/dev/tsmscsi/mt4
define path server1 drive02 srctype=server desttype=drive
library=autodltlib device=/dev/tsmscsi/mt5
```

   The DEVICE parameter specifies the device driver's name for the drive, which is the device special file name. For more about device special file names, see "Determining Device Special File Names" on page 94.

   If you did not include the element address when you defined the drive, the server now queries the library to obtain the default element address for the drive.

   For more information about paths, see "Defining Paths" on page 140.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. Use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical. For example, to classify two drives in the AUTODLTLIB library, use the following command to define a device class named AUTODLT_CLASS:

```
define devclass autodlt_class library=autodltlib devtype=dlt format=drive
```

   See "Defining and Updating Tape Device Classes" on page 198.

6. Verify your definitions by issuing the following commands:

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

   See "Requesting Information About Libraries" on page 186, "Requesting Information about Drives" on page 188, "Requesting Information about a Device Class" on page 211, and "Requesting Information About Paths" on page 193.

7. Define a storage pool named AUTODLT_POOL associated with the device class named AUTODLT_CLASS.

```
define stgpool autodlt_pool autodlt_class maxscratch=20
```

   Key choices:

   a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

   b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 258 and "How Collocation Affects Reclamation" on page 276.

   For more information, see "Defining or Updating Primary Storage Pools" on page 220.

Configuration with Multiple Drive Device Types

In this example, the library is a StorageTek L40 library that contains one DLT drive and one LTO Ultrium drive.

1. Define a SCSI library named MIXEDLIB. The library type is *SCSI* because the library is a SCSI-controlled automated library. Enter the following command:

   ```
   define library mixedlib libtype=scsi
   ```

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

   ```
   define path server1 mixedlib srctype=server desttype=library
   device=/dev/tsmscsi/lb3
   ```

   The DEVICE parameter specifies the device driver's name for the drive, which is the device special file name. For more about device special file names, see "Determining Device Special File Names" on page 94

   See "Defining Libraries" on page 137 and "SCSI Libraries" on page 63. For more information about paths, see "Defining Paths" on page 140.

3. Define the drives in the library:

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

   Both drives belong to the MIXEDLIB library.

   This example uses the default for the drive's element address. The server obtains the element address from the drive itself at the time that the path is defined.

   The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can have the server obtain the element address from the drive itself at the time that the path is defined, or you can specify the element address when you define the drive.

   Depending on the capabilities of the library, the server may not be able to automatically detect the element address. In this case you must supply the element address when you define the drive. If you need the element address, check the Tivoli Storage Manager device driver support web site at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

   See "Defining Drives" on page 138. For more information about paths, see "Defining Paths" on page 140.

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

   ```
   define path server1 dlt1 srctype=server desttype=drive
   library=mixedlib device=/dev/tsmscsi/mt4
   define path server1 lto1 srctype=server desttype=drive
   library=mixedlib device=/dev/tsmscsi/mt5
   ```

   The DEVICE parameter specifies the device driver's name for the drive, which is the device special file name. For more about device special file names, see "Determining Device Special File Names" on page 94.

   If you did not include the element address when you defined the drive, the server now queries the library to obtain the element address for the drive.

   For more information about paths, see "Defining Paths" on page 140.

5. Classify the drives according to type by defining Tivoli Storage Manager device classes, which specify the recording formats of the drives.

   Note: Do not use the DRIVE format, which is the default. Because the drives are different types, Tivoli Storage Manager uses the format specification to select a drive. The results of using the DRIVE format in a mixed media library are unpredictable.

```
define devclass dlt_class library=mixedlib devtype=dlt format=dlt40
define devclass lto_class library=mixedlib devtype=lto format=ultriumc
```

See “Defining and Updating Tape Device Classes” on page 198.

6. Verify your definitions by issuing the following commands:

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

See “Requesting Information About Libraries” on page 186, “Requesting Information about Drives” on page 188, “Requesting Information about a Device Class” on page 211, and “Requesting Information About Paths” on page 193.

7. Define storage pools associated with the device classes. For example:

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

Key choices:

a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping a Client’s Files Together: Collocation” on page 258 and “How Collocation Affects Reclamation” on page 276.

For more information, see “Defining or Updating Primary Storage Pools” on page 220.

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 you might have later.

Notes:

1. If you use the autolabel=yes parameter on the DEFINE LIBRARY command, you will not need to label tapes before you check them in.
2. If a volume has an entry in volume history, you cannot check it in as a scratch volume.
3. Tivoli Storage Manager only accepts tapes labeled with IBM standard labels. IBM standard labels are similar to ANSI Standard X3.27 labels except that the IBM standard labels are written in EBCDIC. For a list of IBM media sales contacts who can provide compatible tapes, go to the IBM Web site. If you are using non-IBM storage devices and media, consult your tape-cartridge distributor.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage

pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries but that are used by the same server.

The procedures for volume check-in and labeling are the same whether the library contains drives of a single device type, or drives of multiple device types.

To check in and label volumes, do the following:

1. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.
   - Check in volumes that are already labeled:
     ```
     checkin libvolume autodltlib search=yes status=scratch checklabel=barcode
     ```
   - Label and check in volumes:
     ```
     label libvolume autodltlib search=yes labelsource=barcode checkin=scratch
     ```
2. Depending on whether you use scratch volumes or private volumes, do one of the following:
   - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
   - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See "Defining Storage Pool Volumes" on page 229.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 399.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 232.

Configuring SCSI Libraries Shared Among Servers on a SAN

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

- Multiple Tivoli Storage Manager servers share storage devices.
- Tivoli Storage Manager client systems directly access storage devices, both tape libraries and disk storage, that are defined to a Tivoli Storage Manager server (LAN-free data movement). Storage agents installed and configured on the client systems perform the data movement. See "Configuring IBM Tivoli Storage Manager for LAN-free Data Movement" on page 135.

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

1. Set up server-to-server communications.
2. Set up the device on the server systems.
3. Set up the library on the Tivoli Storage Manager server that is going to act as the library manager. In the example used for this section, the library manager server is named ASTRO.

4. Set up the library on the Tivoli Storage Manager server that is going to act as the library client. In the example used for this section, the library client server is named JUDY.

Setting up Server Communications

Before Tivoli Storage Manager servers can share a storage device on a SAN, you must set up server communications. This requires configuring each server as you would for Enterprise Administration, which means you define the servers to each other using the cross-define function. See "Setting Up Communications Among Servers" on page 568 for details. For a discussion about the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see "Performing Operations with Shared Libraries" on page 182.

Note: Set up each server with a unique name.

Set up the Device on the Server Systems and the SAN

You must first set up the device on the server system. This involves the following tasks:

1. Set the appropriate SCSI ID for each drive and for the library or medium-changer.
2. Physically attach the devices to the SAN.
3. On each server system that will access the library and drives, install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see "Attaching an Automated Library Device" on page 92 and "Installing and Configuring Device Drivers" on page 93.

Setting up the Library Manager Server

Note: You can configure a SCSI library so that it contains all drives of the same device type or so that it contains drives of different device types. You can modify the procedure described for configuring a library for use by one server ("Configuration with Multiple Drive Device Types" on page 105) and use it for configuring a shared library.

You must set up the server that is the library manager before you set up servers that are the library clients.

Use the following procedure as an example of how to set up a server as a library manager. The server is named ASTRO.

1. Define a shared SCSI library named SANGROUP:

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

   This example uses the default for the library's serial number, which is to have the server obtain the serial number from the library itself at the time that the path is defined. Depending on the capabilities of the library, the server may not be able to automatically detect the serial number. In this case, the server will not record a serial number for the library, and will not be able to confirm the identity of the library when you define the path or when the server uses the library.
2. Define a path from the server to the library:

```
define path astro sangroup srctype=server desttype=library
 device=/dev/IBMchanger1
```

If you did not include the serial number when you defined the library, the server now queries the library to obtain the serial number information. If you did include the serial number when you defined the library, the server verifies what you defined and issues a message if there is a mismatch.

For more information about paths, see "Defining Paths" on page 140.

3. Define the drives in the library:

```
define drive sangroup drivea
define drive sangroup driveb
```

This example uses the default for the drive's serial number. The server obtains the serial number from the drive itself at the time that the path is defined. Depending on the capabilities of the drive, the server may not be able to automatically detect the serial number. In this case, the server will not record a serial number for the drive, and will not be able to confirm the identity of the drive when you define the path or when the server uses the drive.

This example also uses the default for the drive's element address, which is to have the server obtain the element address from the drive itself at the time that the path is defined.

The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can have the server obtain the element address from the drive itself at the time that the path is defined, or you can specify the element address when you define the drive.

Depending on the capabilities of the library, the server may not be able to automatically detect the element address. In this case you must supply the element address when you define the drive. If you need the element address, check the Tivoli Storage Manager device driver support web site at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

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

```
define path astro drivea srctype=server desttype=drive
 library=sangroup device=/dev/IBMtape4
define path astro driveb srctype=server desttype=drive
 library=sangroup device=/dev/IBMtape5
```

If you did not include the serial number or element address when you defined the drive, the server now queries the drive or the library to obtain this information.

For more information about paths, see "Defining Paths" on page 140.

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

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

6. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.
   - Check in volumes that are already labeled:

     ```
     checkin libvolume sangroup search=yes status=scratch checklabel=barcode
     ```

   - Label and check in volumes:

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

7. Set up any required storage pools for the shared library with a maximum of 50 scratch volumes.

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

Setting up the Library Client Servers

You must set up the server that is the library manager server before you set up the servers that are the library clients.

Use the following sample procedure for each Tivoli Storage Manager server that will be a library client. The library client server is named JUDY. With the exception of one step, perform the procedure from the library client servers.

1. Define the server that is the library manager:

   ```
   define server astro serverpassword=secret hladdress=9.115.3.45 lladdress=1580
    crossdefine=yes
   ```

2. Define the shared library named SANGROUP, and identify the library manager server's name as the primary library manager. Ensure that the library name is the same as the library name on the library manager:

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

3. *Perform this step from the library manager*. Define a path from the library client server to each drive that the library client server will be allowed to access. The device name should reflect the way the *library client* system sees the device. There must be a path defined from the library manager to each drive in order for the library client to use the drive:

   ```
   define path judy drivea srctype=server desttype=drive
    library=sangroup device=/dev/IBMtape6
   define path judy driveb srctype=server desttype=drive
    library=sangroup device=/dev/IBMtape7
   ```

   For more information about paths, see "Defining Paths" on page 140.

4. *Return to the library client for the remaining steps.* Define all the device classes that are associated with the shared library.

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

   Set the parameters for the device class the same on the library client as on the library manager. Making the device class names the same on both servers is also a good practice, but is not required.

5. Define the storage pool, BACKTAPE, that will use the shared library.

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

6. Repeat this procedure to define additional servers as library clients.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 399.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 232.

Migrating DEFINE DRIVE and DEFINE PATH Definitions to the Library Manager Server

The DEFINE PATH and DEFINE DRIVE commands are not allowed on the library client for Tivoli Storage Manager versions 5.1 or higher. These definitions are now configured on the library manager server. Definitions from Tivoli Storage Manager versions 4.2 and lower should be migrated over to the library manager server by following these steps:

1. Retrieve the path information from the library client through the device configuration (DEVCONFIG) file or the QUERY PATH command.

2. Use the data gathered in step 1 to issue the DEFINE PATH commands on the library manager server.
3. Delete the path definitions on the library client with DELETE PATH.
4. Delete the drive definitions on the library client with DELETE DRIVE.

Configuring IBM 3494 Libraries

One or more Tivoli Storage Manager servers can use a single IBM 3494 library. See the following sections:

- "Configuring an IBM 3494 Library for Use by One Server" on page 112
- "Sharing an IBM 3494 Library Among Servers" on page 117
- "Migrating an IBM 3494 Library to Control by a Library Manager" on page 119
- "Sharing an IBM 3494 Library by Static Partitioning of Drives" on page 120

See also "Categories in an IBM 3494 Library."

Categories in an IBM 3494 Library

The library manager built into the IBM 3494 library tracks the category number of each volume in the library. A single category number identifies all volumes used for the same purpose or application. These category numbers are useful when multiple systems share the resources of a single library.

Attention: If other systems or other Tivoli Storage Manager servers connect to the same 3494 library, each must use a unique set of category numbers. Otherwise, two or more systems may try to use the same volume, and cause corruption or loss of data.

Typically, a software application that uses a 3494 library uses volumes in one or more categories that are reserved for that application. To avoid loss of data, each application sharing the library must have unique categories. When you define a 3494 library to the server, you can use the PRIVATECATEGORY and SCRATCHCATEGORY parameters to specify the category numbers for private and scratch Tivoli Storage Manager volumes in that library. If the volumes are IBM 3592 WORM (write once, read many) volumes, you can use the WORMSCRATCHCATEGORY parameter to specify category numbers for scratch WORM volumes in the library. See "IBM Tivoli Storage Manager Volumes" on page 68 for more information on private, scratch, and scratch WORM volumes.

When a volume is first inserted into the library, either manually or automatically at the convenience I/O station, the volume is assigned to the insert category (X'FF00'). A software application such as Tivoli Storage Manager can contact the library manager to change a volume's category number. For Tivoli Storage Manager, you use the CHECKIN LIBVOLUME command (see "Checking New Volumes into a Library" on page 171).

When installed on the Linux operating system, a Tivoli Storage Manager server supports only 3590 and 3592 tape drives in an IBM 3494 library. The server reserves two different categories for each 3494 library object. The categories are private and scratch.

Note: A 349X library object will now only contain one device type (3590 or 3592) of drives.Thus, if you have 3590s and 3592s in your 349X library, you need

to define two library objects: one for your 3590 drives and one for your 3592 drives. Each of these library objects will have the same device parameter when defining their paths.

When you define a 3494 library, you can specify the category numbers for volumes that the server owns in that library by using the PRIVATECATEGORY, SCRATCHCATEGORY, and if the volumes are IBM 3592 WORM volumes, the WORMSCRATCHCATEGORY parameters. For example:

```
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 may be acceptable in most cases. However, if you connect other systems or Tivoli Storage Manager servers to a single 3494 library, ensure that each uses unique category numbers. Otherwise, two or more systems may try to use the same volume, and cause a corruption or loss of data.

For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see "Performing Operations with Shared Libraries" on page 182.

Configuring an IBM 3494 Library for Use by One Server

In the following example, an IBM 3494 library containing two drives is configured for use by one Tivoli Storage Manager server.

Set up the Device on the Server System

You must first set up the IBM 3494 library on the server system. This involves the following tasks:

1. Set the symbolic name for the library in the configuration file for the library device driver (/etc/ibmatl.conf). This procedure is described in *IBM TotalStorage Tape Device Drivers Installation and User's Guide*.
2. Physically attach the devices to the server hardware or the SAN.
3. Install and configure the appropriate device drivers for the devices on the server that will use the library and drives.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see "Attaching an Automated Library Device" on page 92 and "Installing and Configuring Device Drivers" on page 93.

Define the Devices to IBM Tivoli Storage Manager

There are two configurations described in this section:

- In the first configuration, both drives in the library are the same device type.
- In the second configuration, the drives are different device types.

  Drives with different device types are supported in a single physical library if you define one library to Tivoli Storage Manager for each type of drive. If you

have two device types (such as 3590E and 3590H), you define two libraries. Then you define drives and device classes for each library. In each device class definition, you can use the FORMAT parameter with a value of DRIVE, if you choose.

Configuration with a Single Drive Device Type

In this example, the 3494 library contains two IBM 3590 tape drives.

1. Define a 3494 library named 3494LIB:

   ```
   define library 3494lib libtype=349x
   ```

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

   ```
   define path server1 3494lib srctype=server desttype=library
    device=library1
   ```

   The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

   See “Defining Libraries” on page 137 and “SCSI Libraries” on page 63. For more information about paths, see “Defining Paths” on page 140.

3. Define the drives in the library:

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

   Both drives belong to the 3494LIB library.

   See “Defining Drives” on page 138.

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

   ```
   define path server1 drive01 srctype=server desttype=drive
    library=3494lib device=/dev/IBMtape0
   define path server1 drive02 srctype=server desttype=drive
    library=3494lib device=/dev/IBMtape1
   ```

   The DEVICE parameter gives the device special file name for the drive. For more about device names, see “Determining Device Special File Names” on page 94. For more information about paths, see “Defining Paths” on page 140.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. For example, for the two 3590 drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

   ```
   define devclass 3494_class library=3494lib devtype=3590 format=drive
   ```

   This example uses FORMAT=DRIVE as the recording format because both drives associated with the device class use the same recording format; both are 3590 drives. If instead one drive is a 3590 and one is a 3590E, you need to use specific recording formats when defining the device classes. See “Configuration with Multiple Drive Device Types” on page 114.

   See also “Defining and Updating Tape Device Classes” on page 198.

6. Verify your definitions by issuing the following commands:

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

   See “Requesting Information About Libraries” on page 186, “Requesting Information about Drives” on page 188, “Requesting Information about a Device Class” on page 211, and “Requesting Information About Paths” on page 193.

7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS.

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

Key choices:

a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

b. Collocation by group is the default setting for primary storage pools. The default for copy storage pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 258 and "How Collocation Affects Reclamation" on page 276.

For more information, see "Defining or Updating Primary Storage Pools" on page 220.

Configuration with Multiple Drive Device Types

In this example, the 3494 library contains two IBM 3590E tape drives and two IBM 3590H tape drives.

1. Define *two* libraries, one for each type of drive. For example, to define 3590ELIB and 3590HLIB enter the following commands:

   ```
   define library 3590elib libtype=349x scratchcategory=301 privatecategory=300
   define library 3590hlib libtype=349x scratchcategory=401 privatecategory=400
   ```

   See "Defining Libraries" on page 137.

   Note: Specify scratch and private categories explicitly. If you accept the category defaults for both library definitions, different types of media will be assigned to the same categories.

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

   ```
   define path server1 3590elib srctype=server desttype=library device=library1
   define path server1 3590hlib srctype=server desttype=library device=library1
   ```

   The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

   For more information about paths, see "Defining Paths" on page 140.

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

   - Define the 3590E drives to 3590ELIB.

     ```
     define drive 3590elib 3590e_drive1
     define drive 3590elib 3590e_drive2
     ```

   - Define the 3590H drives to 3590HLIB.

     ```
     define drive 3590hlib 3590h_drive3
     define drive 3590hlib 3590h_drive4
     ```

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

   See "Defining Drives" on page 138.

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

   - For the 3590E drives:

```
define path server1 3590e_drive1 srctype=server desttype=drive
 library=3590elib device=/IBMtape0
define path server1 3590e_drive2 srctype=server desttype=drive
 library=3590elib device=/IBMtape1
```

- For the 3590H drives:

```
define path server1 3590h_drive3 srctype=server desttype=drive
 library=3590hlib device=/IBMtape2
define path server1 3590h_drive4 srctype=server desttype=drive
 library=3590hlib device=/IBMtape30
```

The DEVICE parameter gives the device special file name for the drive. For more about device names, see "Determining Device Special File Names" on page 94.. For more information about paths, see "Defining Paths" on page 140.

5. Classify the drives according to type by defining Tivoli Storage Manager device classes, which specify the recording formats of the drives. Because there are separate libraries, you can enter a specific recording format, for example 3590H, or you can enter DRIVE.

```
define devclass 3590e_class library=3590elib devtype=3590 format=3590e

define devclass 3590h_class library=3590hlib devtype=3590 format=3590h
```

See "Defining and Updating Tape Device Classes" on page 198.

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

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

See "Requesting Information About Libraries" on page 186, "Requesting Information about Drives" on page 188, "Requesting Information about a Device Class" on page 211, and "Requesting Information About Paths" on page 193.

7. Create the storage pools to use the devices in the device classes you just defined. For example, define a storage pool named 3590EPOOL associated with the device class 3490E_CLASS, and 3590HPOOL associated with the device class 3590H_CLASS:

```
define stgpool 3590epool 3590e_class maxscratch=20

define stgpool 3590hpool 3590h_class maxscratch=20
```

Key choices:

a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

b. Collocation by group is the default setting for primary storage pools. The default for copy storage pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 258 and "How Collocation Affects Reclamation" on page 276.

For more information, see "Defining or Updating Primary Storage Pools" on page 220.

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 you might have later.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries.

The procedures for volume check-in and labeling are the same whether the library contains drives of a single device type, or drives of multiple device types.

Note: If your library has drives of multiple device types, you defined *two* libraries to the Tivoli Storage Manager server in the procedure in "Configuration with Multiple Drive Device Types" on page 114. The two Tivoli Storage Manager libraries represent the *one* physical library. The check-in process finds all available volumes that are not already checked in. You must check in media *separately* to each defined library. Ensure that you check in volumes to the correct Tivoli Storage Manager library.

Do the following:

1. Check in the library inventory. The following shows two examples.
   - Check in volumes that are already labeled:

     ```
     checkin libvolume 3494lib search=yes status=scratch checklabel=no
     ```

   - Label and check in volumes:

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

2. Depending on whether you use scratch volumes or private volumes, do one of the following:
   - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
   - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See "Defining Storage Pool Volumes" on page 229.

For more information about checking in volumes, see "Checking New Volumes into a Library" on page 171.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 399.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 232.

Sharing an IBM 3494 Library Among Servers

Sharing an IBM 3494 library requires one of the following environments:

- The library must be on a SAN.
- Through the use of the dual ports on 3590 drives in the library, the drives and the library are connected to two systems on which Tivoli Storage Manager servers run.

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

1. Set up server-to-server communications.
2. Set up the device on the server systems.
3. Set up the library on the library manager server. In the following example, the library manager server is named MANAGER.
4. Set up the library on the library client server. In the following example, the library client server is named CLIENT.

See "Categories in an IBM 3494 Library" on page 111 for additional information about configuring 3494 libraries.

Setting up Server Communications

Before Tivoli Storage Manager servers can share a storage device over a SAN, you must set up server communications. This requires configuring each server as you would for Enterprise Administration, which means you define the servers to each other using the cross-define function. See "Setting Up Communications Among Servers" on page 568 for details. For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see "Performing Operations with Shared Libraries" on page 182.

Set up the Device on the Server System and the SAN

You must first set up the device on the server system. This involves the following tasks:

1. Set the symbolic name for the library in the configuration file for the library device driver. This procedure is described in *IBM TotalStorage Tape Device Drivers Installation and User's Guide*.
2. Physically attach the devices to the SAN or to the server hardware.
3. On each server system that will access the library and drives, install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see "Attaching an Automated Library Device" on page 92 and "Installing and Configuring Device Drivers" on page 93.

Note: You can configure a 3494 library so that it contains drives all of the same device type, or so that it contains drives of multiple device types. The procedure is similar to the one described for a LAN ("Configuration with Multiple Drive Device Types" on page 114).

Setting up the Library Manager Server

Use the following procedure as an example of how to set up a Tivoli Storage Manager server as a library manager named MANAGER:

1. Define a 3494 library named 3494SAN:

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

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

   ```
   define path manager 3494san srctype=server desttype=library
    device=library1
   ```

   The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

   For more information about paths, see "Defining Paths" on page 140.

3. Define the drives in the library:

   ```
   define drive 3494san drivea
   define drive 3494san driveb
   ```

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

   ```
   define path manager drivea srctype=server desttype=drive library=3494san
    device=/dev/IBMtape0
   define path manager driveb srctype=server desttype=drive library=3494san
    device=/dev/IBMtape1
   ```

   For more information about paths, see "Defining Paths" on page 140.

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

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

6. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.

   To check in volumes that are already labeled, use the following command:

   ```
   checkin libvolume 3494san search=yes status=scratch checklabel=no
   ```

   To label and check in the volumes, use the following command:

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

7. Set any required storage pools for the shared library with a maximum of 50 scratch volumes.

   ```
   define stgpool 3494_sanpool tape maxscratch=50
   ```

Setting up the Library Client Servers

Use the following sample procedure for each Tivoli Storage Manager server that will be a library client server.

1. Define the server that is the library manager:

   ```
   define server manager serverpassword=secret hladdress=9.115.3.45 lladdress=1580
    crossdefine=yes
   ```

2. Define a shared library named 3494SAN, and identify the library manager:

   ```
   define library 3494san libtype=shared primarylibmanager=manager
   ```

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

3. *Perform this step from the library manager*. Define a path from the library client server to each drive that the library client server will be allowed to access. The device name should reflect the way the *library client* system sees the device. There must be a path defined from the library manager to each drive in order for the library client to use the drive:

```
define path client drivea srctype=server desttype=drive
 library=3494san device=/dev/IBMtape0
define path client driveb srctype=server desttype=drive
 library=3494san device=/dev/IBMtape1
```

For more information about paths, see "Defining Paths" on page 140.

4. *Return to the library client for the remaining steps.* Define all the device classes that are associated with the shared library.

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

   Set the parameters for the device class the same on the library client as on the library manager. Making the device class names the same on both servers is also a good practice, but is not required.

5. Define the storage pool, BACKTAPE, that will use the shared library.

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

6. Repeat this procedure to define additional servers as library clients. For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see "Performing Operations with Shared Libraries" on page 182.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 399.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 232.

Migrating an IBM 3494 Library to Control by a Library Manager

If you have been sharing an IBM 3494 library among Tivoli Storage Manager servers by using the 3494SHARED option in the dsmserv.opt file, you can migrate to sharing the library by using a library manager and library clients. To help ensure a smoother migration and to ensure that all tape volumes that are being used by the servers get associated with the correct servers, perform the following migration procedure.

1. Do the following on *each* server that is sharing the 3494 library:
   a. Update the storage pools using the UPDATE STGPOOL command. Set the value for the HIGHMIG and LOWMIG parameters to 100%.
   b. Stop the server by issuing the HALT command.
   c. Edit the dsmserv.opt file and make the following changes:
      1) Comment out the 3494SHARED YES option line
      2) Activate the DISABLESCHEDS YES option line if it is not active
      3) Activate the EXPINTERVAL X option line if it is not active and change its value to 0, as follows:

         ```
         EXPINTERVAL 0
         ```

   d. Start the server.
   e. Enter the following Tivoli Storage Manager command:

      ```
      disable sessions
      ```

2. Set up the library manager on the Tivoli Storage Manager server of your choice (see "Setting up Server Communications" on page 108 and"Setting up the Library Manager Server" on page 108).

3. Do the following on the remaining Tivoli Storage Manager servers (the library clients):
   a. Save the volume history file.
   b. Check out all the volumes in the library inventory. Use the CHECKOUT LIBVOLUME command with REMOVE=NO.
   c. Follow the library client setup procedure ("Setting up the Library Client Servers" on page 118).
4. Do the following on the library manager server:
   a. Check in each library client's volumes. Use the CHECKIN LIBVOLUME command with the following parameter settings:
      - STATUS=PRIVATE
      - OWNER=<*library client name*>

      Note: You can use the saved volume history files from the library clients as a guide.
   b. Check in any remaining volumes as scratch volumes. Use the CHECKIN LIBVOLUME command with STATUS=SCRATCH.
5. Halt all the servers.
6. Edit the dsmserv.opt file and comment out the following lines in the file:
   ```
   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.

Tivoli Storage Manager uses the capability of the 3494 library manager, which allows you to partition a library between multiple Tivoli Storage Manager servers. Library partitioning differs from library sharing on a SAN in that with partitioning, there are no Tivoli Storage Manager library managers or library clients.

When you partition a library on a LAN, each server has its own access to the same library. For each server, you define a library with tape volume categories unique to that server. Each drive that resides in the library is defined to only one server. Each server can then access only those drives it has been assigned. As a result, library partitioning does not allow dynamic sharing of drives or tape volumes because they are pre-assigned to different servers using different names and category codes.

In the following example, an IBM 3494 library containing four drives is attached to a Tivoli Storage Manager server named ASTRO and to another Tivoli Storage Manager server named JUDY.

Note: Tivoli Storage Manager can also share the drives in a 3494 library with other servers by enabling the 3494SHARED server option. When this option is enabled, you can define all of the drives in a 3494 library to multiple servers, if there are SCSI connections from all drives to the systems on which the servers are running. This type of configuration is not

recommended, however, because when this type of sharing takes place there is a risk of contention between servers for drive usage, and operations can fail.

Set up the Device on the Servers

You must first set up the 3494 library on the server system. This involves the following tasks:

1. Set the symbolic name for the library in the configuration file for the library device driver. This procedure is described in *IBM TotalStorage Tape Device Drivers Installation and User's Guide*.
2. Physically attach the devices to the server hardware.
3. On each server system that will access the library and drives, install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see "Attaching an Automated Library Device" on page 92 and "Installing and Configuring Device Drivers" on page 93.

Define the Devices to IBM Tivoli Storage Manager ASTRO

1. Define the 3494 library named 3494LIB:

   ```
   define library 3494lib libtype=349x privatecategory=400 scratchcategory=600
   ```

   The PRIVATECATEGORY and SCRATCHCATEGORY are set differently from the default settings. See "Categories in an IBM 3494 Library" on page 111.
2. Define the path from the server, ASTRO, to the library:

   ```
   define path astro 3494lib srctype=server desttype=library
    device=library1
   ```

   The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

   See "Defining Libraries" on page 137 and "SCSI Libraries" on page 63. For more information about paths, see "Defining Paths" on page 140.
3. Define the drives that are partitioned to server ASTRO:

   ```
   define drive 3494lib drive1
   define drive 3494lib drive2
   ```
4. Define the path from the server, ASTRO, to each of the drives:

   ```
   define path astro drive1 srctype=server desttype=drive library=3494lib
    device=/dev/IBMtape0
   define path astro drive2 srctype=server desttype=drive library=3494lib
    device=/dev/IBMtape1
   ```

   The DEVICE parameter gives the device special file name for the drive. For more about device names, see "Determining Device Special File Names" on page 94. For more information about paths, see "Defining Paths" on page 140.
5. Classify drives according to type by defining Tivoli Storage Manager device classes. For example, to classify the two drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

   ```
   define devclass 3494_class library=3494lib devtype=3590 format=drive
   ```

   This example uses FORMAT=DRIVE as the recording format because both drives associated with the device class use the same recording format; both are 3590 drives. If instead one drive is a 3590 and one is a 3590E, you need to use specific recording formats when defining the device classes. See "Configuration with Multiple Drive Device Types" on page 114.

See “Defining and Updating Tape Device Classes” on page 198.

6. Verify your definitions by issuing the following commands:

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

See “Requesting Information About Libraries” on page 186, “Requesting Information about Drives” on page 188, “Requesting Information about a Device Class” on page 211, and “Requesting Information About Paths” on page 193.

7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS.

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

Key choices:

a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

b. Collocation by group is the default setting for primary storage pools. The default for copy storage pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping a Client’s Files Together: Collocation” on page 258 and “How Collocation Affects Reclamation” on page 276.

For more information, see “Defining or Updating Primary Storage Pools” on page 220.

Define the Devices to Tivoli Storage Manager JUDY

1. Define the 3494 library named 3494LIB:

```
define library 3494lib libtype=3494 privatecategory=112 scratchcategory=300
```

The PRIVATECATEGORY and SCRATCHCATEGORY are defined differently than the first server’s definition. See “Categories in an IBM 3494 Library” on page 111.

2. Define the path from the server, JUDY, to the library:

```
define path judy 3494lib srctype=server desttype=library
 device=library1
```

The DEVICE parameter specifies the symbolic name for the library, as defined in the configuration file for the library device driver (/etc/ibmatl.conf).

See “Defining Libraries” on page 137 and “SCSI Libraries” on page 63. For more information about paths, see “Defining Paths” on page 140.

3. Define the drives that are partitioned to server JUDY:

```
define drive 3494lib drive3
define drive 3494lib drive4
```

4. Define the path from the server, JUDY, to each of the drives:

```
define path judy drive3 srctype=server desttype=drive library=3494lib
 device=/dev/IBMtape2
define path judy drive4 srctype=server desttype=drive library=3494lib
 device=/dev/IBMtape3
```

The DEVICE parameter gives the device special file name for the drive. For more about device names, see "Determining Device Special File Names" on page 94. For more information about paths, see "Defining Paths" on page 140.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. For example, to classify the two drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

   ```
   define devclass 3494_class library=3494lib devtype=3590 format=drive
   ```

   This example uses FORMAT=DRIVE as the recording format because both drives associated with the device class use the same recording format; both are 3590 drives. If instead one drive is a 3590 and one is a 3590E, you need to use specific recording formats when defining the device classes. See "Configuration with Multiple Drive Device Types" on page 114.

   See "Defining and Updating Tape Device Classes" on page 198.

6. Verify your definitions by issuing the following commands:

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

   See "Requesting Information About Libraries" on page 186, "Requesting Information about Drives" on page 188, and "Requesting Information about a Device Class" on page 211.

7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS.

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

   Key choices:

   a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

   b. Collocation by group is the default setting for primary storage pools. The default for copy storage pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 258 and "How Collocation Affects Reclamation" on page 276.

   For more information, see "Defining or Updating Primary Storage Pools" on page 220.

Configuring ACSLS-Managed Libraries

Tivoli Storage Manager supports tape libraries controlled by StorageTek Automated Cartridge System Library Software (ACSLS). The ACSLS library server manages the physical aspects of tape cartridge storage and retrieval. The ACSLS client application communicates with the ACSLS library server to access tape cartridges

in an automated library. Tivoli Storage Manager is one of the applications that gains access to tape cartridges by interacting with ACSLS through its client, which is known as the control path. The Tivoli Storage Manager server reads and writes data on tape cartridges by interacting directly with tape drives through the data path. The control path and the data path are two different paths.

The ACSLS client daemon must be initialized before starting the server. See the script rc.acs_ssi in the server's installation directory for the client daemon invocation. For detailed installation, configuration, and system administration of ACSLS, refer to the appropriate StorageTek documentation.

Set up the Device on the Server System

The library is attached to the ACSLS server, and the drives are attached to the Tivoli Storage Manager server. The ACSLS server and the Tivoli Storage Manager server must be on different systems. Refer to the ACSLS installation documentation for details about how to set up the library.

Define the Devices to IBM Tivoli Storage Manager

There are two configurations described in this section:

- In the first configuration, both drives in the ACSLS library are the same device type.
- In the second configuration, the drives are different device types.

  Drives with different device types are supported in a single physical library if you define one library to Tivoli Storage Manager for each type of drive. If you have two device types (such as 9840 and 9940), you define two libraries. Then you define drives and device classes for each library. In each device class definition, you can use the FORMAT parameter with a value of DRIVE, if you choose.

Configuration with a Single Drive Device Type

1. Define an ACSLS library named ACSLIB:

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

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

   ```
   define drive acslib drive01 acsdrvid=1,2,3,4
   define drive acslib drive02 acsdrvid=1,2,3,5
   ```

   The ACSDRVID parameter specifies the ID of the drive that is being accessed. The drive ID is a set of numbers that indicate the physical location of a drive within an ACSLS library. This drive ID must be specified as *a*, *l*, *p*, *d*, where *a* is the ACSID, *l* is the LSM (library storage module), *p* is the panel number, and *d* is the drive ID. The server needs the drive ID to connect the physical location of the drive to the drive's SCSI address. See the StorageTek documentation for details.

   See "Defining Drives" on page 138.
3. Define a path from the server to each drive:
4. Classify drives according to type by defining Tivoli Storage Manager device classes. For example, to classify the two drives in the ACSLIB library, use the following command to define a device class named ACS_CLASS:

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

This example uses FORMAT=DRIVE as the recording format because both drives associated with the device class use the same recording format; for example, both are 9940 drives. If instead one drive is a 9840 and one is a 9940, you need to use specific recording formats when defining the device classes. See "Configuration with Multiple Drive Device Types."

See "Defining and Updating Tape Device Classes" on page 198.

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

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

See "Requesting Information About Libraries" on page 186, "Requesting Information about Drives" on page 188, "Requesting Information about a Device Class" on page 211, and "Requesting Information About Paths" on page 193.

6. Create the storage pool to use the devices in the device class you just defined. For example, define a storage pool named ACS_POOL associated with the device class ACS_CLASS:

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

Key choices:

a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

b. Collocation by group is the default setting for primary storage pools. The default for copy storage pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 258 and "How Collocation Affects Reclamation" on page 276.

For more information, see "Defining or Updating Primary Storage Pools" on page 220.

Configuration with Multiple Drive Device Types

The following example shows how to set up and ACSLS library with a mix of two 9840 drives and two 9940 drives.

1. Define *two* ACSLS libraries that use the same ACSID. For example to define 9840LIB and 9940LIB, enter the following commands:

```
define library 9840lib libtype=acsls acsid=1
define library 9940lib libtype=acsls acsid=1
```

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

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

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

- Define the 9840 drives to 9840LIB.

```
define drive 9840lib 9840_drive1 acsdrvid=1,2,3,1
define drive 9840lib 9840_drive2 acsdrvid=1,2,3,2
```

- Define the 9940 drives to 9940LIB.

```
define drive 9940lib 9940_drive3 acsdrvid=1,2,3,3
define drive 9940lib 9940_drive4 acsdrvid=1,2,3,4
```

The ACSDRVID parameter specifies the ID of the drive that is being accessed. The drive ID is a set of numbers that indicate the physical location of a drive within an ACSLS library. This drive ID must be specified as *a*, *l*, *p*, *d*, where *a* is the ACSID, *l* is the LSM (library storage module), *p* is the panel number, and *d* is the drive ID. The server needs the drive ID to connect the physical location of the drive to the drive's SCSI address. See the StorageTek documentation for details.

See "Defining Drives" on page 138.

3. Define a path from the server to each drive. Ensure that you specify the correct library.
   - For the 9840 drives:
   - For the 9940 drives:
4. Classify the drives according to type by defining Tivoli Storage Manager device classes, which specify the recording formats of the drives. Because there are separate libraries, you can enter a specific recording format, for example 9840, or you can enter DRIVE. For example, to classify the drives in the two libraries, use the following commands to define one device class for each type of drive:

```
define devclass 9840_class library=9840lib devtype=ecartridge format=9840

define devclass 9940_class library=9940lib devtype=ecartridge format=9940
```

See "Defining and Updating Tape Device Classes" on page 198.

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

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

See "Requesting Information About Libraries" on page 186, "Requesting Information about Drives" on page 188, "Requesting Information about a Device Class" on page 211, and "Requesting Information About Paths" on page 193.

6. Create the storage pools to use the devices in the device classes that you just defined. For example, define storage pools named 9840_POOL associated with the device class 9840_CLASS and 9940_POOL associated with the device class 9940_CLASS:

```
define stgpool 9840_pool 9840_class maxscratch=20

define stgpool 9940_pool 9940_class maxscratch=20
```

Key choices:

a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

b. Collocation by group is the default setting for primary storage pools. The default for copy storage pools is disablement of collocation. Collocation is a process by which the server attempts to keep all files belonging to a group

of client nodes, a single client node, or a client file space on a minimal number of volumes. If collocation is disabled for a storage pool and clients begin storing data, you cannot easily change the data in the pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 258 and "How Collocation Affects Reclamation" on page 276.

For more information, see "Defining or Updating Primary Storage Pools" on page 220.

Setting up the Library Manager Server

You can configure an ACSLS library so that it contains all drives of the same device type or so that it contains drives of different device types. You can modify the procedure described for configuring a library for use by one server ("Configuration with Multiple Drive Device Types" on page 105) and use it for configuring a shared library. You must set up the server that is the library manager before you set up servers that are the library clients.

When upgrading multiple servers participating in library sharing to Version 5.3, upgrade all the servers at once, or do the library manager servers and then the library client servers. A Version 5.3 library manager server is compatible with downlevel library clients. However, 5.3 library clients are not compatible with a library manager server that is at a level earlier than 5.3.

Note: An exception to this rule is when a fix or product enhancement requires concurrent code changes to the server, storage agent, and library client.

The following procedure is an example of how to set up a server named GLENCOE as a library manager. Perform the following steps on the server GLENCOE:

1. Define a server as a library client server named WALLACE:

   ```
   define server wallace HLA=x.x.x.x LLA=y
   ```

2. Define a shared ACSLS library named MACGREGOR:

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

   This example used an ACSLS managed library configured as acs 1.

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

   ```
   define devclass tape library=macgregor devtype=9840
   ```

4. Define the drives in the library MACGREGOR:

   ```
   define drive macgregor drivea acsdrvid=1,0,1,0
   define drive macgregor driveb acsdrvid=1,0,1,1
   ```

   This example uses the acsdrvid value, which specifies the ID of the drive that is being accessed in an ACSLS library. The drive ID is a set of numbers that indicates the physical location of a drive within an ACSLS library. This drive ID must be specified as a,l,p,d, where a is the ACSID, l is the LSM (library storage module), p is the panel number, and d is the drive ID. The server needs the drive ID to connect the physical location of the drive to the drive's SCSI address. See the StorageTek documentation for details.

5. Define a path from the server GLENCOE to each drive in the library MACGREGOR:

   ```
   define path glencoe drivea srctype=server desttype=drive
    library=macgregor device=/dev/IBMtape4
   define path glencoe driveb srctype=server desttype=drive
    library=macgregor device=/dev/IBMtape5
   ```

For more information about paths, see “Defining Paths” on page 140.

Setting up the Library Client Servers

You must set up the server that is the library manager server before you set up the servers that are the library clients.

Use the following sample procedure for each Tivoli Storage Manager server that will be a library client. The library client server is named WALLACE. With the exception of one step, perform the procedure from the library client server.

1. Define the server that is the library manager:

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

2. Define the shared library named MACGREGOR, and identify the library manager server’s name as the primary library manager. Ensure that the library name is the same as the library name on the library manager:

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

3. *Perform this step from the library manager*. Define a path from the library client server to each drive that the library client server will be allowed to access. The device name should reflect the way the *library client* system sees the device. There must be a path defined from the library manager to each drive in order for the library client to use the drive:

   ```
   define path wallace drivea srctype=server desttype=drive
    library=macgregor device=/dev/IBMtape6
   define path wallace driveb srctype=server desttype=drive
    library=macgregor device=/dev/IBMtape7
   ```

   For more information about paths, see “Defining Paths” on page 140.

4. *Return to the library client for the remaining steps.* Define all the device classes that are associated with the shared library.

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

   Set the parameters for the device class the same on the library client as on the library manager. Making the device class names the same on both servers is also a good practice, but is not required.

5. Define the storage pool, LOCHNESS, that will use the shared library.

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

6. Repeat this procedure to define additional servers as library clients.

Check In and Label Library Volumes

Ensure that enough volumes are available to the server in the library. You must label volumes that do not already have a standard label. Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries.

Attention: If your library has drives of multiple device types, you defined *two* libraries to the Tivoli Storage Manager server in the procedure in "Configuration with Multiple Drive Device Types" on page 125. The two Tivoli Storage Manager libraries represent the *one* physical library. The check-in process finds all available volumes that are not already checked in. You must check in media *separately* to each defined library. Ensure that you check in volumes to the correct Tivoli Storage Manager library.

1. Check in the library inventory. The following shows examples for libraries with a single drive device type and with multiple drive device types.
   - Check in volumes that are already labeled:

     ```
     checkin libvolume acslib search=yes status=scratch checklabel=no
     ```

   - Label and check in volumes:

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

2. Depending on whether you use scratch volumes or private volumes, do one of the following:
   - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
   - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See "Defining Storage Pool Volumes" on page 229.

For more information about checking in volumes, see "Checking New Volumes into a Library" on page 171.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 399.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 232.

Configuring Removable File Devices

Removable file support includes rewritable CDs.

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 third-party software. 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.

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 650MB.

Example of Removable File Support

Use these steps as an example of Tivoli Storage Manager REMOVABLEFILE support. This example takes an export object and moves it from one server to another by using a CD.

Server A

1. Define a device class with a device type of FILE.

   ```
   define devclass file devtype=file directory=/home/user1
   ```

2. Export the node. This command results in a file name */home/user1/CDR03* that contains the export data for node USER1.

   ```
   export node user1 filedata=all devclass=file vol=cdr03
   ```

   You can use software for writing CDs to create a CD with volume label CDR03 that contains a single file that is also named CDR03.

Server B

1. Follow the manufacturer's instructions to attach the device to your server.
2. Issue this command on your system to mount the CD.

   ```
   mount -t iso9660 /dev/cdrom /cdrom
   ```

   -t iso9660
   Specifies that the media has a CD file system

   /dev/cdrom
   Specifies the physical description of the first CD on the system

   /cdrom
   Specifies the mount point of the first CD drive

   Note: CD drives lock while the file system is mounted. This prevents use of the eject button on the drive.

3. Ensure that the media is labeled. The software that you use for making a CD also labels the CD. Before you define the drive, you must put formatted, labeled media in the drive. For label requirements, see "Labeling Requirements for Removable Files Devices" on page 131. When you define the drive, the server verifies that a valid file system is present.
4. Define a manual library named CDROM:

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

5. Define the drive in the library:

   ```
   define drive cdrom cddrive
   ```

6. Define a path from the server to the drive at mount point */cdrom*:

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

   For more information about paths, see "Defining Paths" on page 140.

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

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

8. Issue the following Tivoli Storage Manager command to import the node data on the CD volume CDR03.

   ```
   import node user1 filedata=all devclass=cdrom vol=cdr03
   ```

For mounting a removable file device other than CD, use the following command:

```
mount -t ext2 /dev/hdc1 /tsmrfile
```

Labeling Requirements for Removable Files Devices

Tivoli Storage Manager does not provide utilities to format or label media for the REMOVABLEFILE device type. You must use another application to copy the FILE device class data from the CD as a file that has the same name as the volume label. The software used to copy the FILE device class data must also label the removable media. CDs must be formatted using the ISO9660 filesystem. Other removable file volumes such as Jaz or Zip must be formatted using the EXT2 filesystem, and the `-L` option on the `mke2fs` command must be used to specifiy the volume label.

The label on the media must meet the following restrictions:
- No more than 11 characters
- No embedded blanks or periods
- File name must be the same as the volume label

Configuring 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, do the following procedure. This example is for a device containing two StorageTek drives.

1. Set up the media manager to interface with Tivoli Storage Manager. For more information, see Appendix A, "External Media Management Interface Description," on page 741 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 140.
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.

Notes:

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 a Client's Files Together: Collocation" on page 258 and "How Collocation Affects Reclamation" on page 276.

Managing Externally Controlled IBM Tivoli Storage Manager Media

Refer to the documentation for the media manager for detailed setup and management information. The following are some issues specific to Tivoli Storage Manager.

Labeling Media
The media manager handles the labeling of media. However, you must ensure that an adequate supply of blank media is available.

Checking Media into the Library
Externally managed media are not tracked in the Tivoli Storage Manager volume inventory. Therefore, you do *not* perform library check-in procedures by using Tivoli Storage Manager commands.

Using DRM
If you are using DRM, you can use the MOVE DRMEDIA command to request the removal of media from the library. For more information, see Chapter 24, "Using Disaster Recovery Manager," on page 687.

Migrating Media to External Media Manager Control
We strongly recommend that you not migrate media from Tivoli Storage Manager control to control by an external media manager. Instead, use external media management on a new Tivoli Storage Manager configuration or when defining externally managed devices to the server.

Deleting Tivoli Storage Manager Storage Pools from Externally Managed Libraries

Before deleting storage pools associated with externally managed libraries, first delete any volumes associated with the Tivoli Storage Manager library. For more information, see "Deleting a Storage Pool Volume with Data" on page 305.

Troubleshooting 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.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 399.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 232.

Configuring Manually Mounted Devices

In the following example, two DLT drives are attached to the server system. Because an operator must mount tapes for these drives, you must define the drives as part of a *manual* library.

Set up the Device on the Server System

You must first set up the device on the server system. This involves the following tasks:

1. Set the appropriate SCSI ID for the device.
2. Physically attach the device to the server hardware.
3. Install and configure the appropriate device driver for the device.
4. Determine the device name that is needed to define the device to Tivoli Storage Manager.

See "Attaching a Manual Drive" on page 91 and "Installing and Configuring Device Drivers" on page 93 for details.

Define the Device to IBM Tivoli Storage Manager

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 138 and www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

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

```
define path server1 drive01 srctype=server desttype=drive
 library=manualdlt device=/dev/tsmscsi/mt1
define path server1 drive02 srctype=server desttype=drive
 library=manualdlt device=/dev/tsmscsi/mt2
```

For more about device names, see "Determining Device Special File Names" on page 94.

For more information about paths, see "Defining Paths" on page 140.

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 and Updating Tape Device Classes" on page 198.

5. Verify your definitions by issuing the following commands:

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

See "Requesting Information About Libraries" on page 186, "Requesting Information about Drives" on page 188, "Requesting Information about a Device Class" on page 211, and "Requesting Information About Paths" on page 193.

6. Define a storage pool named TAPEDLT_POOL associated with the device class named TAPEDLT_CLASS:

```
define stgpool tapedlt_pool tapedlt_class maxscratch=20
```

Key choices:

a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can use any scratch volumes available without further action on your part. If you do not allow scratch volumes (MAXSCRATCH=0), you must perform the extra step of explicitly defining each volume to be used in the storage pool.

b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 258 and "How Collocation Affects Reclamation" on page 276.

See "Defining or Updating Primary Storage Pools" on page 220.

Label Volumes

Use the following procedure to ensure that volumes are available to the server. Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup. Label and set aside extra scratch volumes for any potential recovery operations you might have later.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries.

Do the following:

1. Label volumes. For example, enter the following command to use one of the drives to label a volume with the ID of vol001:

   ```
   label libvolume manualdlt vol001
   ```

   Note: Tivoli Storage Manager only accepts tapes labeled with IBM standard labels. IBM standard labels are similar to ANSI Standard X3.27 labels except that the IBM standard labels are written in EBCDIC. For a list of IBM media sales contacts who can provide compatible tapes, go to the IBM Web site. If you are using non-IBM storage devices and media, consult your tape-cartridge distributor.

2. Depending on whether you use scratch volumes or private volumes, do one of the following:
   - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
   - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled. For information on defining volumes, see "Defining Storage Pool Volumes" on page 229.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 399.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 232.

Configuring IBM Tivoli Storage Manager for LAN-free Data Movement

You can configure the Tivoli Storage Manager client and server so that the client, through a storage agent, can move its data directly to storage on a SAN. This function, called LAN-free data movement, is provided by IBM Tivoli Storage Manager for Storage Area Networks. As part of the configuration, a storage agent is installed on the client system. Tivoli Storage Manager supports both tape libraries and FILE libraries. This feature supports SCSI, 349X, and ACSLS tape libraries.

The configuration procedure you follow will depend on the type of environment you implement; however in all cases you must do the following:

1. Verify the network connection.
2. Establish communications among client, storage agent, and Tivoli Storage Manager.
3. Configure devices for the storage agent to access.
4. Install and configure Tivoli SANergy if using shared FILE storage.

5. Start the storage agent and verify the LAN-free configuration.

For more information on configuring Tivoli Storage Manager for LAN-free data movement see the *IBM Tivoli Storage Manager Storage Agent User's Guide*.

To help you tune the use of your LAN and SAN resources, you can control the path that data transfers take for clients with the capability of LAN-free data movement. For each client you can select whether data read and write operations use:

- The LAN path only
- The LAN-free path only
- Either path

See the REGISTER NODE and UPDATE NODE commands in the *Administrator's Reference*.

Validating your LAN-free Configuration

Once you have configured your Tivoli Storage Manager client for LAN-free data movement, you can verify your configuration and server definitions by using the VALIDATE LANFREE command. This command allows you to determine which destinations for a given node using a specific storage agent are capable of LAN-free data movement.

The VALIDATE LANFREE command can also be used to determine if there is a problem with an existing LAN-free configuration. You can evaluate the policy, storage pool, and path definitions for a given node using a given storage agent to ensure that an operation is working properly.

To determine if there is a problem with the client node FRED using the storage agent FRED_STA, issue the following:

```
validate lanfree fred fred_sta
```

The output will allow you to see which management class destinations for a given operation type are not LAN-free capable, and provide a brief explanation about why. It will also report the total number of LAN-free destinations.

See the VALIDATE LANFREE command in the *Administrator's Reference* for more information.

Configuring IBM Tivoli Storage Manager for NDMP Operations

Tivoli Storage Manager can use Network Data Management Protocol (NDMP) to communicate with NAS file servers and provide backup and restore services. This feature supports SCSI, 349X, and ACSLS tape libraries. See "NDMP Backup Operations" on page 75.

To configure Tivoli Storage Manager for NDMP operations, you must do the following:

1. Define the libraries and their associated paths.

   Note: Libraries with multiple drive device types cannot be used with NDMP operations.
2. Define a device class for NDMP operations.
3. Define the storage pool for backups performed by using NDMP operations.

4. *Optional:* Select or define a storage pool for storing tables of contents for the backups.
5. Configure Tivoli Storage Manager policy for NDMP operations.
6. Register the NAS nodes with the server.
7. Define a data mover for the NAS file server.
8. Define the drives and their associated paths.

For more information on configuring Tivoli Storage Manager for NDMP operations, see Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143.

Defining Devices and Paths

The following sections describe how to define libraries, drives, and paths to Tivoli Storage Manager. See "Managing Libraries" on page 186, "Managing Drives" on page 188, and "Managing Paths" on page 193 for information about displaying library, drive, and path information, and updating and deleting libraries and drives.

Defining Libraries

Task	Required Privilege Class
Define or update libraries	System or unrestricted storage

Before you can use a drive, you must first define the library to which the drive belongs. This is true for both manually mounted drives and drives in automated libraries. For example, you have several stand-alone tape drives. You can define a library named MANUALMOUNT for these drives by using the following command:

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

For all libraries other than manual libraries, you define the library and then define a path from the server to the library. For example, if you have an IBM 3583 device, you can define a library named ROBOTMOUNT using the following command:

```
define library robotmount libtype=scsi
```

Next, you use the DEFINE PATH command. In the path, you must specify the DEVICE parameter. The DEVICE parameter is required and specifies the device driver's name for the drive, which is the device special file name. The library's robotic mechanism is known by the device special file name. For more about device special file names, see "Determining Device Special File Names" on page 94

```
define path server1 robotmount srctype=server desttype=library
 device=/dev/tsmscsi/lb0
```

For more information about paths, see "Defining Paths" on page 140.

If you have an IBM 3494 Tape Library Dataserver, you can define a library named AUTOMOUNT using the following command:

```
define library automount libtype=349x
```

Next, assuming that you have defined the symbolic name `lib3494` for the library in the configuration file for the library device driver, you define a path for the library:

```
define path server1 automount srctype=server desttype=library device=lib3494
```

Defining SCSI Libraries on a SAN

For a library type of SCSI on a SAN, the server can track the library's serial number. With the serial number, the server can confirm the identity of the device when you define the path or when the server uses the device.

If you choose, you can specify the serial number when you define the library to the server. For convenience, the default is to allow the server to obtain the serial number from the library itself at the time that the path is defined.

If you specify the serial number, the server confirms that the serial number is correct when you define the path to the library. When you define the path, you can set AUTODETECT=YES to allow the server to correct the serial number if the number that it detects does not match what you entered when you defined the library.

Depending on the capabilities of the library, the server may not be able to automatically detect the serial number. Not all devices are able to return a serial number when asked for it by an application such as the server. In this case, the server will not record a serial number for the device, and will not be able to confirm the identity of the device when you define the path or when the server uses the device. See "Recovering from Device Changes on the SAN" on page 142.

Defining Drives

Task	Required Privilege Class
Define drives	System or unrestricted storage

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. For more information about paths, see "Defining Paths" on page 140. When issuing the DEFINE DRIVE command, you must provide some or all of the following information:

Library name
The name of the library in which the drive resides.

Drive name
The name assigned to the drive.

Serial number
The serial number of the drive. The serial number parameter applies only to drives in SCSI libraries. With the serial number, the server can confirm the identity of the device when you define the path or when the server uses the device.

You can specify the serial number if you choose. The default is to allow the server to obtain the serial number from the drive itself at the time that the path is defined. If you specify the serial number, the server confirms that the serial number is correct when you define the path to the drive. When you define the path, you can set AUTODETECT=YES to allow the server to correct the serial number if the number that it detects does not match what you entered when you defined the drive.

Depending on the capabilities of the drive, the server may not be able to automatically detect the serial number. In this case, the server will not record a serial number for the device, and will not be able to confirm the

identity of the device when you define the path or when the server uses the device. See "Recovering from Device Changes on the SAN" on page 142.

Element address
The element address of the drive. The ELEMENT parameter applies only to drives in SCSI libraries. The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can allow the server to obtain the element address from the drive itself at the time that the path is defined, or you can specify the element number when you define the drive.

Depending on the capabilities of the library, the server may not be able to automatically detect the element address. In this case you must supply the element address when you define the drive, if the library has more than one drive. If you need the element address, check the Tivoli Storage Manager device driver support web site at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

For example, to define a drive that belongs to the manual library named MANLIB, enter this command:

```
define drive manlib tapedrv3
```

Next, you define the path from the server to the drive, using the device name used to access the drive:

```
define path server1 tapedrv3 srctype=server desttype=drive library=manlib
 device=/dev/tsmscsi/mt3
```

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.

For example, to define a NAS data mover named NAS1, you enter the following:

```
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. This 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.

For example, if you have a SCSI type library named AUTODLTLIB that has a device name of /dev/tsmscsi/lb3, define the path to the server named ASTRO1 by doing the following:

```
define path astro1 autodltlib srctype=server desttype=library
device=/dev/tsmscsi/lb3
```

If you have a drive, DRIVE01, that resides in library AUTODLTLIB, and has a device name of /dev/tsmscsi/mt4, define it to server ASTRO1 by doing the following:

```
define path astro1 drive01 srctype=server desttype=drive library=autodltlib
 device=/dev/tsmscsi/mt4
```

Shared FILE Volumes

The Tivoli Storage Manager server and any storage agents associated with it are separate systems and each has a different view of the storage it is trying to access. Because of this, problems can arise if path definitions to that storage are not accurate. The server has no way of validating the directory structure and storage paths that storage agents ″see,″ so diagnosing failures of this nature is very difficult.

The mechanisms to map the server view of storage to the storage agent view of storage are DEFINE DEVCLASS-FILE for the server and DEFINE PATH for the storage agent or agents. The DIRECTORY parameter in the DEFINE DEVCLASS-FILE command specifies the directory location or locations where the server places files that represent storage volumes for the FILE device class. For storage agents, the DIRECTORY parameter in the DEFINE PATH command serves the same purpose. The device class definition sets up a directory structure for the server and the DEFINE PATH definition tells the storage agent what that directory structure is. If path information is incorrect, the server and storage agent or agents will not be able to store files.

In order for the server and storage agent to be consistent on the storage they are sharing, the directories defined in the device class definition for the server and on the DEFINE PATH command for the storage agent should reference the same storage, in the same order and with an equal number of directories. This should be the same for each FILE drive that the storage agent is using. Shared file libraries are used to set up the storage pool that will be shared between the server and storage agents. FILE drives within that library are used so that the DEFINE PATH command can convey the information to the storage agent.

Using SANergy with Shared FILE Volumes: In a typical file sharing configuration, a File Server is connected to a server and storage agent or agents on the LAN. When the server or storage agent needs to write data to storage, it contacts the File Server over the LAN. The File Server then contacts the hard disk or storage drive over the SAN and reserves the space needed for the storage agent or server to store volumes. Once the space is reserved, the server or storage agent writes the data to be stored to the File Server over the LAN and then the File Server writes the data again to storage over the SAN. Only one operation can take place at a time, so if the server is in contact with the File Server during an operation, a storage agent attempting to contact the File Server will have to wait its turn.

To maximize throughput and improve performance, SANergy allows a server or storage agent to write directly to a hard disk or storage space over the SAN. A SANergy client is installed on each server and storage agent on the LAN and this client communicates with the SANergy server. When a storage agent or server is ready to write data to storage, the SANergy client on that system contacts the SANergy server. The SANergy server reserves the space on the hard disk and then informs the SANergy client that the server or storage agent can begin writing data

directly to storage. This system bypasses the LAN and maintains the integrity of file systems, while maximizing throughput.

Recovering from Device Changes on the SAN

The SAN environment can shift dramatically due to device or cabling changes. Device IDs assigned by the SAN may be altered due to bus resets or other environmental changes. This ″dynamically″ changing nature of the SAN can cause the ″static″ definitions defined and known to the server (or storage agent) to fail or become unpredictable. For instance, the server may know a device as *id=1* based on the original path specification to the server and original configuration of the LAN. However, some event in the SAN (new device added, cabling change) causes the device to be assigned *id=2*. When the server tries to access the device with *id=1*, it will either get a failure or the wrong target device. The server assists in recovering from changes to devices on the SAN by using serial numbers to confirm the identity of devices it contacts.

When you define a device (drive or library) you have the option of specifying the serial number for that device. If you do not specify the serial number when you define the device, the server obtains the serial number when you define the path for the device. In either case, the server then has the serial number in its database. From then on, the server uses the serial number to confirm the identity of a device for operations.

When the server uses drives and libraries on a SAN, the server attempts to verify that the device it is using is the correct device. The server contacts the device by using the device name in the path that you defined for it. The server then requests the serial number from the device, and compares that serial number with the serial number stored in the server database for that device.

If the serial number does not match, the server begins the process of discovery on the SAN to attempt to find the device with the matching serial number. If the server finds the device with the matching serial number, it corrects the definition of the path in the server's database by updating the device name in that path. The server issues a message with information about the change made to the device. Then the server proceeds to use the device.

You can monitor the activity log for messages if you want to know when device changes on the SAN have affected Tivoli Storage Manager. The following are the number ranges for messages related to serial numbers:

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

Restriction: Some devices do not have the capability of reporting their serial numbers to applications such as the Tivoli Storage Manager server. If the server cannot obtain the serial number from a device, it cannot assist you with changes to that device's location on the SAN.

Chapter 7. Using NDMP for Operations with NAS File Servers

This chapter is about planning, configuring, and managing a backup environment that protects your network-attached storage (NAS) file server by using NDMP. 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 154 "Step 1. Setting Up Tape Libraries for NDMP Operations" on page 155 "Step 2. Configuring Tivoli Storage Manager Policy for NDMP Operations" on page 157 "Step 3. Registering NAS Nodes with the Tivoli Storage Manager Server" on page 158 "Step 4. Defining a Data Mover for the NAS File Server" on page 158 "Step 5. Defining a Path to a Library" on page 158 "Step 6. Defining Tape Drives and Paths for NDMP Operations" on page 159 "Step 7. Labeling Tapes and Checking Tapes into the Library" on page 160 "Step 8. Scheduling NDMP Operations" on page 160 "Step 9. Defining Virtual File Spaces" on page 161
"Backing Up and Restoring NAS File Servers Using NDMP" on page 161
"Managing Table of Contents" on page 165
"Managing NDMP Operations" on page 163
"Managing NAS File Server Nodes" on page 164
"Managing Data Movers Used in NDMP Operations" on page 164
"Managing Storage Pools for NDMP Operations" on page 165

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Requirements

You must meet the following requirements when using NDMP for operations with 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 www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html for a list of NAS file servers that are certified through the Ready for IBM Tivoli software.

© Copyright IBM Corp. 1993, 2005

Note: Vendors on the Ready for IBM Tivoli software list follow guidelines to implement NDMP as specified by Tivoli Storage Manager. If a file server is on the list, it has undergone tests to ensure it is compatible with Tivoli Storage Manager.

The combination of file server model and operating system must be supported by the NAS file server. For more specifics, consult the product information for the NAS file server.

Tape Libraries

The Tivoli Storage Manager server supports two types of libraries for operations using NDMP. The libraries supported are SCSI and ACSLS. 349X tape libraries can also be used with certain NAS file servers.

- **SCSI library**

  A SCSI library that is supported by the Tivoli Storage Manager server. Visit www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html. This type of library can be attached directly either to the Tivoli Storage Manager server or to the NAS file server. When the library is attached directly to the Tivoli Storage Manager server, the Tivoli Storage Manager server controls the library operations by passing the SCSI commands directly to the library. When the library is attached directly to the NAS file server, the Tivoli Storage Manager server controls the library by passing SCSI commands to the library through the NAS file server.

- **ACSLS library**

  An ACSLS library can only be directly connected to the Tivoli Storage Manager server. The Tivoli Storage Manager server controls the library by passing the library request through TCP/IP to the library control server.

  Note: The Tivoli Storage Manager server does not include External Library support for the ACSLS library when the library is used for NDMP operations.

- **349X library**

  A 349X library can only be directly connected to the Tivoli Storage Manager server. The Tivoli Storage Manager server controls the library by passing the library request through TCP/IP to the library manager.

Library Sharing: The Tivoli Storage Manager server that performs NDMP operations can be a library manager for either a SCSI or 349X library, but cannot be a library client. If the Tivoli Storage Manager server that performs NDMP operations is a library manager, that server must control the library directly and not by passing commands through the NAS file server.

Tape Drives

One or more tape drives in the tape library. The NAS file server must be able to access the drives. A NAS device is not supported in a mixed device library. The drives must be supported for tape backup operations by the NAS file server and its operating system. For complete NDMP device support, refer to the NAS file server product documentation.

Drive Sharing: The tape drives can be shared by the Tivoli Storage Manager server and one or more NAS file servers. Also,

when a SCSI or a 349X library is connected to the Tivoli Storage Manager server and not to the NAS file server, the drives can be shared:

- By one or more NAS file servers and one or more Tivoli Storage Manager library clients.
- By one or more NAS file servers and one or more Tivoli Storage Manager storage agents.

Verify the compatibility of specific combinations of a NAS file server, tape devices, and SAN-attached devices with the hardware manufacturers.

Interfaces Used for NDMP Operations

You can use any of the interfaces described in this section to perform NDMP 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, 32-bit or 64-bit AIX®, or 32-bit or 64-bit Sun Solaris system)
- Web client

Server Interfaces:

- Server console
- Command line on the administrative client

  Note: All examples in this chapter use server commands.
- Web administrative interface

The Tivoli Storage Manager Web client interface, available with the backup-archive client, displays the file systems of the 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 "Planning for File-Level Restore" on page 152 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 *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 Backup Operations Using NDMP

During backup operations that use NDMP, the 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 NDMPDUMP for

most devices. You would specify NETAPPDUMP if the NAS file server is a Network Appliance device. You would specify CELERRADUMP if the NAS file server is an EMC Celerra device.

Planning for NDMP Operations

Most of the planning required to implement backup and recovery operations that use NDMP is related to device configuration. You have choices about how to connect and use the libraries and drives.

Planning for Tape Libraries and Drives used in NDMP Operations

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 NAS file server?
- 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.

Determining Where to Attach the Tape Library Robotics

If you are using a SCSI tape library, one of the first steps in planning for NDMP operations is to determine where to attach it. You must determine whether to attach the library robotics to the Tivoli Storage Manager server or to the NAS file server. Regardless of where you connect library robotics, tape drives must always be connected to the NAS file server for NDMP operations.

Distance and your available hardware connections are factors to consider for SCSI libraries. If the library does not have separate ports for robotics control and drive access, the library must be attached to the NAS file server because the NAS file server must have access to the drives. If your SCSI library has separate ports for robotics control and drive access, you can choose to attach the library robotics to either the Tivoli Storage Manager server or the NAS file server. If the NAS file server is at a different location from the Tivoli Storage Manager server, the distance may mean that you must attach the library to the NAS file server.

Whether you are using a SCSI, ACSLS, or 349X library, you have the option of dedicating the library to NDMP operations, or of using the library for NDMP operations as well as most traditional Tivoli Storage Manager operations.

Table 11. Summary of Configurations for NDMP Operations

Configuration	Distance between Tivoli Storage Manager server and library	Library sharing	Drive sharing between Tivoli Storage Manager and NAS file server	Drive sharing between NAS file servers	Drive sharing between storage agent and NAS file server
Configuration 1 (SCSI library connected to the Tivoli Storage Manager server)	Limited by SCSI or FC connection	Supported	Supported	Supported	Supported
Configuration 2 (SCSI library connected to the NAS file server)	No limitation	Not supported	Supported	Supported	Not supported
Configuration 3 (349X library)	May be limited by 349X connection	Supported	Supported	Supported	Supported

Configuration 1: SCSI Library Connected to the Tivoli Storage Manager Server

In this configuration, the tape library must have separate ports for robotics control and for drive access. In addition, the library must be within Fibre-Channel range or SCSI bus range of both the Tivoli Storage Manager server and the 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 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 10. 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 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 11. 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. See Chapter 5, "Attaching Devices to the Server System," on page 91 for more information. 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 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 12. Configuration 3: 349X Library Connected to the Tivoli Storage Manager Server

Configuration 4: ACSLS Library Connected to the Tivoli Storage Manager Server

For this configuration, you connect the tape library to the system as for traditional Tivoli Storage Manager operations. See Chapter 5, "Attaching Devices to the Server System," on page 91 for more information. The ACSLS 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 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 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 13. Configuration 4: ACSLS Library Connected to the Tivoli Storage Manager Server

Determining How to Use the Drives in the Library

Drives can be used for multiple purposes because of the flexible configurations allowed by Tivoli Storage Manager. For NDMP operations, the NAS file server must have access to the drive. The Tivoli Storage Manager server can also have access to the same drive, depending on your hardware connections and limitations. All drives are defined to the Tivoli Storage Manager server. However, the same drive may be defined for both traditional Tivoli Storage Manager operations and NDMP operations. Figure 14 on page 152 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 the figure, you would do the following:

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.

See "Step 6. Defining Tape Drives and Paths for NDMP Operations" on page 159 for more information.

Planning for File-Level Restore

When you do a backup via NDMP, 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. See "Managing Table of Contents" on page 165 for more 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 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. See the *Administrator's Reference* for more information about the BACKUP NODE command. 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.

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.

International Characters for Network Appliance File Servers

All systems that create or access data on a particular NAS 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 Network Appliance 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).

  Note: 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.

Directory-Level Backup and Restore

If you have a large NAS 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.

Configuring Tivoli Storage Manager for NDMP Operations

Before beginning the configuration of Tivoli Storage Manager for NDMP operations, ensure that you register the required license. See "Licensing IBM Tivoli Storage Manager" on page 457.

The following is a checklist to use when configuring:

1. Set up the tape library and media. See "Step 1. Setting Up Tape Libraries for NDMP Operations" on page 155, where the following steps are described in more detail.
   a. Attach the SCSI library to the NAS file server or to the Tivoli Storage Manager server, or attach the ACSLS library or 349X library to the Tivoli Storage Manager server.
   b. Define the library with a library type of SCSI, ACSLS, or 349X.
   c. Define a device class for the tape drives.
   d. Define a storage pool for NAS backup media.
   e. Define a storage pool for storing a table of contents. This step is optional.
2. Configure Tivoli Storage Manager policy for managing NAS image backups. See "Step 2. Configuring Tivoli Storage Manager Policy for NDMP Operations" on page 157.
3. Register a NAS file server node with the Tivoli Storage Manager server. See "Step 3. Registering NAS Nodes with the Tivoli Storage Manager Server" on page 158.
4. Define a data mover for the NAS file server. See "Step 4. Defining a Data Mover for the NAS File Server" on page 158.
5. Define a path from either the Tivoli Storage Manager server or the NAS file server to the library. See "Step 5. Defining a Path to a Library" on page 158.

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 "Step 6. Defining Tape Drives and Paths for NDMP Operations" on page 159.
7. Check tapes into the library and label them. See "Step 7. Labeling Tapes and Checking Tapes into the Library" on page 160.
8. Set up scheduled backups for NAS file servers. This step is optional. See "Step 8. Scheduling NDMP Operations" on page 160.
9. Define a virtual file space name. This step is optional. See "Step 9. Defining Virtual File Spaces" on page 161.

Step 1. Setting Up Tape Libraries for NDMP Operations

A. Connect the Library and Drives for NDMP Operations

Connect the SCSI Library

Before setting up a SCSI tape library for NDMP operations, you should have already determined whether you want to attach your library robotics control to the Tivoli Storage Manager server or to the NAS file server. See "Planning for Tape Libraries and Drives used in NDMP Operations" on page 146.

Connect the SCSI tape library robotics to the Tivoli Storage Manager server or to the NAS file server. See the manufacturer's documentation for instructions.

Library Connected to Tivoli Storage Manager
Make a SCSI or Fibre Channel connection between the Tivoli Storage Manager server and the library robotics control port. Then connect the NAS file server with the drives you want to use for NDMP operations.

Library Connected to NAS File Server
Make a SCSI or Fibre Channel connection between the NAS file server and the library robotics and drives.

Connect the ACSLS Library

Connect the ACSLS tape library to the Tivoli Storage Manager server. See Chapter 5, "Attaching Devices to the Server System," on page 91 for more information.

Connect the 349X Library

Connect the 349X tape library to the Tivoli Storage Manager server. See Chapter 5, "Attaching Devices to the Server System," on page 91 for more information.

B. Define the Library for NDMP Operations

Define the tape library to Tivoli Storage Manager.

Note: The library has to be a single device type, not a mixed device one.

SCSI Library

```
define library tsmlib libtype=scsi
```

ACSLS Library

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

349X Library

```
define library tsmlib libtype=349x
```

C. Define a Device Class for NDMP Operations

Create a device class for NDMP operations. A device class defined with a device type of NAS is not explicitly associated with a specific drive type (for example, 3570 or 8mm). However, we recommend that you define separate device classes for different drive types.

In the device class definition:

- Specify NAS as the value for the DEVTYPE parameter.
- Specify 0 as the value for the MOUNTRETENTION parameter. MOUNTRETENTION=0 is required for NDMP operations.
- Specify a value for the ESTCAPACITY parameter.

For example, to define a device class named NASCLASS for a library named NASLIB and media whose estimated capacity is 40GB, enter the following command:

```
define devclass nasclass devtype=nas library=naslib mountretention=0
 estcapacity=40g
```

D. Define a Storage Pool for NDMP Media

The storage pools you define for storage of file system images produced during backups using NDMP are different from storage pools used for conventional Tivoli Storage Manager media. They are defined with different data formats. Tivoli Storage Manager operations use storage pools defined with a NATIVE or NONBLOCK data format. NDMP operations require storage pools with a data format that matches the NAS file server and the backup method to be used.

When you use the NDMPDUMP data format for NDMP operations, it is possible to store backup data from different NAS vendors into the same storage pool. The preferred method, however, is to place data from each vendor into a separate storage pool.

For example, to define a storage pool named NDMPPOOL for a file server which is neither a Network Appliance nor a Celerra file server, enter the following command:

```
define stgpool ndmppool nasclass maxscratch=10 dataformat=ndmpdump
```

To define a storage pool named NASPOOL for a Network Appliance file server, enter the following command:

```
define stgpool naspool nasclass maxscratch=10 dataformat=netappdump
```

To define a storage pool named CELERRAPOOL for an EMC Celerra file server, enter the following command:

```
define stgpool celerrapool nasclass maxscratch=10 dataformat=celerradump
```

Attention: Ensure that you do not accidentally use storage pools that have been defined for NDMP operations in traditional Tivoli Storage Manager operations. Be especially careful when assigning the storage pool name as the value for the DESTINATION parameter of the DEFINE COPYGROUP command. Unless the destination is a storage pool with the appropriate data format, the backup will fail.

E. Define a Storage Pool for a Table of Contents

This step is optional. 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, enter the following command:

```
define stgpool tocpool disk
```

Then, you must define volumes for the storage pool. See "Configuring Random Access Volumes on Disk Devices" on page 87 for more information.

Step 2. Configuring Tivoli Storage Manager Policy for NDMP Operations

Policy allows you to manage the number and retention time of NDMP image backup versions. See "Configuring Policy for NDMP Operations" on page 401.

1. Create a policy domain for NAS 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, enter 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, enter 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, enter 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, enter 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 "Step 3. Registering NAS Nodes with the Tivoli Storage Manager Server" on page 158.

Applying Policy to 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 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 file server backups. See "Creating Client Option Sets on the Server" on page 340 for more information.

Step 3. Registering NAS Nodes with the Tivoli Storage Manager Server

Register the NAS 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. For example, to register a NAS file server as a node named NASNODE1, with a password of NASPWD1, in a policy domain named NASDOMAIN, enter the following 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. You must specify TYPE=NAS so that only NAS nodes are displayed:

```
query node type=nas
```

Step 4. Defining a Data Mover for the NAS File Server

Define a data mover for each NAS file server, using NDMP 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. For example, to define a data mover for a NAS node named NASNODE1, enter the following 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 Network Data Management Protocol (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 Network Appliance 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 Network Appliance file server uses for tape backup. This data format must match the data format of the target storage pool.

Step 5. Defining a Path to a Library

Define a path to the SCSI library from either the Tivoli Storage Manager or the NAS file server.

SCSI Library Connected to Tivoli Storage Manager
For example, issue the following 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/tsmscsi/lb1
```

SCSI Library Connected to NAS File Server
For example, issue the following command to define a path between a Network Appliance NAS data mover named NASNODE1 and a library named NASLIB, use the following command:

```
define path nasnode1 naslib srctype=datamover desttype=library device=mc0
```

The value of the DEVICE parameter is the special file name for the tape library as it is known to the NAS file server. See "Obtaining Special File Names for Path Definitions."

Define a path to the 349X library from the Tivoli Storage Manager server.

349X Library Connected to Tivoli Storage Manager
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
```

Note: DEFINE PATH is not needed for an ACSLS library.

Obtaining Special File Names for Path Definitions

When you are creating paths, you must provide special file names for tape libraries and drives. For paths from a NAS data mover, the value of the DEVICE parameter in the DEFINE PATH command is the name by which the NAS file server knows a library or drive. You can obtain these names, known as special file names, by querying the NAS file server. For information about how to obtain names for devices that are connected to a NAS file server, consult the product information for the file server.

For example, for a Network Appliance file server, connect to the file server using telnet and issue the SYSCONFIG command. To display the device names for tape libraries, use this command:

```
sysconfig -m
```

To display the device names for tape drives, use this command:

```
sysconfig -t
```

For the Celerra file server, connect to the Celerra control workstation using telnet. To see the devices attached to a particular data mover, use the "server_devconfig" command on the control station:

```
 server_devconfig  server_#  -p  -s -n
```

The SERVER_# is the data mover on which the command should be run.

Step 6. Defining Tape Drives and Paths for NDMP Operations

Define the tape drives that you want to use in NDMP operations and the paths to those drives. Depending on your hardware and network connections, you can use the drives for only NDMP operations, or for both traditional Tivoli Storage Manager operations and NDMP operations. For example,

1. Define a drive named NASDRIVE1 for the library named NASLIB.

```
define drive naslib nasdrive1 element=117
```

Note: When you define SCSI drives to the Tivoli Storage Manager server, the ELEMENT parameter must contain a number if the library has more than one drive. If the drive is shared between the NAS file server and the Tivoli Storage Manager server, the element address is automatically detected. If the library is connected to a NAS file server only, there is no automatic detection of the element address and you must supply it. Element numbers are available from device manufacturers. Element numbers for tape drives are also available in the device support information available on the Tivoli Web site at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

2. Define a path for the drive:
   - For example, if the drive is to be used only for NDMP operations, issue the following command:

     ```
     define path nasnode1 nasdrive1 srctype=datamover desttype=drive
      library=naslib device=rst0l
     ```

     Note: For a drive connected only to the NAS file server, do not specify ASNEEDED for the CLEANFREQUENCY parameter of the DEFINE DRIVE command.
   - For example, if a drive is to be used for both Tivoli Storage Manager and NDMP operations, enter the following commands:

     ```
     define path server1 nasdrive1 srctype=server desttype=drive
      library=naslib device=/dev/rmt0
     define path nasnode1 nasdrive1 srctype=datamover desttype=drive
      library=naslib device=rst0l
     ```

Step 7. Labeling Tapes and Checking Tapes into the Library

You must label the tapes and check the tapes into the tape library. These tasks are the same as for other libraries. See “Labeling Removable Media Volumes” on page 168 for more information.

Step 8. Scheduling NDMP Operations

You can schedule the backup or restore of images produced by NDMP operations by using administrative schedules that process the BACKUP NODE or RESTORE NODE administrative commands. The BACKUP NODE and RESTORE NODE commands can be used only for nodes of TYPE=NAS. See “Backing Up and Restoring NAS File Servers Using NDMP” on page 161 for information about the commands.

For example, to create an administrative schedule called NASSCHED to back up all file systems for a node named NASNODE1, enter the following:

```
define schedule nassched type=administrative cmd='backup node nasnode1' active=yes
starttime=20:00 period=1 perunits=days
```

The schedule is active, and is set to run at 8:00 p.m. every day. See Chapter 18, “Automating Server Operations,” on page 471 for more information.

Step 9. Defining Virtual File Spaces

Use a virtual file space definition to perform NAS directory level backups. In order to reduce backup and restore times for large file systems, map a directory path from a NAS file server to a virtual file space name on the Tivoli Storage Manager server.

To create a virtual file space name for the directory path on the NAS device, use 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 "Performing Directory-Level Backup and Restore" on page 162 for more information.

Backing Up and Restoring NAS File Servers Using NDMP

After you have completed the steps in "Configuring Tivoli Storage Manager for NDMP Operations" on page 154, you are ready for NDMP operations. Use either a client interface or an administrative interface described in "Interfaces Used for NDMP Operations" on page 145 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 file server named NAS1, enter the following command:

```
dsmc backup nas -nasnodename=nas1 {/vol/vol1}
```

For more information on the command, see *Tivoli Storage Manager for Windows Backup-Archive Clients Installation and User's Guide* or *Tivoli Storage Manager for UNIX Backup-Archive Clients Installation and User's Guide*.

Note: 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 entering the following command:

```
backup node nas1 /vol/vol1
```

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.

Performing 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. See "Planning for File-Level Restore" on page 152 for instructions on creating a table of contents. 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, 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.

See "Managing Table of Contents" on page 165 for more information on table of contents.

Performing Directory-Level Backup and Restore

The DEFINE VIRTUALFSMAPPING command maps a directory path of a NAS file server to a virtual file space name on the Tivoli Storage Manager server. Once 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.

File Level Restore From a Directory-Level Backup Image

File-level restore is supported for directory-level backup images. As with a NAS file system backup, a 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.

Snapshot Backup and Restore

NDMP directory level backup gives you the ability to back up user created snapshots of a NAS file system; those are then stored as subdirectories. The snapshots can be taken at any time, and the backup to tape can be deferred to a more convenient time. For example, to backup a snapshot created for a Network Appliance file system, perform the following:

1. On the console for the NAS device, issue the command to create the snapshot. SNAP CREATE is the command for a Network Appliance 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 Network Appliance, 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. Once 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.

Managing NDMP Operations

Administrator activities for NDMP operations include managing:

- 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 nodes. For example, assume you have created a new policy domain named NASDOMAIN for NAS nodes and you want to update a NAS node named NASNODE1 to include it in the new domain. You might first query the node.

```
query node nasnode1 type=nas
```

Then you might change the domain of the node with the following command:

```
update node nasnode1 domain=nasdomain
```

Renaming a NAS Node

To rename a NAS node, you must also rename the corresponding NAS data mover; both must have the same name. For example, to rename NASNODE1 to NAS1 you must perform the following steps:

1. Delete all paths between data mover NASNODE1 and libraries and between data mover NASNODE1 and drives. See "Deleting Paths" on page 194.
2. Delete the data mover defined for the NAS node. See "Managing Data Movers Used in NDMP Operations."
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. See "Step 4. Defining a Data Mover for the NAS File Server" on page 158.

   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. See "Step 5. Defining a Path to a Library" on page 158.
6. Define paths between the NAS data mover and any drives used for NDMP operations. See "Step 6. Defining Tape Drives and Paths for NDMP Operations" on page 159.

Deleting a NAS Node

To delete the NAS node, first delete any file spaces for the node. Then delete any paths from the data mover before deleting the data mover. Delete any virtual file space definitions for the node. Then you can enter the following command:

```
remove node nas1
```

Managing Data Movers Used in NDMP Operations

You can update, query, and delete the data movers you define for NAS file servers. For example, if you shut down a NAS file server for maintenance, you might want to take the data mover offline. To do this, first query your data movers to identify the data mover for the NAS file server you want to maintain.

```
query datamover nasnode1
```

Then 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. Then 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 operations. Remove Tivoli Storage Manager server access by deleting the path definition with the following command:

```
delete path server1 nasdrive1 srctype=server desttype=drive library=naslib
```

Managing Storage Pools for NDMP Operations

Because of the different data format, managing storage pools that store backup images produced by NDMP operations are 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 back up a storage pool that is used for NDMP backups.

The NDMPDUMP data format allows you to restore data from one NAS vendor system to another NAS vendor system, but you should verify compatibility between the two systems prior to restoring data. Another option is to maintain a separate storage pool for each NAS vendor.

The following DEFINE STGPOOL and UPDATE STGPOOL parameters are ignored because storage pool hierarchies, reclamation, and migration are not supported for these storage pools:

MAXSIZE
NEXTSTGPOOL
LOWMIG
HIGHMIG
MIGDELAY
MIGCONTINUE
RECLAIMSTGPOOL
OVFLOLOCATION

Attention: Ensure that you do not accidentally use storage pools that have been defined for NDMP operations in traditional Tivoli Storage Manager operations. Be especially careful when assigning the storage pool name as the value for the DESTINATION parameter of the DEFINE COPYGROUP command. Unless the destination is a storage pool with the appropriate data format, the backup will fail.

Managing Table of Contents

Use the SET TOCLOADRETENTION command to specify the approximate number of minutes that an unreferenced table of contents will remain loaded in the Tivoli Storage Manager database. The Tivoli Storage Manager server-wide table of contents retention value will determine how long a loaded table of contents is retained in the database after the latest access to information in the table of

contents. Because table of contents information is loaded into temporary database tables, this information is lost if the server is halted, even if the table of contents retention period has not elapsed.

At installation, the retention time is set to 120 minutes. Use the QUERY STATUS command to see the table of contents retention time.

Use the QUERY NASBACKUP command to display information about the file system image objects that have been backed up for a specific NAS node and file space. By issuing the command, you can see a display of all backup images generated by NDMP and whether each image has a corresponding table of contents.

Note: The Tivoli Storage Manager server may store a full backup in excess of the number of versions you specified, if that full backup has dependent differential backups. QUERY NASBACKUP will not display the extra versions.

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.

Chapter 8. Managing Removable Media Operations

This chapter describes routine removable media operations including the following:

- Preparing media for use (checking volumes into automated libraries and labeling volumes)
- Controlling how and when media are reused
- Ensuring that sufficient media are available
- Responding to Tivoli Storage Manager requests to operators
- Managing libraries, drives (including drive cleaning), paths, and data movers

See the following sections:

Tasks:
"Preparing Removable Media"
"Labeling Removable Media Volumes" on page 168
"Checking New Volumes into a Library" on page 171
"Controlling Access to Volumes" on page 176
"Reusing Tapes in Storage Pools" on page 176
"Reusing Volumes Used for Database Backups and Export Operations" on page 178
"Managing Volumes in Automated Libraries" on page 179
"Managing Server Requests for Media" on page 183
"Managing Libraries" on page 186
"Managing Drives" on page 188
"Managing Paths" on page 193
"Managing Data Movers" on page 194

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Preparing Removable Media

When Tivoli Storage Manager accesses a removable media volume, it checks the volume name in the label header to ensure that the correct volume is accessed. To prepare a volume for use, do the following:

1. Label the volume. Any tape or optical volumes must be labeled before the server can use them. See "Labeling Removable Media Volumes" on page 168.
2. For automated libraries, check the volume into the library. See "Checking New Volumes into a Library" on page 171.

© Copyright IBM Corp. 1993, 2005

Tip: When you use the LABEL LIBVOLUME command with drives in an automated library, you can label and check in the volumes with one command.

3. If the storage pool cannot contain scratch volumes (MAXSCRATCH=0), identify the volume to Tivoli Storage Manager by name so that it can be accessed later. For details, see "Defining Storage Pool Volumes" on page 229.

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

Labeling Removable Media Volumes

To check in and label volumes in one operation, you can use the LABEL LIBVOLUME command or the AUTOLABEL parameter on the DEFINE and UPDATE LIBRARY commands. (For information about the AUTOLABEL parameter, see "Labeling New Volumes using AUTOLABEL" on page 169.) When you use the LABEL LIBVOLUME command, which is issued from the server console or an administrative client, you provide parameters that specify:

- The name of the library where the storage volume is located
- The name of the storage volume
- Whether to overwrite a label on the volume
- Whether to search an automated library for volumes for labeling
- Whether to read media labels:
  - To prompt for volume names in SCSI libraries
  - To read the barcode label for each cartridge in SCSI, 349X, and ACSLS libraries
- Whether to check in the volume:
  - To add the volume to the scratch pool
  - To designate the volume as private
- The type of device (applies to 349X libraries only)

To use the LABEL LIBVOLUME command, there must be at least one drive that is not in use by another Tivoli Storage Manager process. This includes volumes that are mounted but idle. If necessary, use the DISMOUNT VOLUME command to dismount the idle volume to make that drive available.

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

Attention:

- By overwriting a volume label, you destroy all of the data that resides on the volume. Use caution when overwriting volume labels to avoid destroying important data.
- The labels on VolSafe volumes can be overwritten only once. Therefore, you should use the LABEL LIBVOLUME command only once for VolSafe volumes. You can guard against overwriting the label by using the OVERWRITE=NO option on the LABEL LIBVOLUME command.

By overwriting a volume label, you destroy all of the data that resides on the volume. Use caution when overwriting volume labels to avoid destroying important data.

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

- Explicitly name one volume.
- Enter a range of volumes by using the VOLRANGE parameter.
- Use the VOLLIST parameter to specify a file that contains a list of volume names or to explicitly name one or more volumes.

For automated libraries, you are prompted to insert the volume in the entry/exit slot of the library. If no I/O convenience station is available, insert the volume in an empty slot. For manual libraries, you are prompted to load the volume directly into a drive.

Labeling Volumes In a Manual Drive

Suppose that you want to label a few new volumes by using a manual tape drive that is defined as */dev/tsmscsi/mt5*. The drive is attached at SCSI address 5. Enter the following command:

```
label libvolume tsmlibname volname
```

Note: The LABEL LIBVOLUME command selects the next free drive. If you have more than one free drive, this may not be */dev/tsmscsi/mt5*.

Labeling Volumes in a SCSI or ACSLS Library

You can label volumes one at a time or let Tivoli Storage Manager search the library for volumes.

Labeling Volumes One at a Time: If you choose to label volumes one at a time, do the following:

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: Suppose you want to label a few new volumes in a SCSI library. 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.

Note: This example works for libraries that do not have entry and exit ports. Enter the following command:

```
label libvolume tsmlibname volname overwrite=yes checkin=scratch
```

Labeling New Volumes using AUTOLABEL: To automatically label tape volumes, you can use the AUTOLABEL parameter on the DEFINE and UPDATE LIBRARY commands. Using this parameter eliminates the need to pre-label a set of tapes. It is also more efficient than using the LABEL LIBVOLUME command, which requires you to mount volumes separately. If you use the AUTOLABEL parameter with a SCSI library, you must check in tapes by specifying

CHECKLABEL=BARCODE on the CHECKIN LIBVOLUME command. The AUTOLABEL parameter defaults to YES for all non-SCSI libraries and to NO for SCSI libraries.

Searching the Library: The LABEL LIBVOLUME command searches all of the storage slots in the library for volumes and tries to label each one that it finds. You choose this mode when you specify the SEARCH=YES parameter. 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 those volumes in the INSERT category and the library's private, scratch, and WORM scratch categories. All other volumes are ignored by the labeling process. This precaution prevents the inadvertent destruction of that data on volumes being actively used by other systems connected to the library device.

Note: The LABEL LIBVOLUME command labels volumes in the INSERT category, the private category (PRIVATECATEGORY), the scratch category (SCRATCHCATEGORY) and the WORM scratch category (WORMSCRATCHCATEGORY), but does not label the volumes already checked into the library.

Suppose that you want to label all of the volumes that are in the INSERT category in an IBM 3494 tape library. Enter the following command:

```
label libvolume tsmlibname search=yes devtype=3590
```

Checking New Volumes into a Library

Task	Required Privilege Class
Inform the server when a new volume is available in an automated library	System or unrestricted storage

To inform the server that a new volume is available in an automated library, check in the volume with the CHECKIN LIBVOLUME command or LABEL LIBVOLUME command with the CHECKIN option specified. When a volume is checked in, the server adds the volume to its library volume inventory. You can use the LABEL LIBVOLUME command to check in and label volumes in one operation.

Notes:

1. Do not mix volumes with barcode labels and volumes without barcode labels in a library device because barcode scanning can take a long time for unlabeled volumes.
2. You must use the CHECKLABEL=YES (not NO or BARCODE) option on the CHECKIN LIBVOLUME command when checking VolSafe volumes into a library. This is true for both ACSLS and SCSI libraries.
3. When exporting data from a Tivoli Storage Manager server other than an OS/400 PASE server (for example, from a Tivoli Storage Manager Windows server) to a Tivoli Storage Manager OS/400 PASE server, use a server-to-server export rather than an export to sequential media. The CHECKIN LIBVOLUME command fails on a Tivoli Storage Manager OS/400 PASE server when the server attempts to check in a sequential-media volume containing export data from servers other than OS/400 PASE servers. If you must use an LTO device to create an export tape, follow these steps:
   a. Define a Tivoli Storage Manager library of type MANUAL.
   b. Define an LTO tape drive in the MANUAL library.
   c. Label a tape with six characters or less.
   d. Perform the export.

When you check in a volume, you must supply the name of the library and the status of the volume (private or scratch).

To check in one or just a few volumes, you can specify the name of the volume with the command, and issue the command for each volume. See “Checking Volumes into a SCSI Library One at a Time” on page 172.

To check in a larger number of volumes, you can use the search capability of the CHECKIN command (see “Checking in Volumes in Library Slots” on page 173) or you can use the VOLRANGE parameter of the CHECKIN command.

When using the CHECKIN LIBVOLUME command, be prepared to supply some or all of the following information:

Library name
Specifies the name of the library where the storage volume is to be located.

Volume name
Specifies the volume name of the storage volume being checked in.

Status Specifies the status that is assigned to the storage volume being checked in. If you check in a volume that has already been defined in a storage pool or in the volume history file, you must specify a volume status of *private*

(STATUS=PRIVATE). This status ensures that the volume is not overwritten when a scratch mount is requested. The server does not check in a volume with scratch status when that volume already belongs to a storage pool or is a database, export, or dump volume.

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

Check label
Specifies whether Tivoli Storage Manager should read sequential media labels of volumes during CHECKIN command processing, or use a barcode reader. See "Checking Media Labels" on page 174.

Swap Specifies whether Tivoli Storage Manager will initiate a swap operation when an empty slot is not available during CHECKIN command processing. See "Allowing Swapping of Volumes When the Library Is Full" on page 174.

Mount wait
Specifies the maximum length of time, in minutes, to wait for a storage volume to be mounted.

Search
Specifies whether Tivoli Storage Manager searches the library for volumes that have not been checked in. See "Checking Volumes into a SCSI Library One at a Time," "Checking in Volumes in Library Slots" on page 173, and"Checking in Volumes in Library Entry/Exit Ports" on page 173.

Device type
This parameter allows you to specify the device type for the volume being checked in. Use this parameter only for 349X libraries in which the drives do not have defined paths.

Checking Volumes into a SCSI Library One at a Time

Specify SEARCH=NO if you want to check in only a single volume that is not currently in the library. Tivoli Storage Manager requests that the mount operator load the volume in the entry/exit port of the library.

If the library does not have an entry/exit port, Tivoli Storage Manager requests that the mount operator load the volume into a slot within the library. The request specifies the location with an *element address*. For any library or medium changer that does not have an entry/exit port, you need to know the element addresses for the cartridge slots and drives. If there is no worksheet listed for your device in www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html, 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. See www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html to find the worksheet. 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

Specify SEARCH=NO for a 349X library, to search for volumes that have already been inserted into the library via the convenience or bulk I/O station.

```
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

Specify SEARCH=YES if you want the server to search the library slots for new volumes that have not already been added to the library volume inventory. Use this mode when you have a large number of volumes to check in, and you want to avoid issuing an explicit CHECKIN LIBVOLUME command for each volume. For example, for a SCSI library you can simply open the library access door, place all of the new volumes in unused slots, close the door, and issue the CHECKIN LIBVOLUME command with SEARCH=YES.

If you are using a 349X library, the server searches only for new volumes in the following categories:

- INSERT
- Tivoli Storage Manager's private category (PRIVATECATEGORY, specified when you define the library)
- Tivoli Storage Manager's scratch category (SCRATCHCATEGORY, specified when you define the library)
- Tivoli Storage Manager's WORM scratch category (WORMSCRATCHCATEGORY, specified when you define the library)

This restriction prevents the server from using volumes owned by another application that is accessing the library simultaneously.

Checking in Volumes in Library Entry/Exit Ports

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. 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 Tivoli Storage Manager-controlled libraries.

Checking Media Labels

When you check in a volume, you can specify whether Tivoli Storage Manager should read the labels of the media during check-in 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. For information on how to label new volumes, see "Preparing Removable Media" on page 167.

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
```

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 check-in fails unless you allow *swapping*. If you allow swapping and the library is full, Tivoli Storage Manager selects a volume to eject before checking in the volume you requested.

Use the CHECKIN LIBVOLUME command to allow swapping. When you specify YES for the SWAP parameter, Tivoli Storage Manager initiates a swap operation if an empty slot is not available to check in a volume. Tivoli Storage Manager ejects the volume that it selects for the swap operation from the library and replaces the ejected volume with the volume that is being checked in. For example:

```
checkin libvolume auto wpdv00 swap=yes
```

Tivoli Storage Manager selects the volume to eject by checking first for any available scratch volume, then for the least frequently mounted volume.

Special Considerations for WORM Tape Media

Tivoli Storage Manager supports the following types of WORM (write once, read many) media:

- StorageTek VolSafe
- Sony AIT50 and AIT100
- IBM 3592
- LTO
- Quantum SDLT 600

To receive the full benefit of using WORM media with Tivoli Storage Manager, consider the following:

- **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.

- **Checking in Volumes:** Note the following differences when checking in volumes:
  - **VolSafe, Sony AIT, LTO, and SDLT:** Library changers cannot identify the difference between WORM tape media and standard RW tape media. To determine what type of media are being used, a volume must be loaded into a drive. Therefore, when checking in a VolSafe, Sony AIT, LTO or SDLT WORM volume, you must use the CHECKLABEL=YES option on the CHECKIN LIBVOLUME command.
  - **IBM 3592:** 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 do provide the required support.
- **Potential mount failures:** If WORM tape media are loaded into a drive for a 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.
- **Device drivers:** For information about selecting device drivers for IBM and non-IBM devices, see "Installing and Configuring Device Drivers" on page 93.
- **Loading the correct media:** External and manual libraries use separate logical libraries to segregate their media. Ensuring that the correct media are loaded is the responsibility of the operator and the library manager software.
- **Composition of a storage pool:** A storage pool can consist of either WORM or RW media, but not both.
- **Database backup and export operations:** Do not use WORM tapes for database backup or export operations. Doing so wastes tape following a restore or import operation.
- **Relabeling volumes:** You cannot relabel a WORM cartridge if it contains data. This applies to Sony AIT WORM, LTO WORM, SDLT 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. Therefore, you should use the LABEL LIBVOLUME command only once for VolSafe volumes. You can guard against overwriting the label by using the OVERWRITE=NO option on the LABEL LIBVOLUME command.
- **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 use the CHECKOUT LIBVOLUME command.
- **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.
- **Automatic creation of DLT WORM volumes in SCSI and ACSLS libraries:** If you have SDLT600 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 RW volumes or blank volumes to WORM volumes. Alternatively, in SCSI or ACSLS libraries, the Tivoli Storage Manager server will create scratch DLT WORM volumes automatically

when they are required. Conversion takes place if the server cannot locate any scratch WORM volumes in a library's inventory. The server then converts any 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.

- **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).

For information about defining device classes for WORM tape media, see "Defining and Updating VOLSAFE Device Classes" on page 210 and "Defining and Updating Tape Device Classes" on page 198.

Managing the Volume Inventory

With Tivoli Storage Manager, you manage your volume inventory by performing the following tasks:

- Controlling Tivoli Storage Manager access to volumes
- Reusing tapes in storage pools
- Reusing volumes used for database backups and export operations
- Maintaining a supply of scratch volumes

Note: 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

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 to send the data they contain offsite for safekeeping, a more controlled way to do this is to use a copy storage pool. You can back up your primary storage pools to a copy storage pool and then send the copy storage pool volumes offsite. You can track these copy storage pool volumes by changing their access mode to offsite, and updating the volume history to identify their location. For more information, see "Backing Up Storage Pools" on page 648.

Reusing Tapes in Storage Pools

To reuse tapes in storage pools, you must do two things:

Expiration Processing of 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 391.

Reclamation of 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 267.

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 271.

Setting Up a Tape Rotation

Over time, media ages, and some of the backup data located on it may no longer be needed. You can set Tivoli Storage Manager policy to determine how many backup versions are retained and how long they are retained. See "Basic Policy Planning" on page 358. Then, expiration processing allows the server to delete files you no longer want to keep. See "File Expiration and Expiration Processing" on page 361. 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. See "Running Expiration Processing to Delete Expired Files" on page 391.

Reusing Media - Reclamation Processing

Data on tapes may expire, move, or be deleted. Reclamation processing consolidates any unexpired data by moving it from multiple volumes onto fewer volumes. The media can then be returned to the storage pool and reused.

You can set a reclamation threshold that allows Tivoli Storage Manager to reclaim volumes whose valid data drops below a threshold. The threshold is a percentage of unused space on the volume and is set for each storage pool. The amount of data on the volume and the reclamation threshold for the storage pool affects when the volume is reclaimed. See "Reclaiming Space in Sequential Access Storage Pools" on page 267.

Determining When Media Have Reached End of Life

You can use Tivoli Storage Manager to display statistics about volumes, including the number of write operations performed on the media and the number of write errors. For media initially defined as private volumes, Tivoli Storage Manager maintains this statistical data, even as the volume

is reclaimed. You can compare the information with the number of write operations and write errors recommended by the manufacturer. For media initially defined as scratch volumes, Tivoli Storage Manager overwrites this statistical data each time the media are reclaimed.

Reclaim any valid data from volumes that have reached end of life. If the volumes are in automated libraries, check them out of the volume inventory. Delete private volumes the database with the DELETE VOLUME command. See "Reclaiming Space in Sequential Access Storage Pools" on page 267.

Ensuring Media are Available for the Tape Rotation

Over time, the demand for volumes may cause the storage pool to run out of space. You can set the maximum number of scratch volumes high enough to meet demand by doing one or both of the following:

- Increase the maximum number of scratch volumes by updating the storage pool definition. Label and check in new volumes to be used as scratch volumes if needed.
- Make volumes available for reuse by running expiration processing and reclamation, to consolidate data onto fewer volumes. See "Reusing Tapes in Storage Pools" on page 176.

For automated libraries, see "Managing Server Requests for Media" on page 183.

Reusing Volumes Used for Database Backups and Export Operations

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 have previously been used for database backup or export, use the DELETE VOLHISTORY command. For information about the volume history file, see "Saving the Volume History File" on page 656.

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 24, "Using Disaster Recovery Manager," on page 687.

Maintaining a Supply of Scratch Volumes

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.

Ensure that you set the maximum number of scratch volumes high enough for the expected usage. When you exceed this number, 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 176.

Keep in mind that you may need additional volumes for potential recovery operations in the future, so it is recommended that you label and set aside extra scratch volumes.

For automated libraries, see also "Maintaining a Supply of Scratch Volumes in an Automated Library" on page 182.

Managing Volumes in Automated Libraries

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. For details on the check-in procedure, see "Checking New Volumes into a Library" on page 171.

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 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

Changing the Status of a Volume

The UPDATE LIBVOLUME command lets you change the status of a volume in an automated library from scratch to private, or private to scratch. However, 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 may want to remove a volume from an automated library. The following two examples illustrate this:

- You have exported data to a volume in the library and want to take it to another system for an import operation.
- All of the volumes in the library are full, and you want to remove some that are not likely to be accessed to make room for new volumes that can be used to store more data.

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.

Returning Volumes to a Library

When you check out a volume that is defined to a storage pool, to make the volume available again, do the following:

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 may exceed the physical capacity of the library. To make room for new volumes while keeping track of existing volumes, you can define a storage pool overflow location near the library. You then move media to the overflow location as needed. The following shows a typical sequence of steps to manage a full library:

1. Define or update the storage pool associated with the automated library, including the overflow location parameter. For example, you have a storage pool named ARCHIVEPOOL associated with an automated library. Update the storage pool to add an overflow location of Room2948. Enter this command:

   ```
   update stgpool archivepool ovflocation=Room2948
   ```

2. When the library becomes full, move the full volumes out of the library and to the overflow location that you defined for the storage pool. For example, to move all full volumes in the specified storage pool out of the library, enter this command:

   ```
   move media * stgpool=archivepool
   ```

   All full volumes are checked out of the library. Tivoli Storage Manager records the location of the volumes as Room2948. You can use the DAYS parameter to specify the number of days that must elapse before a volume is eligible for processing by the MOVE MEDIA command.
3. Check in new scratch volumes, if needed. If a volume has an entry in volume history, you cannot check it in as a scratch volume.

4. Reuse the empty scratch storage volumes in the overflow location. For example, enter this command:

```
query media * stg=* whereovflocation=Room2948 wherestatus=empty
move media * stg=* wherestate=mountablenotinlib wherestatus=empty
cmd="checkin libvol autolib &vol status=scratch"
cmdfilename=/tsm/move/media/checkin.vols
```

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

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

Task	Required Privilege Class
Audit the volume inventory of a library	System or unrestricted storage

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 database problems. Use 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 (through the device class), you can specify a maximum number of scratch volumes equal to the physical capacity of the library. When the number of scratch volumes that Tivoli Storage Manager is using for the storage pool exceeds that number, do the following:

1. Add scratch volumes to the library by checking in volumes. Label them if necessary.

   You may need to use an overflow location to move volumes out of the library to make room for these scratch volumes. See "Maintaining a Supply of Scratch Volumes" on page 178.
2. Increase the maximum number of scratch volumes by updating the storage pool definition. The increase should equal the number of scratch volumes that you checked in.

Keep in mind that you may need additional volumes for potential recovery operations in the future, so it is recommended that you label and set aside extra scratch volumes.

Performing Operations with Shared Libraries

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 12 on page 183 shows the interaction between library clients and the library manager in processing Tivoli Storage Manager operations.

Table 12. How SAN-enabled Servers Process Tivoli Storage Manager Operations

Operation **(Command)**	**Library Manager**	**Library Client**
Query library volumes (QUERY LIBVOLUME)	Displays the volumes that are checked into the library. For private volumes, the owner server is also displayed.	Not applicable.
Check in and check out library volumes (CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME)	Performs the commands to the library device.	Not applicable. When a check-in 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 check-in of media.	Not applicable.
Dismount a volume (DISMOUNT VOLUME)	Sends the request to the library device.	Requests that the library manager server perform the operation.
Query a volume (QUERY VOLUME)	Checks whether the volume is owned by the requesting library client server and checks whether the volume is in the library device.	Requests that the library manager server perform the operation.

Managing Server Requests for Media

Tivoli Storage Manager displays requests and status messages to all administrative clients that are started in console mode. These request messages often have a time limit. If the request is not fulfilled within the time limit, the operation times out and fails.

For manual libraries, Tivoli Storage Manager detects when there is a cartridge loaded in a drive, and no operator reply is necessary. For automated libraries, the CHECKIN LIBVOLUME and LABEL LIBVOLUME commands involve inserting cartridges into slots and, depending on the value of the WAITTIME parameter, issuing a reply message. (If the value of the parameter is zero, no reply is required.) The CHECKOUT LIBVOLUME command involves inserting cartridges into slots and, in all cases, issuing a reply message.

Using the Administrative Client for Mount Messages

The server sends mount request status messages to the server console and to all administrative clients in mount mode or console mode parameter. For example, to start an administrative client in mount mode, enter this command:

```
> dsmadmc -mountmode
```

Mount Operations for Manual Libraries

Volumes are mounted as a result of mount requests from Tivoli Storage Manager. For manual libraries, you can monitor the mount requests on the server console or through an administrative client in mount mode or console mode. Someone you designate as the operator must respond to the mount requests by putting in tape volumes as requested.

Handling Messages for Automated Libraries

For automated libraries, mount messages are sent to the library and not to an operator. Messages about problems with the library are sent to the mount message queue. You can see these messages on administrative clients in mount mode or console mode. However, you cannot use the Tivoli Storage Manager REPLY command to respond to these messages.

Requesting Information about Pending Operator Requests

Task	Required Privilege Class
Request information about operator requests or mounted volumes	Any administrator

You can get information about pending operator requests either by using the QUERY REQUEST command or by checking the mount message queue on an administrative client started in mount mode.

When you issue the QUERY REQUEST command, Tivoli Storage Manager displays requested actions and the amount of time remaining before the requests time out. For example, you enter the command as follows:

```
query request
```

The following shows an example of a response to the command:

```
ANR8352I Requests outstanding:
ANR8326I 001: Mount 8MM volume DSM001 R/W in drive TAPE01 (/dev/tsmscsi/mt1)
of MANUAL8MM within 60 minutes.
```

Replying to Operator Requests

Task	Required Privilege Class
Reply to operator requests	Operator

When the server requires that an explicit reply be provided when a mount request is completed, you can reply with the REPLY command. The first parameter for this command is the request identification number that tells the server which of the pending operator requests has been completed. This 3-digit number is always

displayed as part of the request message. It can also be obtained by issuing a QUERY REQUEST command. If the request requires the operator to provide a device to be used for the mount, the second parameter for this command is a device name.

For example, enter the following command to respond to request 001 for tape drive TAPE01:

```
reply 1
```

Canceling an Operator Request

Task	Required Privilege Class
Cancel operator requests	Operator

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 cause the operation that needed the requested volume to fail.

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, as described in "Requesting Information about Pending Operator Requests" on page 184.

You can specify the PERMANENT parameter if you want to mark the requested volume as UNAVAILABLE. This process is useful if, for example, the volume has been moved to a remote site or is otherwise inaccessible. By specifying PERMANENT, you ensure that the server does not try to mount the requested volume again.

For most of the requests associated with automated (SCSI) libraries, an operator must perform a hardware or system action to cancel the requested mount. For such requests, the CANCEL REQUEST command is not accepted by the server.

Responding to Requests for Volume Check-In

If the server cannot find a particular volume to be mounted in an automated library, the server requests that the operator check in the volume. For example, a client requests that an archived file be retrieved. The file was archived in a storage pool in an automated library. The server looks for the volume containing the file in the automated library, but cannot find the volume. The server then requests that the volume be checked in.

If the volume that the server requests is available, put the volume in the library and check in the volume using the normal procedures ("Checking New Volumes into a Library" on page 171).

If the volume requested is unavailable (lost or destroyed), update the access mode of the volume to UNAVAILABLE by using the UPDATE VOLUME command. Then cancel the server's request for check-in by using the CANCEL REQUEST command. (Do *not* cancel the client process that caused the request.) To get the ID of the request to cancel, use the QUERY REQUEST command.

If you do not respond to the server's check-in request within the mount-wait period of the device class for the storage pool, the server marks the volume as unavailable.

Determining Which Volumes Are Mounted

Task	Required Privilege Class
Request information about which volumes are mounted	Operator

For a report of all volumes currently mounted for use by the server, you can 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.

Dismounting an Idle Volume

Task	Required Privilege Class
Request a volume dismount	Operator

After a volume becomes idle, the server keeps it mounted for a time specified by the mount retention parameter for the device class. Use of mount retention can reduce the access time if volumes are repeatedly used.

An administrator can explicitly request to dismount an idle volume by issuing the DISMOUNT VOLUME command. This command causes the server to dismount the named volume from the drive in which it is currently mounted.

For information about setting mount retention times, see "Mount Retention Period" on page 201.

Managing Libraries

You can query, update, and delete libraries.

Requesting Information About Libraries

Task	Required Privilege Class
Request information about libraries	Any administrator

You can request information about one or more libraries by using the QUERY LIBRARY command. 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.

Note: You cannot update a MANUAL library.

Task	Required Privilege Class
Update libraries	System or unrestricted storage

Automated Libraries

If your system or device is reconfigured, and the device name changes, you may need to update the device name. The examples below show how you can use the UPDATE LIBRARY and UPDATE PATH commands for the following library types:

- SCSI
- 349X
- ACSLS
- External

Examples:

- **SCSI Library**

  Update the path from SERVER1 to a SCSI library named SCSILIB:

  ```
  update path server1 scsilib srctype=server desttype=library
  device=/dev/tsmscsi/lb1
  ```

  Update the definition of a SCSI library named SCSILIB defined to a library client so that a new library manager is specified:

  ```
  update library scsilib primarylibmanager=server2
  ```

- **349X Library**

  Update the path from SERVER1 to an IBM 3494 library named 3494LIB with new device names.

  ```
  update path server1 3494lib srctype=server desttype=library
   device=library2,library3,library4
  ```

  Update the definition of an IBM 3494 library named 3494LIB defined to a library client so that a new library manager is specified:

  ```
  update library 3494lib primarylibmanager=server2
  ```

- **ACSLS Library**

  Update an 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

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. See "Deleting Drives" on page 193.

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
```

Managing Drives

You can query, update, clean, and delete drives.

Requesting Information about Drives

Task	Required Privilege Class
Request information about drives	Any administrator

You can request information about drives by using the QUERY DRIVE command. This 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 by issuing the UPDATE DRIVE command.

Task	Required Privilege Class
Update drives	System or unrestricted storage

You can change the following attributes of a drive by issuing the UPDATE DRIVE command.

- The element address, if the drive resides in a SCSI library
- The ID of a drive in an ACSLS library
- The cleaning frequency
- Change whether the drive is online or offline

For example, to change the element address of a drive named DRIVE3 to 119, issue the following command:

```
update drive auto drive3 element=119
```

If you are reconfiguring your system, you can change the device name of a drive by issuing the UPDATE PATH command. For example, to change the device name of a drive named DRIVE3, issue the following command:

```
update path server1 drive3 srctype=server desttype=drive library=scsilib
 device=/dev/IBMtape0
```

Note: You cannot change the element number or the device name if a drive is in use. See "Taking Drives Offline." If a drive has a volume mounted, but the volume is idle, it can be explicitly dismounted. See "Dismounting an Idle Volume" on page 186.

Taking Drives Offline

You can take a drive offline while it is in use. For example, you might take a drive offline for another activity, such as maintenance. If you take a drive offline while it is in use, the mounted volume completes its current process. If this volume was part of a series of volumes in a transaction, the drive is no longer available to complete mounting the series. If no other drives are available, the active process may fail. The offline state is retained even if the server is halted and brought up again. If a drive is marked offline when the server is brought up, a warning is issued noting that the drive must be manually brought online. If all the drives in a library are taken offline, processes requiring a library mount point will fail, rather than queue up for one.

The ONLINE parameter specifies the value of the drive's online state, even if the drive is in use. ONLINE=YES indicates that the drive is available for use. ONLINE=NO indicates that the drive is not available for use (offline). Do not specify other optional parameters along with the ONLINE parameter. If you do, the drive will not be updated, and the command will fail when the drive is in use. You can specify the ONLINE parameter when the drive is involved in an active process or session, but this is not recommended.

Cleaning Drives

Task	Required Privilege Class
Clean drives	System or unrestricted storage

The server can control cleaning tape drives in SCSI libraries and offers partial support for cleaning tape drives in manual libraries. For automated library devices, you can automate cleaning by specifying the frequency of cleaning operations and checking a cleaner cartridge into the library's volume inventory. Tivoli Storage Manager mounts the cleaner cartridge as specified. For manual library devices, Tivoli Storage Manager issues a mount request for the cleaner cartridge.

Deciding Whether the Server Controls Drive Cleaning

Some SCSI libraries provide automatic drive cleaning. In such cases, choose either the library drive cleaning or the Tivoli Storage Manager drive cleaning, but not both. Manufacturers that include library cleaning recommend its use to prevent premature wear on the read/write heads of the drives.

Drives and libraries from different manufacturers differ in how they handle cleaner cartridges and how they report the presence of a cleaner cartridge in a drive. The device driver may not be able to open a drive that contains a cleaner cartridge. Sense codes and error codes that are issued by devices for drive cleaning vary. Library drive cleaning is usually transparent to all applications. Therefore, Tivoli

Storage Manager may not always detect cleaner cartridges in drives and may not be able to determine when cleaning has begun.

Some devices require a small amount of idle time between mount requests to start drive cleaning. However, Tivoli Storage Manager tries to minimize the idle time for a drive. The result may be to prevent the library drive cleaning from functioning effectively. If this happens, try using Tivoli Storage Manager to control drive cleaning. Set the frequency to match the cleaning recommendations from the manufacturer.

If you have Tivoli Storage Manager control drive cleaning, disable the library drive cleaning function to prevent problems. If the library drive cleaning function is enabled, some devices automatically move any cleaner cartridge found in the library to slots in the library that are dedicated for cleaner cartridges. An application does not know that these dedicated slots exist. You will not be able to check a cleaner cartridge into the Tivoli Storage Manager library inventory until you disable the library drive cleaning function.

Cleaning Drives in an Automated Library

Set up server-controlled drive cleaning in an automated library with these steps:

1. Define or update the drives in a library, using the CLEANFREQUENCY parameter. The CLEANFREQUENCY parameter sets how often you want the drive cleaned. Refer to the DEFINE DRIVE and UPDATE DRIVE commands. Consult the manuals that accompany the drives for recommendations on cleaning frequency.

   Note: The CLEANFREQUENCY parameter is not valid for externally managed libraries, for example, 3494 libraries or StorageTek libraries managed under 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 the drive manufacturer's information for cleaning recommendations. If the information gives recommendations for cleaning frequency in terms of hours of use, convert to a gigabytes value by doing the following:

   a. Use the bytes-per-second rating for the drive to determine a gigabytes-per-hour value.
   b. Multiply the gigabytes-per-hour value by the recommended hours of use between cleanings.
   c. Use the result as the cleaning frequency value.

   Note: For IBM 3590 and 3592 drives, we recommend that you specify a value for the CLEANFREQUENCY parameter rather than specify ASNEEDED. Using the cleaning frequency recommended by the product documentation will not overclean the drives.

2. Check a cleaner cartridge into the library's volume inventory with the CHECKIN LIBVOLUME command. For example:

   ```
   checkin libvolume autolib1 cleanv status=cleaner cleanings=10 checklabel=no
   ```

   After the cleaner cartridge is checked in, the server will mount the cleaner cartridge in a drive when the drive needs cleaning. The server will use that cleaner cartridge for the number of cleanings specified. See "Checking In Cleaner Cartridges" on page 191 and "Operations with Cleaner Cartridges in a Library" on page 191 for more information.

For details on the commands, see the *Administrator's Reference*.

Checking In Cleaner Cartridges: You must check a cleaner cartridge into an automated library's volume inventory to have the server control drive cleaning without further operator intervention.

It is recommended that you check in cleaner cartridges one at a time and do not use the search function of check-in for a cleaner cartridge.

Note: 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: 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 cancelled, or if no cleaner cartridge is available, then the indication that the drive

needs cleaning is lost. Monitor cleaning messages for these problems to ensure that drives are cleaned as needed. 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 more than one cleaner cartridge, the server uses one of them for its designated number of cleanings. Then the server begins to use the next cleaner 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.

Cleaning Drives in a Manual Library

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 181.

Deleting Drives

You can delete a drive by issuing the DELETE DRIVE command.

Task	Required Privilege Class
Delete drives	System or unrestricted storage

A drive cannot be deleted if it is currently in use. If a drive has a volume mounted, but the volume is currently idle, it can be dismounted as described in "Dismounting an Idle Volume" on page 186. 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

You can query, update, and delete paths.

Requesting Information About Paths

By using the QUERY PATH command, you can obtain information about paths. You can request either a standard or a detailed report. This command accepts wildcard characters for both a source name and a destination name. See the *Administrator's Reference* for information about this command and the use of wildcard characters.

For example, to display information about all paths, issue the following command:

```
query path
```

The following shows an example of the output from this command.

```
Source Name  Source Type  Destination Name  Destination Type  Online
-----------  -----------  ----------------  ----------------  ------
SERVER1      server       TSMLIB            Library           Yes
NETAPP1      Data mover   DRIVE1            Drive             Yes
NETAPP1      Data mover   NASLIB            Library           Yes
datamover2   Data mover   drive4            Drive             Yes
```

Updating Paths

You can update an existing path by issuing the UPDATE PATH command. The examples below show how you can use the UPDATE PATH commands for the following 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/tsmscsi/lb1
  ```

- **Drive Paths**

  Update the path to change the device name for a drive named NASDRV1:

  ```
  update path nas1 nasdrv1 srctype=datamover desttype=drive
   library=naslib device=/dev/tsmscsi/mt1
  ```

Deleting Paths

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

You can query, update, and delete data movers.

Requesting Information About Data Movers

By using the QUERY DATAMOVER command, you can obtain information about SCSI and NAS data movers. 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 by issuing the UPDATE DATAMOVER command.

For example, to update the data mover for the node named NAS1 to change the IP address, issue the following command:

```
update datamover nas1 hladdress=9.67.97.109
```

Deleting Data Movers

Before you can delete a data mover, you must delete all paths defined for the data mover.

To delete a data mover named NAS1, issue the following command:

```
delete datamover nas1
```

Handling Tape Alert Messages

Tape alert messages are generated by tape and library devices to report hardware errors. 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. These messages help to determine problems that are not related to the IBM Tivoli Storage Manager server.

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 using the SET TAPEALERTMSG command. You may query tape alert messages by using the QUERY TAPEALERTMSG command.

Chapter 9. Defining Device Classes

A device class represents a device type that can be used by IBM Tivoli Storage Manager. The server uses device class definitions to determine which types of devices and volumes to use to:

- Store backup, archive, or space-managed data in primary storage pools
- Store copies of primary storage pool data in copy storage pools
- Store database backups
- Export or import Tivoli Storage Manager data

One device class can be associated with multiple storage pools, but each storage pool is associated with only one device class.

For random access storage, Tivoli Storage Manager supports only the DISK device class, which is defined by Tivoli Storage Manager. However, you can define many storage pools associated with the DISK device class.

See the following sections:

Tasks:
"Defining and Updating Tape Device Classes" on page 198
"Defining and Updating REMOVABLEFILE Device Classes" on page 203
"Defining and Updating FILE Device Classes" on page 204
"Defining and Updating LTO Device Classes" on page 207
"Defining and Updating SERVER Device Classes" on page 208
"Defining and Updating VOLSAFE Device Classes" on page 210
"Requesting Information about a Device Class" on page 211
"How Tivoli Storage Manager Fills Volumes" on page 213
"Deleting a Device Class" on page 212

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Defining and Updating Device Classes for Sequential Media

Task	Required Privilege Class
Define or update device classes	System or unrestricted storage

For sequential access storage, Tivoli Storage Manager supports the following device types:

© Copyright IBM Corp. 1993, 2005

Device Type	Media Type	Device Examples
3590	IBM 3590 cartridges	IBM 3590, 3590E drives
3592	IBM 3592 cartridges	IBM 3592 drives
4MM	4-mm cartridges	IBM 7206-005
8MM	8-mm cartridges	IBM 7208-001 and 7208-011
DLT	Digital linear tape (DLT) cartridges	DLT2000, DLT4000, DLT7000 and DLT8000 drives
ECARTRIDGE	Tape cartridges	StorageTek SD-3, 9490, 9840, and 9940 drives
FILE	File system or storage volumes	Server
LTO	LTO Ultrium cartridges	IBM 3580
NAS	Unknown	Tape drives supported by the NAS file server for backups
QIC	Quarter-inch tape cartridges	IBM 7207
REMOVABLEFILE	Iomega Zip or Jaz, or CD media	Removable media devices that are attached as local, removable file systems
SERVER	Storage volumes or files archived in another Tivoli Storage Manager server	Tivoli Storage Manager target server
VOLSAFE	Write-once read-many (WORM) tape cartridges	StorageTek 9840 drives

For all device types other than FILE or SERVER, you must define libraries and drives to Tivoli Storage Manager before you define the device classes.

You can define multiple device classes for each device type. For example, you may need to specify different attributes for different storage pools that use the same type of tape drive. Variations may be required that are not specific to the device, but rather to how you want to use the device (for example, mount retention or mount limit).

Tivoli Storage Manager now allows SCSI libraries to include tape drives of more than one device type. When you define the device class in this environment, you must declare a value for the FORMAT parameter. See “Configuration with Multiple Drive Device Types” on page 105 and “Mixing Device Types in Libraries” on page 100 for additional information.

If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command. When you use this option, the files specified are automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

The following sections explain the device classes for each supported device type.

Defining and Updating Tape Device Classes

To use tape devices, you must define a device class by issuing a DEFINE DEVCLASS command with the DEVTYPE parameter.

Other parameters specify how to manage data storage operations involving the new device class:

ESTCAPACITY
See "Estimated Capacity."

FORMAT
See "Recording Format."

LIBRARY
See "Library" on page 200.

MOUNTLIMIT
See "Mount Limit" on page 200.

MOUNTRETENTION
See "Mount Retention Period" on page 201.

MOUNTWAIT
See "Mount Wait Period" on page 201.

PREFIX
See "Tape Label Prefix" on page 202.

SCALECAPACITY
See "Scale Capacity" on page 202.

WORM
See "WORM" on page 202.

You can update the device class by issuing the UPDATE DEVCLASS command.

Estimated Capacity

The ESTCAPACITY parameter specifies the estimated capacity for volumes assigned to this device class. Tivoli Storage Manager estimates the capacity of the volumes in a storage pool based on the parameters assigned to the device class associated with the storage pool. For tape device classes, the default values selected by the server depend on the recording format used to write data to the volume. You can either accept the default for a given device type or specify a value.

Note: For a device class with DEVTYPE=NAS, this value is required.

See the *Administrator's Reference* for information about the estimated capacities of recording formats for each device type.

Tivoli Storage Manager also uses estimated capacity to determine when to begin reclamation storage pool volumes. For more information on how Tivoli Storage Manager uses the estimated capacity value, see "How Tivoli Storage Manager Fills Volumes" on page 213.

Recording Format

The FORMAT parameter specifies the recording format used by Tivoli Storage Manager when writing data to removable media. See the *Administrator's Reference* for information about the recording formats for each device type.

Specify the FORMAT=DRIVE parameter only if all drives associated with that device class are identical. If some drives associated with the device class support a higher density format than others and 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.

Tivoli Storage Manager now supports drives with different tape technologies in a single SCSI library. You must specify a value for the FORMAT parameter in this configuration. See "Configuration with Multiple Drive Device Types" on page 105 for an example.

The recording format that Tivoli Storage Manager uses for a given volume is selected when the first piece of data is written to the volume. Updating the FORMAT parameter does not affect media that already contain data until those media are rewritten from the beginning. This process may happen after a volume is reclaimed or deleted, or after all of the data on the volume expires.

Library

Before the server can mount a volume, it must know which drives can be used to satisfy the mount request. This process is done by specifying the library when the device class is defined. The library must contain 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.

There is no default value for this parameter. It is required, and so must be specified when the device class is defined.

Mount Limit

The MOUNTLIMIT parameter specifies the maximum number of volumes that can be simultaneously mounted for a device class. You can limit the number of drives that the device class has access to at one time with the MOUNTLIMIT parameter.

The default mount limit value is DRIVES. The DRIVES parameter indicates that every time a mount point is allocated, the number of drives online and defined to the library is used to calculate the true mount limit value. The maximum value for this parameter is 4096 and the minimum value is 0. A zero value prevents new transactions from gaining access to the storage pool.

When selecting a mount limit for a device class, be sure to consider the following questions:

- 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 and copy storage pools?

  Specify a mount limit value that provides a sufficient number of mount points to support a simultaneous write to the primary storage pool and all associated copy storage pools.
- Are you associating multiple device classes with a single library?

  A device class associated with a library can use any drive in the library that is compatible with the device class' device type. Because you can associate more than one device class with a library, a single drive in the library can be used by more than one device class. However, Tivoli Storage Manager does not manage how a drive is shared among multiple device classes.

- How many Tivoli Storage Manager processes do you want to run at the same time, using devices in this device class?

  Tivoli Storage Manager automatically cancels some processes to run other, higher priority processes. If the server is using all available drives in a device class to complete higher priority processes, lower priority processes must wait until a drive becomes available. For example, Tivoli Storage Manager cancels the process for a client backing up directly to tape if the drive being used is needed for a server migration or tape reclamation process. Tivoli Storage Manager cancels a tape reclamation process if the drive being used is needed for a client restore operation. For additional information, see "Preemption of Client or Server Operations" on page 467.

  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 primary and copy storage pool simultaneous write function. You must have enough drives available to allow for a successful simultaneous write.

Note: If the library associated with this device class is EXTERNAL type, it is recommended that you explicitly specify the mount limit instead of using MOUNTLIMIT=DRIVES.

Mount Retention Period

The MOUNTRETENTION parameter specifies the amount of time that a mounted volume should remain mounted after its last I/O activity. If this idle time limit is reached, the server dismounts the volume. The default mount retention period is 60 minutes. The maximum value for this parameter is 9999 minutes.

Note: A device class with DEVTYPE=NAS allows only a value of zero (0).

For example, if the mount retention value is 60, and a mounted volume remains idle for 60 minutes, then the server dismounts the volume.

If a volume is used frequently, you can improve performance by setting a longer mount retention period to avoid unnecessary mount and dismount operations.

If mount operations are being handled by manual, operator-assisted activities, you may want to use a large 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.

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 466 and "Dismounting an Idle Volume" on page 186.

Mount Wait Period

The MOUNTWAIT parameter specifies the maximum amount of time, in minutes, that the server waits for a drive to become available for the current mount request. The default mount wait period is 60 minutes. The maximum value for this parameter is 9999 minutes.

Note: This parameter is not valid for EXTERNAL library types.

Tape Label Prefix

By using the PREFIX parameter, you can specify a prefix value that is used to construct the *file name* string that is stored in the label area of each tape volume.

The prefix string is used as the prefix of the file name that is written to the label of each tape. The default value for the tape label prefix string is ADSM.

Note: This parameter is used primarily in the OS/390 and z/OS operating systems.

Scale Capacity

The SCALECAPACITY parameter allows you to reduce media capacity in order to create volumes with faster data access speeds. The percentage values can be set to 20, 90 or 100, where 20 percent provides the fastest access time and 100 percent provides the largest storage capacity.

Note: The SCALECAPACITY parameter is only valid when defining and updating a 3592 device class.

If you select 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. This feature allows you to partition data into storage pools with faster data access speeds. For more information on setting up storage pool hierarchies, see "Setting Up a Storage Pool Hierarchy" on page 233.

The scale capacity can be specified using either the DEFINE or UPDATE DEVCLASS commands; however, the scale capacity will only take effect when data is first written to a volume. Any updates to the device class for scale capacity will not affect volumes that already have data written to them until the volume is returned to scratch status.

See the *Administrator's Reference* for information about defining a 3592 device class.

WORM

The WORM parameter specifies whether the drive being defined is a WORM (write once, read many) device. This parameter is not supported for all device classes.

The following is an example for configuring a VolSafe device using the WORM parameter. You can also use the DEFINE DEVCLASS – VOLSAFE command. Suppose you want to configure a VolSafe device using the WORM parameter and a SCSI library. You would enter the following series of commands. (The values you select for the library variable, the drive variable, and so on may be the same or different for your environment.)

1. Define a library:

   ```
   define library volsafelib libtype=scsi
   ```

2. Define a drive:

   ```
   define drive volsafelib drive01
   ```

3. Define a path:

   ```
   define path server01 drive01 srctype=server destype=drive
   device=/dev/tsmscsi/mt0 library=volsafelib
   ```

4. Define a device class:

```
define devclass volsafeclass library=volsafelib devtype=ecartridge
format=drive worm=yes
```

Note: You cannot change the value of the WORM parameter using the UPDATE DEVCLASS command.

For more information, see "Special Considerations for WORM Tape Media" on page 174.

Defining and Updating REMOVABLEFILE Device Classes

Removable file devices include devices such as Iomega Zip drives, Iomega Jaz drives, and CD drives. Volumes in this device class are sequential access volumes. Define a device class for these devices by issuing the DEFINE DEVCLASS command with the DEVTYPE=REMOVABLEFILE parameter.

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.

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.

The server recognizes that the media can be removed and that additional media can be inserted. This is subject to limits set with the MOUNTLIMIT parameter for the device class and the MAXSCRATCH parameter for the storage pool.

When using CD-ROM media for the REMOVABLEFILE device type, the library type must be specified as MANUAL. Access this media through a mount point, for example, */mnt/cdrom*.

Use the device manufacturer's utilities to format (if necessary) and label the media. The following restrictions apply:

- The label on the media must be no more than 11 characters.
- The label on the media must have the same name for file name and volume label.

See "Configuring Removable File Devices" on page 129 for more information.

Other parameters specify how to manage storage operations involving this device class:

LIBRARY
See "Library" on page 200.

MAXCAPACITY

You can specify a maximum capacity value that restricts the size of volumes (that is, files) associated with a REMOVABLEFILE device class. Use the MAXCAPACITY parameter with the DEFINE DEVCLASS command.

Because the server opens only one file per physical removable medium, specify a value such that the one file makes full use of your media

capacity. 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.

The default MAXCAPACITY value for a REMOVABLEFILE device class is the remaining space in the file system where the removable media volume is added to Tivoli Storage Manager.

The MAXCAPACITY parameter must be set at less value than the capacity of the media. For CD media, the maximum capacity is typically 650 MB or 750 MB.

MOUNTRETENTION
See "Mount Retention Period" on page 201.

MOUNTWAIT
See "Mount Wait Period" on page 201.

You can update the device class information by issuing the UPDATE DEVCLASS command.

Defining and Updating FILE Device Classes

The FILE device type is used for storing data on disk in *simulated* storage volumes. The storage volumes are actually files. Data is written sequentially into standard files in the file system of the server machine. You can define this device class by issuing a DEFINE DEVCLASS command with the DEVTYPE=FILE parameter. Because each volume in a FILE device class is actually a file, a volume name must be a fully qualified file name.

Note: Do not use raw partitions with a device class type of FILE.

When you define or update the FILE device class, you can specify the following parameters:

DIRECTORY
See "Directory."

MAXCAPACITY
See "Maximum Capacity" on page 206.

MOUNTLIMIT
See "Mount Limit" on page 206.

Directory

You can specify one or more directories as the location of the files used in the FILE device class. The default is the current working directory of the server at the time the command is issued, unless the DSMSERV_DIR environment variable is set. For more information on setting the environment variable, refer to the *Installation Guide.*

The directory name identifies the location where the server places the files that represent storage volumes for this device class. While processing the command, the server expands the specified directory name into its fully qualified form, starting from the root directory.

Later, if the server needs to allocate a scratch volume, it creates a new file in 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.

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.

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	.DBB
Database dump and unload	.DMP

Attention: Tivoli Storage Manager supports the use of remote file systems or drives for reading and writing storage pool data, database backups, and other data operations. Disk subsystems and file systems must not report successful write operations when they can fail after a successful write report to Tivoli Storage Manager. A write failure after a successful notification constitutes a data integrity problem because the data that was reported as successfully written is unavailable for retrieval. In this situation, all data subsequently written is also at risk due to positioning mismatches within the target file. To avoid these problems, ensure that disk subsystems and file systems, whatever implementation you use, are *always* able to return data when the data is requested.

For important disk-related information, see "Requirements for Disk Subsystems, File Systems and Remote File Access" on page 83.

You must ensure that storage agents can access newly created FILE volumes. To access FILE volumes, storage agents replace names from the directory list in the device-class definition with the names in the directory list for the associated path definition. The following illustrates the importance of matching device classes and paths to ensure that storage agents can access newly created FILE volumes.

Suppose you want to use these three directories for a FILE library:

/opt/tivoli1

/opt/tivoli2

/opt/tivoli3

1. You use the following command to set up a FILE library named CLASSA with one drive named CLASSA1 on SERVER1:

   ```
   define devclass classa devtype=file
   directory="/opt/tivoli1,/opt/tivoli2,/opt/tivoli3"
   shared=yes mountlimit=1
   ```

2. You want the storage agent STA1 to be able to use the FILE library, so you define the following path for storage agent STA1:

   ```
   define path server1 sta1 srctype=server desttype=drive device=file
   directory="/opt/ibm1,/opt/ibm2,/opt/ibm3" library=classa
   ```

In this scenario, the storage agent, STA1, will replace the directory name /opt/tivoli1 with the directory name /opt/ibm1/ to access FILE volumes that are in the /opt/tivoli1 directory on the server.

3. If file volume /opt/tivoli1/file1.dsm is created on SERVER1, and if the following command is issued,

   ```
   update devclass classa directory="/opt/otherdir,/opt/tivoli2,
   /opt/tivoli3"
   ```

   SERVER1 will still be able to access file volume /opt/tivoli1/file1.dsm, but the storage agent STA1 will not be able to access it because a matching directory name in the PATH directory list no longer exists. If a directory name is not available in the directory list associated with the device class, the storage agent can lose access to a FILE volume in that directory. Although the volume will still be accessible from the 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.

Maximum Capacity

You can specify a maximum capacity value that restricts the size of volumes (that is, files) associated with a FILE device class. Use the MAXCAPACITY parameter of the DEFINE DEVCLASS command. 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.

The default MAXCAPACITY value for a FILE device class is 2GB.

Mount Limit

The mount limit value for FILE device classes is used to restrict the number of mount points (volumes or files) that can be concurrently opened for access by server storage and retrieval operations. Any attempts to access more volumes than indicated by the mount limit causes the requester to wait. The default value is 20. The maximum value for this parameter is 4096.

Note: The MOUNTLIMIT=DRIVES parameter is not valid for the FILE device class.

When selecting a mount limit for this device class, consider how many Tivoli Storage Manager processes you want to run at the same time.

Tivoli Storage Manager automatically cancels some processes to run other, higher priority processes. If the server is using all available mount points in a device class to complete higher priority processes, lower priority processes must wait until a mount point becomes available. For example, Tivoli Storage Manager cancels the process for a client backup if the mount point being used is needed for a server migration or reclamation process. Tivoli Storage Manager cancels a reclamation process if the mount point being used is needed for a client restore operation. For additional information, see “Preemption of Client or Server Operations” on page 467.

If processes are often cancelled 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.

Defining and Updating LTO Device Classes

The LTO device class supports LTO Ultrium drives and media. When working with the LTO device class, you need to consider the following parameters:

LIBRARY
See "Library" on page 200.

ESTCAPACITY
See "Estimated Capacity" on page 199.

MOUNTRETENTION
See "Mount Retention Period" on page 201.

MOUNTWAIT
See "Mount Wait Period" on page 201.

PREFIX
See "Tape Label Prefix" on page 202.

WORM
See "WORM" on page 202.

In addition to the parameters listed above, you also need to consider the FORMAT and MOUNTLIMIT parameters for which there are special considerations when mixing LTO Ultrium media and drives.

Format

If you are considering mixing different generations of LTO media and drives, be aware of the following restrictions:

- LTO Generation 1 drives can read and write Generation 1 media only.
- LTO Generation 2 drives can read and write both Generation 1 and Generation 2 media.
- LTO Generation 3 drives can read and write both Generation 2 and Generation 3 media. However, LTO Generation 3 drives can only read Generation 1 media.

If you are mixing different types of drives and media, you should configure different types of device classes: one for each type of media. Use the FORMAT parameter in each of the device class definitions to specify the exact media type. (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.

Note: If you are mixing Ultrium Generation 1 and Ultrium Generation 3 drives and media in a single library, you need to mark the Generation 1 media as read-only and all Generation 1 scratch volumes need to be checked out.

Mount Limit

In a mixed media library, in which multiple device classes point to the same library, compatible drives will be shared between storage pools. Special attention

needs to be paid to setting an appropriate value for the MOUNTLIMIT parameter in each of the device classes. For example, in a mixed media library containing Ultrium Generation 1 and Ultrium Generation 2 drives and media, Ultrium Generation 1 media can get mounted in Ultrium Generation 2 drives.

Consider the example of a mixed library that consists of the following:

- Four LTO Ultrium Generation 1 drives
- LTO Ultrium Generation 1 media
- Four LTO Ultrium Generation 2 drives
- LTO Generation 2 media
- LTO Ultrium Generation 1 device LTO1CLASS specifying FORMAT=ULTRIUMC
- LTO Ultrium Generation 2 device LTO2CLASS specifying FORMAT=ULTRIUM2C
- LTO Ultrium Generation 1 storage pool LTO1POOL based on device class LTO1CLASS
- LTO Ultrium Generation 2 storage pool LTO1POOL 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.

As part of standard operational management, the MOUNTLIMIT setting needs to be monitored and adjusted 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 and Ultrium Generation 2 drives because of the way in which mount points are allocated. 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.

Defining and Updating SERVER Device Classes

The SERVER device type is used for special device classes whose storage volumes are not directly attached to this server. A volume with device type SERVER consists of one or more files archived in the server storage of another server, called a target server. You can define this device class by issuing a 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 601.

The following parameters specify how to manage data storage operations for the new device class:

MAXCAPACITY
“Maximum Capacity.”

MOUNTLIMIT
“Mount Limit.”

MOUNTRETENTION
See “Mount Retention” on page 210.

PREFIX
See “Prefix” on page 210.

RETRYINTERVAL
See “Retry Interval” on page 210.

RETRYPERIOD
See “Retry Period” on page 210.

SERVERNAME
See “Server Name” on page 210.

You can update the device class information by issuing the UPDATE DEVCLASS command.

Maximum Capacity

You can specify a maximum capacity value that restricts the size of files that are created on the target server to store data for the source server. The default MAXCAPACITY value is 500 MB. The storage pool volumes of this device type are explicitly set to full when the volume is closed and dismounted.

Mount Limit

Use the mount limit value for SERVER device classes to restrict 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. The default mount limit value is 1. The maximum value for this parameter is 4096.

Note: The MOUNTLIMIT=DRIVES parameter is not valid for the SERVER device class.

When selecting 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.

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 cancelled 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.

Mount Retention

You can specify the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. The default value is 60. The maximum value you can specify for this parameter is 9999. A value of 1 to 5 minutes is recommended. This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

Prefix

You can specify a prefix that the source server will use as the beginning portion of the high-level archive file name on the target server.

Retry Interval

You can specify how often the source server tries to connect to the target server when there is a communications failure. During the retry period, the source server tries to connect again as often as indicated by the retry interval.

Retry Period

You can specify a retry period for communications with the target server. When there is a communications failure, this period determines the amount of time during which the source server continues to attempt to connect to the target server.

Server Name

The Tivoli Storage Manager server on which you define a SERVER device class is called a source server. The source server uses the SERVER device class to store data on another Tivoli Storage Manager server, called a target server.

When defining a SERVER device class, specify the name of the target server. The target server must already be defined by using the DEFINE SERVER command. See “Using Virtual Volumes to Store Data on Another Server” on page 601 for more information.

Defining and Updating VOLSAFE Device Classes

To use StorageTek VolSafe brand media and drives, you can define a device class by issuing the DEFINE DEVCLASS command with a DEVTYPE=VOLSAFE parameter. 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.

You can also define a VolSafe device by issuing the DEFINE DEVCLASS command with a DEVTYPE=ECARTRIDGE parameter and by setting the WORM parameter to WORM=YES. For VolSafe devices, WORM=YES is required and must be specified when the device class is defined. The WORM parameter cannot be updated using the UPDATE DEVCLASS command. For more information about VolSafe media, see “Special Considerations for WORM Tape Media” on page 174.

Consult your StorageTek hardware documentation to enable VolSafe function on the drives. Attempting to write to VolSafe media without a VolSafe-enabled drive results in errors.

You can use this device class with EXTERNAL, SCSI, and ACSLS libraries. All drives in a library must be enabled for VolSafe use.

Other parameters specify how to manage data storage operations involving this device class:

ESTCAPACITY
See "Estimated Capacity" on page 199.

FORMAT
See "Recording Format" on page 199.

LIBRARY
If any drives in a library are VolSafe-enabled, all drives in the library must be VolSafe-enabled. Consult your hardware documentation to enable VolSafe function on the StorageTek 9840 drives. Attempting to write to VolSafe media without a VolSafe-enabled drive results in errors. The media needs to be loaded into a drive during the check-in process to determine whether it is WORM or read-write.

Only one library can be associated with a given device class. However, multiple device classes can reference the same library. Unless you are using the DRIVES value for MOUNTLIMIT, you must ensure that the numeric value of the mount limits of all device classes does not exceed the number of drives defined in the referenced library.

This parameter is required and must be specified when the device class is defined.

MOUNTLIMIT
See "Mount Limit" on page 200.

MOUNTRETENTION
See "Mount Retention Period" on page 201.

MOUNTWAIT
See "Mount Wait Period" on page 201.

You can update the device class information by issuing the UPDATE DEVCLASS command.

Requesting Information about a Device Class

You can choose to view a standard or detailed report for a device class.

Task	Required Privilege Class
Request information about device classes	Any administrator

To display a standard report on device classes, enter:

```
query devclass
```

Figure 15 on page 212 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 15. Example of a Standard Device Class Report

To display a detailed report on the GEN1 device class, enter:

```
query devclass gen1 format=detailed
```

Figure 16 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 16. Example of a Detailed Device Class Report

Deleting a Device Class

Task	Required Privilege Class
Delete a device class	System or unrestricted storage

You can delete a device class with the DELETE DEVCLASS command when:

- No storage pools are assigned to the device class. For information on deleting storage pools, see "Deleting a Storage Pool" on page 303.
- The device class is not being used by an export or import process.

Note: You cannot delete the DISK device class from the server.

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. 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.

Using 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	Files that have already been compressed by the Tivoli Storage Manager client can become larger

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 312 and “Registering Nodes with the Server” on page 310.

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 17 on page 215 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 281.

Figure 17. 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 312 and "Registering Nodes with the Server" on page 310.

Chapter 10. Managing Storage Pools and Volumes

When you configure devices so that the IBM Tivoli Storage Manager server can use them to store client data, you create storage pools and storage volumes. This section gives you overviews and details on storage pools and storage volumes.

The procedures in Chapter 4, "Using Magnetic Disk Devices," on page 83 and Chapter 6, "Configuring Storage Devices," on page 99 show you how to set up and use devices to provide Tivoli Storage Manager with server storage. The procedures use the set of defaults that Tivoli Storage Manager provides for storage pools and volumes. The defaults can work well, but you may have specific requirements not met by the defaults. Three common reasons to change the defaults are the following:

- 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 Table 14 on page 225.

Concepts:
"Overview: Storage Pools" on page 218
"Overview: Volumes in Storage Pools" on page 226
"Access Modes for Storage Pool Volumes" on page 231
"Overview: The Storage Pool Hierarchy" on page 232
"Migration of Files in a Storage Pool Hierarchy" on page 238
"Using Cache on Disk Storage Pools" on page 248
"Using the Simultaneous Write Function to Protect Your Data" on page 249
"Keeping a Client's Files Together: Collocation" on page 258
"Reclaiming Space in Sequential Access Storage Pools" on page 267
"Estimating Space Needs for Storage Pools" on page 277

Tasks:
"Defining or Updating Primary Storage Pools" on page 220
"Task Tips for Storage Pools" on page 225
"Preparing Volumes for Random Access Storage Pools" on page 228
"Preparing Volumes for Sequential Access Storage Pools" on page 229
"Defining Storage Pool Volumes" on page 229
"Updating Storage Pool Volumes" on page 230
"Setting Up a Storage Pool Hierarchy" on page 233
"Monitoring Storage Pools and Volumes" on page 279
"Monitoring the Use of Storage Pool Volumes" on page 281

© Copyright IBM Corp. 1993, 2005

Tasks:
“Moving Files from One Volume to Another Volume” on page 294
“Moving Data for a Client Node” on page 298
“Renaming a Storage Pool” on page 300
“Defining a Copy Storage Pool” on page 301
“Deleting a Storage Pool” on page 303
“Deleting Storage Pool Volumes” on page 304

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see “Managing Servers with the New Administration Center” on page 36.

Overview: Storage Pools

A storage volume is the basic unit of storage, such as allocated space on a disk or a single tape cartridge. A storage pool is a collection of storage volumes. The server uses the storage volumes to store backed-up, archived, or space-managed files. The group of storage pools that you set up for the Tivoli Storage Manager server to use is called *server storage*. Storage pools can be arranged in a storage hierarchy.

The server has two types of storage pools that serve different purposes: primary storage pools and copy storage pools.

Primary Storage Pool

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.

A primary storage pool can use random access storage (DISK device class) or sequential access storage (for example, tape or FILE device classes).

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)

The server does not require separate storage pools for archived, backed-up, or space-managed files. However, you may want to have a separate storage pool for

space-managed files. Clients are likely to require fast access to their space-managed files. Therefore, you may want to have those files stored in a separate storage pool that uses your fastest disk storage.

Copy Storage Pool

When an administrator backs up a primary storage pool, the data is stored in a copy storage pool. See "Backing Up Storage Pools" on page 648 for details.

A copy storage pool can use only sequential access storage (for example, a tape device class or FILE device class).

The copy storage pool provides 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 obtains the file from a copy storage pool. For details, see "Restoring Storage Pools" on page 666, "Backing Up Storage Pools" on page 648, "Recovering a Lost or Damaged Storage Pool Volume" on page 683, and "Maintaining the Integrity of Files" on page 678.

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.

Note: You cannot back up data from a primary storage pool defined with the NETAPPDUMP, the CELERRADUMP or the NDMPDUMP data format.

An Example of Server Storage

Figure 18 on page 220 shows one way to set up server storage. In this example, the storage defined for the server includes:

- Three disk storage pools, which are primary storage pools: ARCHIVE, BACKUP, and HSM
- One primary storage pool that consists of tape cartridges
- One copy storage pool that consists of tape cartridges

Policies defined in management classes direct the server to store files from clients in the ARCHIVE, BACKUP, or HSM disk storage pools. 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 may go directly to tape. For more information about setting up a storage hierarchy, see "Overview: The Storage Pool Hierarchy" on page 232.

You can back up all four of the primary storage pools to the one copy storage pool. For more information on backing up primary storage pools, see "Backing Up Storage Pools" on page 648.

Figure 18. Example of Server Storage

To set up this server storage hierarchy, do the following:

1. Define the three disk storage pools, or use the three default storage pools that are defined when you install the server. Add volumes to the disk storage pools if you have not already done so.

   See "Configuring Random Access Volumes on Disk Devices" on page 87.
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 360 for details.
3. 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

   See Chapter 6, "Configuring Storage Devices," on page 99 for more information. For detailed information on defining a storage pool, see "Defining or Updating Primary Storage Pools."
4. 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 225.
5. Define a copy storage pool. This storage pool can use the same tape device or a different tape device as the primary tape storage pool. See "Defining a Copy Storage Pool" on page 301.
6. Set up administrative schedules or a script to back up the disk storage pools and the tape storage pool to the copy storage pool. Send the volumes offsite for safekeeping. See "Backing Up Storage Pools" on page 648.

Defining or Updating Primary Storage Pools

This section provides a summary of parameters you can set and change for storage pools. The section also provides examples of defining and updating storage pools.

Note: When defining or updating storage pools that use LTO Ultrium media, special considerations might apply. For details, see "Defining and Updating LTO Device Classes" on page 207.

Task	Required Privilege Class
Define storage pools	System
Update storage pools	System or unrestricted storage

When you define a primary storage pool, be prepared to provide some or all of the information that is shown in Table 13. 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 13. 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. Once you define a storage pool, you cannot change whether it is a primary or a copy storage pool.	random, sequential
Maximum number of scratch volumes *(Required for sequential access)*	When you specify a value greater than zero, the server dynamically acquires scratch volumes when needed, up to this maximum number. For automated libraries, set this value equal to the physical capacity of the library. See "Maintaining a Supply of Scratch Volumes in an Automated Library" on page 182.	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

Table 13. Information for Defining a Storage Pool (continued)

Information	Explanation	Type of Storage Pool
Maximum file size [1]	To exclude large files from a storage pool, set a maximum file size. The maximum file size applies to the size of a physical file (a single client file or an aggregate of client files). Do not set a maximum file size for the last storage pool in the hierarchy unless you want to exclude very large files from being stored in server storage.	random, sequential
Cyclic Redundancy Check (CRC) [1]	Specifies whether the server uses CRC to validate storage pool data during audit volume processing. For additional information see "Data Validation During Audit Volume Processing" on page 673. Using the CRC option in conjunction with scheduling audit volume processing continually ensures the integrity of data stored in your storage hierarchy. If you always want your storage pool data validated, set your primary storage pool crcdata definition to YES.	random, sequential
Name of the next storage pool [1]	Specifies the name of the next storage pool in the storage pool hierarchy, where files can be migrated or where files are stored that exceed the maximum size for this storage pool. See "Overview: The Storage Pool Hierarchy" on page 232.	random, sequential
Migration thresholds [1]	Specifies a percentage of storage pool occupancy at which the server begins migrating files to the next storage pool (high threshold) and the percentage when migration stops (low threshold). See "Migration of Files in a Storage Pool Hierarchy" on page 238.	random, sequential
Migration processes [1]	Specifies the number of concurrent processes to use for migrating files from this storage pool. See "Migration for Disk Storage Pools" on page 239 and "Specifying Multiple Concurrent Migration Processes" on page 247.	random, sequential
Migration delay [1]	Specifies whether migration of files should be delayed for a minimum number of days. See "Keeping Files in a Storage Pool" on page 243 and "How IBM Tivoli Storage Manager Migrates Data from Sequential Access Storage Pools" on page 245.	random, sequential
Continue migration process [1]	Specifies whether migration of files should continue even if files do not meet the requirement for migration delay. This setting is used only when the storage pool cannot go below the low migration threshold without moving additional files. See "Keeping Files in a Storage Pool" on page 243 and "How IBM Tivoli Storage Manager Migrates Data from Sequential Access Storage Pools" on page 245.	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 "Using Cache on Disk Storage Pools" on page 248.	random
Collocation	With collocation enabled, the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of sequential access storage volumes. See "Keeping a Client's Files Together: Collocation" on page 258.	sequential
Reclamation threshold [1]	Specifies what percentage of reclaimable space can accumulate on a volume before the server initiates a space reclamation process for the volume. See "Choosing a Reclamation Threshold" on page 269.	sequential
Reclamation processes [1]	Specifies the number of concurrent processes to use for reclaiming the volumes in a storage pool. See "Specifying the Number of Reclamation Processes" on page 270.	sequential

Table 13. Information for Defining a Storage Pool (continued)

Information	Explanation	Type of Storage Pool
Offsite reclaim limit	Specifies the number of offsite volumes to have their space reclaimed during reclamation for a storage pool. See "Reclamation of Offsite Volumes" on page 274.	sequential
Reclamation storage pool [1]	Specifies the name of the storage pool to be used for storing data from volumes being reclaimed in this storage pool. Use for storage pools whose device class only has one drive or mount point. See "Reclaiming Volumes in a Storage Pool with One Drive" on page 271.	sequential
Reuse delay period	Specifies the number of days that must elapse after all of the files have been deleted from a volume, before the volume can be rewritten or returned to the scratch pool. See "Delaying Reuse of Volumes for Recovery Purposes" on page 651.	sequential
Overflow location [1]	Specifies the name of a location where volumes are stored when they are ejected from an automated library by the MOVE MEDIA command. Use for a storage pool that is associated with an automated library or an external library. See "Managing a Full Library" on page 180.	sequential
Data Format	The format in which data will be stored. NATIVE is the default data format. NETAPPDUMP and NONBLOCK are examples of other data formats.	sequential
Copy Storage Pools [1]	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 "Using the Simultaneous Write Function to Protect Your Data" on page 249 for related information. **Note:** 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]	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.	sequential

[1] This information is not available for sequential-access storage pools that use the following data formats:

- CELERRADUMP
- NDMPDUMP
- NETAPPDUMP

Example: Defining Storage Pools

For this example, suppose you have determined that an engineering department requires a separate storage hierarchy. You want the department's backed-up files to go to a disk storage pool. When that pool fills, you want the files to migrate to a tape storage pool. You want the 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 5 MB. Files larger than 5 MB are stored in the tape storage pool.
- Files migrate from the disk storage pool to the tape storage pool when the disk storage pool is 85% full. File migration to the tape storage pool stops when the disk storage pool is down to 40% full.
- The access mode is the default, read/write.
- Cache is used.

• Tape primary storage pool
  - The name of the pool is BACKTAPE.
  - The pool uses the device class TAPE, which has already been defined.
  - No limit is set for the maximum file size, because this is the last storage pool in the hierarchy.
  - To group files from the same client on a small number of volumes, use collocation at the client node level.
  - Use scratch volumes for this pool, with a maximum number of 100 volumes.
  - The access mode is the default, read/write.
  - Use the default for reclamation: Reclaim a partially full volume (to allow tape reuse) when 60% of the volume's space can be reclaimed.

You can define the storage pools in a storage pool hierarchy from the top down or from the bottom up. Defining the hierarchy from the bottom up requires fewer steps. To define the hierarchy from the bottom up, perform the following steps:

1. Define the storage pool named BACKTAPE with the following command:

```
define stgpool backtape tape
description='tape storage pool for engineering backups'
maxsize=nolimit collocate=node maxscratch=100
```

2. Define the storage pool named ENGBACK1 with the following command:

```
define stgpool engback1 disk
description='disk storage pool for engineering backups'
maxsize=5m nextstgpool=backtape highmig=85 lowmig=40
```

Restrictions:

1. 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.
2. The storage pool hierarchy includes only primary storage pools, not copy storage pools.
3. If a storage pool uses the data format NETAPPDUMP, CELERRADUMP, or NDMPDUMP, the server will not perform any of the following functions:
   - Storage pool backup
   - Migration
   - Reclamation
   - Moving files from one storage pool to other storage pool volumes
   - Volume audits
   - Data validation
   - Simultaneous write

   For more information about data formats, see Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143.

Example: Updating Storage Pools

You can update storage pools to change the storage hierarchy and other characteristics.

For example, suppose you had already defined the ENGBACK1 disk storage pool according to the previous example. Now you have decided to increase the maximum size of a physical file that may be stored in the storage pool. Use the following command:

```
update stgpool engback1 maxsize=100m
```

Note:

- 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

Table 14 gives tips on how to accomplish some tasks that are related to storage pools.

Table 14. 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 a Client's Files Together: Collocation" on page 258
Reduce the number of volume mounts needed to back up multiple clients.	Disable collocation for the storage pool.	"Keeping a Client's Files Together: Collocation" on page 258
Perform a simultaneous write to a primary storage pool and copy storage pools.	Provide a list of copy storage pools when defining the primary storage pool.	"Using the Simultaneous Write Function to Protect Your Data" on page 249
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 267
Move data from disk to tape automatically as needed.	Set a migration threshold for the storage pool. Identify the next storage pool.	"Migration for Disk Storage Pools" on page 239
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.	"Migration for Disk Storage Pools" on page 239

Table 14. Task Tips for Storage Pools (continued)

For this Goal	Do This	For More Information
Back up your storage pools.	Define a copy storage pool. Set up a backup schedule.	"Defining a Copy Storage Pool" on page 301 "Automating a Basic Administrative Command Schedule" on page 471
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 or Updating Primary Storage Pools" on page 220 "Changing Policy" on page 360
Make the best use of available tape drives and FILE volumes during reclamation and migration.	Specify multiple concurrent processes.	"Specifying the Number of Reclamation Processes" on page 270 "Specifying Multiple Concurrent Migration Processes" on page 247
Ensure that reclamation completes within the desired amount of time.	Limit the number of offsite volumes to be reclaimed.	"Reclamation of Offsite Volumes" on page 274 "Starting Reclamation Manually or in a Schedule" on page 270
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 228 "Defining Storage Pool Volumes" on page 229
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 228 "Defining Storage Pool Volumes" on page 229

Overview: Volumes in Storage Pools

Storage pool volumes are the physical media that are assigned to a storage pool. Some examples of volumes are:

- Space allocated on a disk drive
- A tape cartridge

Storage pools and their volumes are either random access or sequential access, depending on the device type of the device class to which the pool is assigned.

Random Access Storage Pool Volumes

Random access storage pools consist of volumes on disk. Random access storage pools are always associated with the DISK device class. All volumes are fixed-size files on a disk. You can create these volumes using the DEFINE VOLUME command, or you can use space triggers to automatically create and assign volumes to specified storage pools. See "Preparing Volumes for Random Access Storage Pools" on page 228 for details.

For important disk-related information, see "Requirements for Disk Subsystems, File Systems and Remote File Access" on page 83.

Sequential Access Storage Pool Volumes

Volumes in sequential access storage pools include any supported device type to which the server writes data sequentially. Some examples of sequential access volumes are:

- Tape cartridge
- File

Each volume defined in a sequential access storage pool must be of the same type as the device type of the associated device class. See Table 15 for the type of volumes associated with each device type.

For preparing sequential access volumes, see "Preparing Volumes for Sequential Access Storage Pools" on page 229.

Table 15. Volume Types

Device Type	Volume Description	Label Required
3590	IBM 3590 tape cartridge	Yes
3592	IBM 3592 tape cartridge	Yes
4MM	4-mm tape cartridge	Yes
8MM	8-mm tape cartridge	Yes
DLT	A digital linear tape	Yes
ECARTRIDGE	A cartridge tape that is used by a tape drive such as the StorageTek SD-3 or 9490 tape drive	Yes
FILE	A file in the file system of the server machine	No
LTO	IBM Ultrium tape cartridge	Yes
NAS	A tape drive that is used for backups via NDMP by a network-attached storage (NAS) file server	Yes
QIC	A 1/4-inch tape cartridge	Yes
REMOVABLEFILE	A file on a removable medium. If the medium has two sides, each side is a separate volume.	Yes
SERVER	One or more objects that are archived in the server storage of another server	No
VOLSAFE	A StorageTek cartridge tape that is for write-once use on tape drives that are enabled for VolSafe function.	No

Scratch Volumes Versus Defined Volumes

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 need over individual volumes.

Use defined volumes when you want to control precisely which volumes are used in the storage pool. Using defined volumes may be useful when you want to establish a naming scheme for volumes, or when you want to reduce potential disk fragmentation and maintenance overhead for storage pools associated with random-access and sequential-access disk.

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. 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.

Scratch volumes in a copy storage pool are handled in the same way as scratch volumes in a primary storage pool, except for volumes with the access value of offsite. If an offsite 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 onsite location. The administrator can query the server for empty offsite copy storage pool volumes and return them to the onsite location. The volume is returned to the scratch pool only when the access value is changed to READWRITE, READONLY, or UNAVAILABLE.

Preparing Volumes for Random Access Storage Pools

For a random access storage pool, you must define volumes.

Task	Required Privilege Class
Define volumes in any storage pool	System or unrestricted storage
Define volumes in specific storage pools	System, unrestricted storage, or restricted storage for those pools

Prepare a volume for use in a random access storage pool by defining 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 the path */usr/lpp/adsmserv/bin* and named stgvol.001. Enter the following command:

```
define volume backuppool /usr/lpp/adsmserv/bin/stgvol.001 formatsize=21
```

If you do not specify a full path name for the volume name, the command uses the path associated with the registry key of this server instance.

You can also define volumes in a single step using the DEFINE VOLUME command. For example, to define ten, 5000 MB volumes in a random-access storage pool that uses a DISK device class, you would enter the following command.

```
define volume diskpool diskvol numberofvolumes=10 formatsize=5000
```

Notes:

1. 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.
2. This one-step process replaces the former two-step process of first formatting a volume (using DSMFMT) and then defining the volume. If you choose to use the two-step process, the DSMFMT utility is available from the operating system command line. See the *Administrator's Reference* for details.

You can also use a space trigger to automatically create volumes assigned to a particular storage pool.

Preparing Volumes for Sequential Access Storage Pools

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 with other than a FILE or SERVER device type, you must prepare volumes for use. When the server accesses a sequential access volume, it checks the volume name in the header to ensure that the correct volume is being accessed. To prepare a volume:

1. Label the volume. Table 15 on page 227 shows the types of volumes that require labels. You must label those types of volumes before the server can use them.

   See "Labeling Removable Media Volumes" on page 168.

   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. See "Checking New Volumes into a Library" on page 171.
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 "Using Scratch Volumes" on page 230 for information about scratch volumes.

Defining Storage Pool Volumes

Task	Required Privilege Class
Define volumes in any storage pool	System or unrestricted storage
Define volumes in specific storage pools	System, unrestricted storage, or restricted storage for those pools

When you define a storage pool volume, you inform the server that the volume is available for storing backup, archive, or space-managed data.

For a sequential access storage pool, the server can use dynamically acquired scratch volumes, volumes that you define, or a combination.

To define a volume named VOL1 in the ENGBACK3 tape storage pool, enter:

```
define volume engback3 vol1
```

For storage pools associated with FILE device classes, you can define private volumes in a single step using the DEFINE VOLUME command. For example, to define ten, 5000 MB volumes, in a sequential-access storage pool that uses a FILE device class, you would enter the following command.

```
define volume filepool filevol numberofvolumes=10 formatsize=5000
```

For storage pools associated with the FILE device class, you can also use the DEFINE SPACETRIGGER and UPDATE SPACETRIGGER commands to have the server create volumes and assign them to a specified storage pool when predetermined space-utilization thresholds have been exceeded. One volume must be pre-defined.

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.

Using Scratch Volumes

You do not need to define volumes in sequential access storage pools if you allow storage pools to use scratch volumes. Use the MAXSCRATCH parameter when you define or update the storage pool. Setting the MAXSCRATCH parameter to a value greater than zero lets the storage pool dynamically acquire volumes as needed. 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. See "Preparing Volumes for Sequential Access Storage Pools" on page 229.

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.

Updating Storage Pool Volumes

Task	Required Privilege Class
Update volumes	System or operator

You can update the attributes of a storage pool volume assigned to a primary or copy storage pool. Update a volume to:

- Reset any error state for a volume, by updating the volume to an access mode of read/write.
- Change the access mode of a volume, for example if a tape cartridge is moved offsite (offsite access mode) or damaged (destroyed access mode). See "Access Modes for Storage Pool Volumes" on page 231.
- Change the location for a volume in a sequential access storage pool.

An example of when to use the UPDATE VOLUME command is if you accidentally damage a volume. You can change the access mode to unavailable so that the server does not try to write or read data from the volume. For example, if the volume name is VOL1, enter the following command:

```
update volume vol1 access=unavailable
```

When using the UPDATE VOLUME command, be prepared to supply some or all of the information shown in Table 16 on page 231.

Table 16. Information for Updating a Storage Pool Volume

Information	Explanation
Volume name *(Required)*	Specifies the name of the storage pool volume to be updated. You can specify a group of volumes to update by using wildcard characters in the volume name. You can also specify a group of volumes by specifying the storage pool, device class, current access mode, or status of the volumes you want to update. See the parameters that follow.
New access mode	Specifies the new access mode for the volume (how users and server processes such as migration can access files in the storage pool volume). See "Access Modes for Storage Pool Volumes" for descriptions of access modes. A random access volume must be varied offline before you can change its access mode to *unavailable* or *destroyed*. To vary a volume offline, use the VARY command. See "Varying Disk Volumes Online or Offline" on page 89. If a scratch volume that is empty and has an access mode of offsite is updated so that the access mode is read/write, read-only, or unavailable, the volume is deleted from the database.
Location	Specifies the location of the volume. This parameter can be specified only for volumes in sequential access storage pools.
Storage pool	Restricts the update to volumes in the specified storage pool.
Device class	Restricts the update to volumes in the specified device class.
Current access mode	Restricts the update to volumes that currently have the specified access mode.
Status	Restricts the update to volumes with the specified status (online, offline, empty, pending, filling, or full).
Preview	Specifies whether you want to preview the update operation without actually performing the update.

Access Modes for Storage Pool Volumes

Access to any volume in a storage pool is determined by the access mode assigned to that volume. You can change the access mode of a volume. The server can also 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 access modes are:

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 offsite 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 offsite 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 89.

If a scratch volume that is empty and has an access mode of offsite 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 "How Restore Processing Works" on page 640 for more information.

Only volumes in primary storage pools can be updated to an access mode of destroyed.

You must vary offline a random access volume before you can change its access mode to *destroyed*. To vary a volume offline, use the VARY command. See "Varying Disk Volumes Online or Offline" on page 89. 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 is at an offsite location and therefore cannot be mounted. Use this mode to help you track volumes that are offsite. The server treats offsite volumes differently, as follows:

- Mount requests are not generated for offsite volumes
- Data can be reclaimed or moved from offsite volumes by retrieving files from other storage pools
- Empty, offsite scratch volumes are not deleted from the copy storage pool

You can only update volumes in a copy storage pool to offsite 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 offsite.

Overview: The Storage Pool Hierarchy

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. To do this, you set up different primary storage pools to form a storage pool hierarchy. A typical implementation

may have a disk storage pool with a subordinate tape storage pool. When a client backs up a file, the server may initially store the file on disk according to the policy for that file. Later, the server may 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 can 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 "Scale Capacity" on page 202.

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. See "Keeping a Client's Files Together: Collocation" on page 258 for details.

Setting Up a Storage Pool Hierarchy

You can set up a storage pool hierarchy when you first define storage pools. You can also change the storage pool hierarchy later.

You establish a hierarchy by identifying 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. See "Migration of Files in a Storage Pool Hierarchy" on page 238 for detailed information on how migration between storage pools works.

Restrictions:

1. 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.
2. The storage pool hierarchy includes only primary storage pools, not copy storage pools. See "Using Copy Storage Pools to Back Up a Storage Hierarchy" on page 237.
3. A storage pool must use the NATIVE or NONBLOCK data formats to be part of a storage pool hierarchy. For example, a storage pool using the NETAPPDUMP data format cannot be part of a storage pool hierarchy.

Example: Defining a Storage Pool Hierarchy

For this example, suppose that you have determined that an engineering department requires a separate storage hierarchy. You set up policy so that the server initially stores backed up files for this department to a disk storage pool. When that pool fills, you want the server to migrate files to a tape storage pool. You want the 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, perform the following steps:

1. Define the storage pool named BACKTAPE with the following command:

```
define stgpool backtape tape
description='tape storage pool for engineering backups'
maxsize=nolimit collocate=node maxscratch=100
```

2. Define the storage pool named ENGBACK1 with the following command:

```
define stgpool engback1 disk
description='disk storage pool for engineering backups'
maxsize=5M nextstgpool=backtape highmig=85 lowmig=40
```

Example: Updating a Storage Pool Hierarchy

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.

For example, suppose that you had already defined the ENGBACK1 disk storage pool. Now you have decided to set up a tape storage pool to which files from ENGBACK1 can migrate. Perform the following steps to define the new tape storage pool and update the hierarchy:

1. Define the storage pool named BACKTAPE with the following command:

```
define stgpool backtape tape
description='tape storage pool for engineering backups'
maxsize=nolimit collocate=node maxscratch=100
```

2. Specify that BACKTAPE is the next storage pool defined in the storage hierarchy for ENGBACK1. To update ENGBACK1, enter:

```
update stgpool engback1 nextstgpool=backtape
```

How the Server Groups Files before Storing

When a user backs up or archives files from a client node, the server may group multiple client files into an *aggregate* of files. The size of the aggregate depends on the sizes of the client files being stored, and the number of bytes and files allowed for a single transaction. Two options affect the number of files and bytes allowed for a single transaction. TXNGROUPMAX, located in the server options file, affects the number of files allowed. TXNBYTELIMIT, located in the client options file, affects the number of bytes allowed in the aggregate.

- The TXNGROUPMAX option in the server options file indicates the maximum number of logical files (client files) that a client may send to the server in a single transaction. The server might create multiple aggregates for a single transaction, depending on how large the transaction is.

It is possible to affect the performance of client backup, archive, restore, and retrieve operations by using a larger value for this option. When transferring multiple small files, increasing the TXNGROUPMAX option can improve throughput for operations to tape.

Note: If you increase the value of the TXNGROUPMAX option by a large amount, watch for possible effects on the recovery log. A larger value for the TXNGROUPMAX option can result in increased utilization of the recovery log, as well as an increased length of time for a transaction to commit. If the effects are severe enough, they can lead to problems with operation of the server. For more information, see "Performance Considerations: Transferring Files as a Group between Client and Server" on page 490.

You can override the value of the TXNGROUPMAX server option for individual client nodes by using the TXNGROUPMAX parameter in the REGISTER NODE and UPDATE NODE commands.

- The TXNBYTELIMIT option in the client options file indicates the total number of bytes that the client can send to the server in a single transaction.

When a Tivoli Storage Manager for Space Management client (HSM client) migrates files to the server, the files are not grouped into an aggregate.

Where the Files Are Stored

When a user backs up, archives, or migrates a file from a client node, the server looks at the management class that is bound to the file. The management class specifies the destination, the storage pool in which to store the file. The server then checks that storage pool to determine the following:

- 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.

How the Server Stores Files in a Storage Hierarchy

As an example of how the server stores files in a storage hierarchy, assume a company has a storage pool hierarchy as shown in Figure 19.

Figure 19. 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 19.

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.

Using Copy Storage Pools to Back Up a Storage Hierarchy

Copy storage pools enable you to back up your primary storage pools for an additional level of data protection for clients. For details, see "Backing Up Storage Pools" on page 648 and "Using the Simultaneous Write Function to Protect Your Data" on page 249. Copy storage pools are not part of a storage hierarchy.

For efficiency, it is recommended that you use one copy storage pool to back up all primary storage pools that are linked to form a storage hierarchy. By backing up all primary storage pools to one copy storage pool, you do not need to recopy 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 can be used for backup of all primary storage pools. The number of copy storage pools you need depends on whether you have more than one primary storage pool hierarchy and on what type of disaster recovery protection you want to implement.

Multiple copy storage pools may be needed to handle particular situations, including:

- 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 onsite and one copy offsite)
- Rotation of full storage pool backups (see "Backing Up Storage Pools" on page 648 for more information)

Using the Hierarchy to Stage Client Data from Disk to Tape

A common way to use the storage hierarchy is to initially store client data on disk, then let the server migrate the data to tape. Typically you would need to ensure that you have enough disk storage to handle 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, if you have enough disk space for nightly incremental backups for clients and have tape devices, 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
- A copy storage pool on tape

You can then schedule the following steps every night:

1. Perform an incremental backup of the clients to the disk storage pool.
2. After clients complete their backups, back up the disk primary storage pool (now containing the incremental backups) to the copy storage 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.

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. Back up the tape primary storage pool to the copy storage pool to ensure that all files have been backed up.

   The tape primary storage pool must still be backed up 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).

See "Estimating Space Needs for Storage Pools" on page 277 for more information about storage pool space.

Migration of Files in a Storage Pool Hierarchy

The server provides automatic migration to maintain free space in a primary storage pool. The server can migrate data from one storage pool to the next storage pool in the hierarchy. This process helps to ensure that there is sufficient free space in the storage pools at the top of the hierarchy, where faster devices can provide the most benefit to clients. For example, the server can migrate data stored in a random access disk storage pool to a slower but less expensive sequential access storage pool.

You can control:

When migration begins and ends
: You use migration thresholds to control when migration begins and ends. Thresholds are set as levels of the space that is used in a storage pool, expressed as a percent of total space available in the storage pool. For a disk storage pool, the server compares the threshold with a calculation of the amount of data stored in the pool as a percent of the actual data capacity of the volumes in the pool. For a sequential access storage pool, the server compares the threshold with a calculation of the number of volumes containing data as a percent of the total number of volumes available to the pool.

  You can also schedule migration activities to occur when they are most convenient to you. In addition, you can specify how long migration will run before being automatically cancelled, whether the server attempts reclamation before migration, and whether the migration process runs in the background or foreground.

How the server chooses files to migrate
: By default, the server does not consider how long a file has been in a storage pool or how long since a file was accessed before choosing files to migrate. Optional parameters allow you to change the default. You can ensure that files remain in a storage pool for a minimum amount of time before the server migrates them to another pool. To do this, you set a migration delay period for a storage pool. Before the server can migrate a file, the file must be stored in the storage pool at least as long as the migration delay period. For disk storage pools, the last time the file was accessed is also considered for migration delay.

The number of concurrent migration processes
: You can specify a single migration process or multiple concurrent migration processes for a random-access or sequential-access storage pool.

Multiple concurrent processes let you make better use of your available tape drives and FILE volumes. However, because you can perform migration concurrently on different storage pools during auto-migration, you must carefully consider the resources (for example, drives) you have available for the operation.

Migration processing can differ for disk storage pools versus sequential access storage pools. If you plan to modify the default migration parameter settings for storage pools or want to understand how migration works, you should read the following sections:

"Migration for Disk Storage Pools"

"Migration for Sequential Access Storage Pools" on page 244

"Starting Migration Manually or in a Schedule" on page 246

Note: If you receive an error message during the migration process, refer to *IBM Tivoli Storage Manager Messages*. which can provide useful information for diagnosing and fixing problems. For example, suppose you receive the following message:

```
08/19/03 06:53:10  ANR1168W Migration skipping damaged file on volume
                   HLB311: Node node Louie, Type incremental,
                   File space C_drive, fsId 2, File name acct_data
```

IBM Tivoli Storage Manager Messages contains the following information about ANR1168W:

Explanation:
During migration, a file is encountered that was previously found to be damaged. If this file is part of an aggregate, the entire aggregate was previously marked damaged, possibly because an integrity error was detected for some other file within the aggregate.

System Action:
The damaged file will not be moved.

User Response:
Audit the indicated volume with FIX=NO to verify that the file is damaged. The audit will reset the file status if the file is found to be undamaged during the audit. If the file is part of an aggregate, the audit will reset the aggregate status if the entire aggregate is found to be undamaged. If this volume is in a primary storage pool that has previously been backed up to a copy storage pool, you can also attempt to restore damaged files by using the RESTORE STGPOOL command.

Migration for Disk Storage Pools

When you define or update a storage pool, you can set migration thresholds to specify when the server should begin and stop migrating data to the next storage pool in the storage hierarchy. Migration thresholds are defined in terms of a percentage of total data capacity for the disk storage pool. 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. See "How the Server Selects Files to Migrate" on page 240 and "Choosing Appropriate Migration Threshold Values" on page 242 for more information about migration thresholds.

You can control how long files must stay in a storage pool before they are eligible for migration by setting a migration delay for a storage pool. See "Keeping Files in a Storage Pool" on page 243.

If you decide to enable cache for disk storage pools, files can temporarily remain on disks even after migration. You may want to set migration thresholds lower when you use cache. See "Minimizing Access Time to Migrated Files" on page 244 and "Using Cache on Disk Storage Pools" on page 248 for information about using the cache.

How the Server Selects Files to Migrate

When data in a storage pool uses a percentage of the pool's capacity that is equal to the high migration threshold, the server migrates files from the pool to the next storage pool. The server selects the files to migrate as follows:

1. The server checks for the client node that has backed up or migrated the largest single file space or has archived files that occupy the most space.
2. For *all* files from *every* file space belonging to the client node that was identified, the server examines the number of days since the files were stored in the storage pool and last retrieved from the storage pool. The server compares the number (whichever is less) to the migration delay that is set for the storage pool. The server migrates any of these files for which the number is more than the migration delay set for the storage pool.
3. After the server migrates the files for the first client node to the next storage pool, the server checks the low migration threshold for the storage pool. If the amount of space that is used in the storage pool is now below the low migration threshold, migration ends. If not, the server chooses another client node by using the same criteria as described above, and the migration process continues.

The server may not be able to reach the low migration threshold for the pool by migrating only files that have been stored longer than the migration delay period. When this happens, the server checks the storage pool characteristic that determines whether migration should stop even if the pool is still above the low migration threshold. See "Keeping Files in a Storage Pool" on page 243 for more information.

If multiple migration processes are running (controlled by the MIGPROCESS parameter of the DEFINE STGPOOL command), the server may choose the files from more than one node for migration at the same time.

For example, Table 17 displays information that is contained in the database that is used by the server to determine which files to migrate. This example assumes that the storage pool contains no space-managed files. This example also assumes that the migration delay period for the storage pool is set to zero, meaning any files can be migrated regardless of time stored in the pool or the last time of access.

Table 17. Database Information on Files Stored in DISKPOOL

Client Node	Backed-Up File Spaces and Sizes		Archived Files (All Client File Spaces)
TOMC	TOMC/C	200 MB	55 MB
	TOMC/D	100 MB	
CAROL	CAROL	50 MB	5 MB

Table 17. Database Information on Files Stored in DISKPOOL (continued)

Client Node	Backed-Up File Spaces and Sizes		Archived Files (All Client File Spaces)
PEASE	PEASE/home	150 MB	40 MB
	PEASE/temp	175 MB	

Figure 20. The Migration Process and Migration Thresholds

Figure 20 shows what happens when the high migration threshold defined for the disk storage pool DISKPOOL is exceeded. When the amount of migratable data in DISKPOOL reaches 80%, the server performs the following tasks:

1. Determines that the TOMC/C file space is taking up the most space in the DISKPOOL storage pool, more than any other single backed-up or space-managed file space and more than any client node's archived files.
2. Locates all data belonging to node TOMC stored in DISKPOOL. In this example, node TOMC has backed up or archived files from file spaces TOMC/C and TOMC/D stored in the DISKPOOL storage pool.
3. Migrates all data from TOMC/C and TOMC/D to the next available storage pool. In this example, the data is migrated to the tape storage pool, TAPEPOOL.

   The server migrates all of the data from both file spaces belonging to node TOMC, even if the occupancy of the storage pool drops below the low migration threshold before the second file space has been migrated.

   If the cache option is enabled, files that are migrated remain on disk storage (that is, the files are *cached*) until space is needed for new files. For more information about using cache, see "Using Cache on Disk Storage Pools" on page 248.
4. After all files that belong to TOMC are migrated to the next storage pool, the server checks the low migration threshold. If the low migration threshold has not been reached, then the server again determines which client node has backed up or migrated the largest single file space or has archived files that occupy the most space. The server begins migrating files belonging to that node.

In this example, the server migrates *all* files that belong to the client node named PEASE to the TAPEPOOL storage pool.

5. After all the files that belong to PEASE are migrated to the next storage pool, the server checks the low migration threshold again. If the low migration threshold has been reached or passed, then migration ends.

Choosing Appropriate Migration Threshold Values

Setting migration thresholds for disk storage pools ensures sufficient free space on faster speed devices, which can lead to better performance. Choosing thresholds appropriate for your situation takes some experimenting, and you can start by using the default 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.

Choosing the High-Migration Threshold: To choose the high-migration threshold, consider:

- The amount of storage capacity provided for each storage pool
- The amount of free storage needed for users 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.

Choosing the Low-Migration Threshold: To choose the low-migration threshold, consider:

- The amount of free disk storage space needed for normal daily processing. If you have disk space to spare, you can keep more data on the disk (a larger low threshold). If clients' daily backups are enough to fill the disk space every day, you may need to empty the disk (a smaller low threshold).

  If your disk space is limited, try setting the threshold so that migration frees enough space for the pool to handle the amount of client data that is typically stored every day. Migration then runs about every day, or you can force it to run every day by lowering the high-migration threshold at a time you choose.

  You may also want to identify clients that are transferring large amounts of data daily. For these clients, you may want to set up policy (a new copy group or a new policy domain) so that their data is stored directly to tape. Using a separate policy in this way can optimize the use of disk for the majority of clients.

- Whether you use cache on disk storage pools to improve how quickly some files are retrieved. If you use cache, you can set the low threshold lower, yet still maintain faster retrieval for some data. Migrated data remains cached on the disk until new client data pushes the data off the disk. Using cache requires more disk space for the database, however, and can slow backup and archive operations that use the storage pool.

  If you do not use cache, you may want to keep the low threshold at a higher number so that more data stays on the disk.
- How frequently you want migration to occur, based on the availability of sequential access storage devices and mount operators. The larger the low threshold, the shorter time that a migration process runs (because there is less data to migrate). But if the pool refills quickly, then migration occurs more frequently. The smaller the low threshold, the longer time that a migration process runs, but the process runs less frequently.

  You may need to balance the costs of larger disk storage pools with the costs of running migration (drives, tapes, and either operators or automated libraries).
- Whether data in the next storage pool is being collocated by group. During migration from a disk storage pool, all the data for all nodes belonging to the same collocation group are migrated by the same process. Migration will continue regardless whether the low migration threshold has been reached or the amount of data that the group has to migrate.

Keeping Files in a Storage Pool

For some applications, you may want to ensure that files remain in the storage pool where they were initially stored by the server for a certain period of time. For example, you may have backups of monthly summary data that you want to keep in your disk storage pool for quicker access until the data is 30 days old. After the 30 days, the server can then move the file off into a tape storage pool.

You can delay migration of files for a specified number of days. The number of days is counted from the day that a file was stored in the storage pool or retrieved 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 whether the file has been in the storage pool for at least the migration delay period before migrating 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. See the *Administrator's Reference* for more information on the server option.

If you set migration delay for a pool, you need to decide what is more important: either ensuring that files stay in the storage pool for 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 file then must be obtained from the storage pool to which it was migrated.

Note: The use of cache has some disadvantages. See "Using Cache on Disk Storage Pools" on page 248.

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 once the pool becomes full, client files are stored directly to tape instead of to disk. Performance may be affected as a result.

Migration for Sequential Access Storage Pools

You can set up migration thresholds for sequential access storage pools. However, you probably will not want the server to perform this type of migration 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. Migrating data from one sequential access storage pool to another may be appropriate in some cases, for example, when you install a tape drive that uses a different type of tape and want to move data to that tape.

You 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 Files from One Volume to Another Volume" on page 294.

To control the migration process, you can 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 utilize your available tape drives or FILE volumes. (For details, see "Specifying Multiple Concurrent Migration Processes" on page 247.) 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 246.

Note: Storage pools using the NETAPPDUMP, the CELERRADUMP, or the NDMPDUMP data format are unable to use migration.

How IBM Tivoli Storage Manager Migrates Data from Sequential Access Storage Pools

The server begins the migration process when the number of volumes containing data as a percentage of the total volumes in the storage pool reaches the high migration threshold. The server migrates data from sequential storage pools by volume to minimize the number of mounts for volumes. The server performs the following processing for migration:

1. The server first reclaims volumes that have exceeded the reclamation threshold. Reclamation is a server process of consolidating data from several volumes onto one volume. (See "Reclaiming Space in Sequential Access Storage Pools" on page 267.)
2. After reclamation processing, the server compares the space used in the storage pool to the low migration threshold.
3. If the space used is now below the low migration threshold, the server stops processing. If the space used is still above the low migration threshold, the server determines which volume is the least recently referenced volume.
4. If the number of days since data was written is greater than the migration delay, the server migrates the volume. Otherwise, the server does not migrate this volume.
5. The server repeats steps 3 and 4 until the storage pool reaches the low migration threshold.

Because migration delay can prevent volumes from being migrated, the server can migrate data from all eligible volumes yet still find that the storage pool is above the low migration threshold. If you set migration delay for a pool, you need to decide what is more important: either ensuring that data stays in the storage pool for as long as the migration delay, or ensuring there is enough space in the storage pool for new data. 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 migrating only volumes that meet the migration delay requirement, you can choose one of the following:

- Allow the server to migrate volumes from the storage pool even if they do not meet the migration delay criteria (MIGCONTINUE=YES). This is the default. Allowing migration to continue ensures that space is made available in the storage pool for new files that need to be stored there.
- Have the server stop migration without reaching the low migration threshold (MIGCONTINUE=NO). Stopping migration ensures that volumes are not migrated for the time you specified with the migration delay. The administrator must ensure that there is always enough space available in the storage pool to hold the data for the required number of days.

Selecting Migration Criteria for Sequential Access Storage Pools

When defining migration criteria for sequential access storage pools, consider:

- The capacity of the volumes in the storage pool
- The time required to migrate data to the next storage pool
- The speed of the devices that the storage pool uses
- The time required to mount media, such as tape volumes, into drives
- Whether operator presence is required
- The number of concurrent migration processes

If you decide to migrate data from one sequential access storage pool to another, ensure that:

- Two drives (mount points) are available, one in each storage pool.
- The access mode for the next storage pool in the storage hierarchy is set to read/write.

  For information about setting an access mode for sequential access storage pools, see "Defining or Updating Primary Storage Pools" on page 220.
- Collocation is set the same in both storage pools. For example, if collocation is set to NODE in the first storage pool, then collocation should be set to NODE in the next storage pool.

  When you enable collocation for a storage pool, the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of volumes. For information about collocation for sequential access storage pools, see "Keeping a Client's Files Together: Collocation" on page 258.
- You have sufficient resources (for example, staff) available to handle 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 247.) More mount operations occur because the server attempts to reclaim space from sequential access storage pool volumes before it migrates files to the next storage pool.

  If you want to limit migration from a sequential access storage pool to another storage pool, set the high-migration threshold to a high percentage, such as 95%.

  For information about setting a reclamation threshold for tape storage pools, see "Reclaiming Space in Sequential Access Storage Pools" on page 267.

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 for a Client Node" on page 298.

Starting Migration Manually or in a Schedule

To gain more control over how and when the migration process occurs, you can use the MIGRATE STGPOOL command. Issuing this command starts migration from one storage pool to the next storage pool in the hierarchy, regardless of the value of the HIGHMIG parameter of the storage pool definition. You can specify the maximum number of minutes the migration will run before automatically cancelling. If you prefer, you can include this command in a schedule to perform migration when it is least intrusive to normal production needs.

For example, to migrate data from a storage pool named ALTPOOL to the next storage pool, and specify that it end as soon as possible after one hour, you would 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 command, refer to the *Administrator's Reference*.

Specifying Multiple Concurrent Migration Processes

You can specify one or more migration processes for each primary sequential access storage pool using the MIGPROCESS parameter on the DEFINE STGPOOL and UPDATE STGPOOL commands. Multiple migration processes run concurrently, allowing you to make better use of your available tape drives or FILE volumes.

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, you must 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 handle 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 "Mount Limit" on page 200.

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 for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT time, the migration processes will terminate.

The Tivoli Storage Manager server will start the specified number of migration processes regardless of the number of volumes that are eligible for migration. For example, if you specify ten migration processes and only six volumes are eligible for migration, the server will start ten processes and four of them will complete without processing a volume.

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.

Migration and Copy Storage Pools

Copy storage pools are not part of the hierarchy for migration. Files are not migrated to or from copy storage pools. The only way to store files in copy storage pools is by backing up primary storage pools (the BACKUP STGPOOL command).

Migration of files between primary storage pools does not affect copy storage pool files. Copy storage pool files do not 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.

Using Cache on Disk Storage Pools

When defining or updating disk storage pools, you can enable or disable cache.

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.

You can enable cache by specifying CACHE=YES when you define or update a storage pool. When cache is enabled, the migration process leaves behind duplicate copies of files on disk after the server migrates these files to the next storage pool in the storage hierarchy. The copies remain in the disk storage pool, but in a *cached* state, so that subsequent retrieval requests can be satisfied quickly. However, 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.

The advantage of using cache for a disk storage pool is that cache can improve how quickly the server retrieves some files. When you use cache, a copy of the file remains on disk storage after the server migrates the primary file to another storage pool. You may want to consider using a disk storage pool with cache enabled for storing space-managed files that are frequently accessed by clients.

However, 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 leave cache disabled, you may want to consider higher migration thresholds for the disk storage pool. A higher migration threshold keeps files on disk longer because migration occurs less frequently.

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 and occupy the largest amount of disk space 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 05/16/99 and File B was last retrieved on 06/19/99, then File A is deleted to reclaim space first.

You can change whether the server tracks the retrieval date for a file with the server option, NORETRIEVEDATE. When you include this option in the server options file, the server does not update the retrieval date for files. As a result, the server may remove copies of files in cache even though clients retrieved the files recently.

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 uses the statistic on migratable data (Pct Migr) to compare with migration threshold parameters to determine when migration should begin or end. For more information on storage pool statistics, see "Monitoring Storage Pools and Volumes" on page 279.

Using the Simultaneous Write Function to Protect Your Data

The simultaneous write function lets you write data simultaneously to a primary storage pool and up to three copy storage pools, increasing your level of data protection and reducing the time required for storage pool backup.

Simultaneous write is supported for the following operations:

- Backup and archive operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API
- Migration operations by Hierarchical Space Management (HSM) clients
- Import operations that involve copying exported file data from external media to a primary storage pool associated which is configured for simultaneous write configuration

The following sections describe the simultaneous write function in detail:

- "Function Overview"
- "How Simultaneous Write Works" on page 251
- "Operational Considerations" on page 255
- "Example: Simultaneous Write as Part of a Backup Strategy" on page 257

Function Overview

Tivoli Storage Manager has improved the resource management for simultaneous write. The parameters used to control this function are the COPYSTGPOOLS parameter and the COPYCONTINUE parameter that you can specify in a storage pool definition or update using the DEFINE STGPOOL or UPDATE STGPOOL commands. For details about these commands, refer to the *Administrator's Reference*.

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 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 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 store operation when the server has to switch primary storage pools:

- If the destination primary storage pool has one or more copy storage pools defined using the COPYSTGPOOL parameter, the server will write the data to the next storage pool and to the copy storage pools that are defined to the destination primary pool, regardless if the next pool has copy pools defined. The setting of the COPYCONTINUE of the destination primary storage pool will be inherited by the next primary storage pool. If the next pool has copy pools defined, they will be ignored as well as the value of its COPYCONTINUE attribute.
- If no copy storage pools are defined in the destination primary storage pool, the server will write the data to the next primary storage pool. If the next pool has copy storage pools defined, they will be ignored.

These rules apply to all the primary storage pools within the storage pool hierarchy.

If a write failure occurs for any of the copy storage pools, the setting of the COPYCONTINUE parameter determines how the server will react.

- If the COPYCONTINUE parameter is set to YES, the server will stop writing to the failing copy pools for the remaining of the session, but continue storing files into the primary pool and any remaining copy pools. The copy storage pool list is active only for the life of the session and applies to all the primary storage pools in a particular storage pool hierarchy.
- If the COPYCONTINUE parameter is set to NO, the server will fail the current transaction and discontinue the store operation.

Notes:

1. Simultaneous write is not supported for data movements performed by the server, such as server migration, reclamation, moving data from one storage pool to another storage pool, or backing up a storage pool.
2. Simultaneous write takes precedence over LAN-free operations. The operations go over the LAN, and the simultaneous write configuration is honored.
3. Create current backup and archive versions of the files before the HSM client migrates them. If you back up or archive a copy of an HSM migrated file to the same Tivoli Storage Manager server to which it was migrated, the file will only be stored into the primary storage pool.
4. The COPYSTGPOOLS parameter is available only to primary storage pools that use NATIVE or NONBLOCK data format. This parameter is not available for storage pools that use the following data formats:
   - NETAPPDUMP
   - CELERRADUMP
   - NDMPDUMP
5. When a NAS backup operation is writing a TOC file. If the primary storage pool specified in the TOCDESTINATION in the copy group of the management class has copy storage pools defined, the copy storage pools are ignored and the data is stored into the primary storage pool only.
6. Target storage pools used for simultaneous write operations can have different device classes. Performance is limited by the speed of the slowest device.

Attention: Use of the simultaneous write function is not intended to replace regular backup of storage pools. If you use the function to simultaneously write to copy storage pools, ensure that the copy of each primary storage pool is complete by regularly issuing the BACKUP STGPOOL command. If you fail to perform regular storage pool backups, you could 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 will remove 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 will continue to write to the primary storage pool and to any remaining copy storage 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.

How Simultaneous Write Works

This section provides three examples that show how simultaneous write works. These are examples only.

For these examples, assume the following:

- 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.
- Files A, B and C are bound to management class STANDARD, and the destination specified in its backup copy group is DISKPOOL. For detailed information about creating policies, see Chapter 13, “Implementing Policies for Client Data,” on page 357.
- DISKPOOL has enough space to store only files C and B, and its next pool (TAPEPOOL) has enough space for file A.

Example One

Suppose that you want to use the simultaneous write function to automatically copy client data to two copy storage pools, COPYPOOL1 and COPYPOOL2 during a backup operation. If a write failure occurs to either or both of the copy storage pools, you want the server to stop writing to the failing copy pools for the remainder of the session, but continue to store files into the primary pool and any remaining copy pools. With DISKPOOL and TAPEPOOL already defined as your storage pool hierarchy, use commands the following commands to enable simultaneous write:

```
define stgpool copypool1 mytapedevice pooltype=copy
define stgpool copypool2 mytapedevice pooltype=copy
update stgpool diskpool copystgpools=copypool1,copypool2 copycontinue=yes
```

where MYTAPEDEVICE is the device-class name associated with the copy storage pools.

Figure 21 on page 252 shows the storage pool hierarchy and the copy storage pools associated with DISKPOOL.

Figure 21. 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. When files A, B and C are backed up, the following events occur. (See Figure 22.)

- C and B are written to DISKPOOL, COPYPOOL1 and COPYPOOL2.
- File A is written to TAPEPOOL, COPYPOOL1 and COPYPOOL2.

Figure 22. Inheriting a List of Copy Storage Pools

As a precaution, you should issue the BACKUP STGPOOL command after the backup operation has completed.

Example Two

This example describes a situation in which simultaneous write is not used by the next storage pools in a hierarchy. Suppose you do not specify a list of copy storage pools for DISKPOOL. However, you do specify copy storage pools for TAPEPOOL (COPYPOOL1 and COPYPOOL2). You also specify a value of YES for the COPYCONTINUE parameter:

```
define stgpool copypool1 mytapedevice pooltype=copy
define stgpool copypool2 mytapedevice pooltype=copy
update stgpool tapepool copystgpools=copypool1,copypool2
    copycontinue=yes
```

where MYTAPEDEVICE is the device-class name associated with the copy storage pools. Figure 23 shows this configuration:

Figure 23. Example of Storage Pool Hierarchy with Copy Storage Pools Defined for TAPEPOOL

When files A, B and C are backed up, the following events occur. (See Figure 24 on page 254.)

- C and B are written to DISKPOOL.
- File A is written to TAPEPOOL.

Figure 24. Inheriting an Empty Copy Storage Pool List

Although TAPEPOOL has copy storage pools defined, file A is not copied because TAPEPOOL inherits an empty copy storage pool list from DISKPOOL.

Example Three

The following example shows what happens if an error occurs during a simultaneous write operation. Suppose you specify COPYPOOL1 and COPYPOOL2 as copy storage pools for DISKPOOL and you set the value of the COPYCONTINUE parameter to YES. This configuration is identical to that in the first example. See Figure 21 on page 252.

When files A, B and C are backed up, the following events occur. (See Figure 25 on page 255.)

- An error occurs while writing to COPYPOOL1, and it is removed from the copy storage pool list held in memory by the server. The transaction fails.
- Because the value of the COPYCONTINUE parameter is YES, the client retries the backup operation. 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 C and B are simultaneously written to DISKPOOL and COPYPOOL2.
- File A is simultaneously written to TAPEPOOL and COPYPOOL2.

Figure 25. Inheriting a List of Copy Storage Pools

Note that, 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, you should issue the BACKUP STGPOOL command for the copy storage pool that failed:

```
backup diskpool copystgpool1
backup tapepool copystgpool1
```

Operational Considerations

This section describes important issues that you should consider before implementing simultaneous write.

Setting the Node's MAXNUMMP Parameter

Careful consideration should be given to the number of mount points available for a simultaneous write operation. A client session requires a mount point in order 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.

Suppose, for example, you create a storage pool hierarchy like that shown in Figure 21 on page 252. DISKPOOL is a random access storage pool, and TAPEPOOL, COPYPOOL1 and COPYPOOL2 are sequential access storage pools. For each client backup session, the client might have to acquire three mount points if it has to write data to TAPEPOOL. To run two backup sessions concurrently, the client requires a total of six mount points.

To indicate the number of mount points a client can have, you specify a value for the MAXNUMMP parameter on the REGISTER NODE or UPDATE NODE commands. Be sure to check the value of the MAXNUMMP parameter and, if necessary, update it if you want to enable simultaneous write. A value of 2 for the MAXNUMMP parameter might be sufficient if, during a client session, all the data is stored in DISKPOOL, COPYPOOL1 and COPYPOOL2.

If the number of mount points required for a client session exceeds the value of the client's MAXNUMMP parameter, the transaction fails. The setting of the COPYCONTINUE parameter determines whether the transaction is retried:

- If the value of the COPYCONTINUE parameter is NO, the client does not retry the operation.
- 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's session. The client retries the operation.

Setting the Device Class MOUNTLIMIT Parameter

The value of the MOUNTLIMIT parameter in the device class definition specifies the maximum number of sequential access volumes that can be simultaneously mounted.

If the number of sequential access volumes that need to be mounted for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the server will not be able to acquire the mount points and the operation will fail. The value of the COPYCONTINUE parameter determines whether the client retries the operation:

- If the value of the COPYCONTINUE parameter is NO, the client does not retry the operation.
- If the value of the COPYCONTINUE parameter is YES, the Tivoli Storage Manager server attempts to remove the copy storage pools that use this device class until enough mount points can be acquired. The transaction fails, and the client retries the operation. If sufficient mount points can be acquired when the operation is retried, the data is written into the primary storage pool and any remaining copy storage pools.

Storing Data Without Using Simultaneous Write

Using simultaneous write to copy storage pools may not be an efficient solution for every primary storage pool. For example, suppose you use a DISK primary storage pool that is accessed by a large number of clients at the same time during client data-storage operations. If this storage pool is associated with copy storage pools, the clients might have to wait until enough tape drives are available to perform the store operation. In this scenario, simultaneous write could extend the amount of time required for client data-storage operations. It might be more efficient, then, to store the data in the primary storage pool and use the BACKUP STGPOOL command to backup the DISK storage pool to the copy storage pools.

Reducing the Potential for Switching Storage Pools

Switching primary storage pools can delay the completion of a simultaneous write operation. 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 current destination primary pool cannot contain the data being stored, the Tivoli Storage Manager server will attempt to store the data into a next storage pool in the storage hierarchy. This next storage pool normally uses a sequential-access device class. If new resources have to be acquired for the next storage pool, or the allocated resources have to be released because the server has to wait to acquire the new resources, the client session will have to 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 HSM migrations.
- 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 will cause the server to switch pools. To prevent switching pools, avoid using this parameter if possible.

Compartmentalize Your Data

When considering simultaneous write as an element of your backup strategy, you should, as a general rule, try to compartmentalize your data in discrete storage pool hierarchies. For example, as shown in Figure 26, you could configure your production servers to store mission critical data in one storage pool hierarchy and use simultaneous write to backup the data to copy storage pools. In addition, you could configure the servers to store non-critical, workstation data in another storage pool hierarchy and back up that data using the BACKUP STGPOOL command.

Figure 26. Separate Storage Pool Hierarchies for Different Types of Data

Example: Simultaneous Write as Part of a Backup Strategy

The following example shows how simultaneous write can be used to create onsite backups of a storage pool for easy availability and how the BACKUP STGPOOL command can be used to create storage pool backups and database backups that can be moved offsite to provide data protection in case a disaster occurs. This example is provided for illustrative purposes only. When designing a backup strategy, you should carefully consider your own system, data storage and disaster recovery requirements.

1. Define two copy storage pools, ONSITECOPYPOOL and DRCOPYPOOL, and two primary storage pools, DISKPOOL and TAPEPOOL. As part of the storage pool definition for DISKPOOL, specify TAPEPOOL as the next storage pool and ONSITECOPYPOOL as the copy storage pool. Set the copy continue parameter to YES so that if an error occurs writing to the copy pool, the operation will continue storing data into the primary pool.

   ```
   define stgpool tapepool mytapedevice
   define stgpool onnsitepool mytapedevice
   define stgpool drcopypoool mytapedevice
   define stgpool diskpool mydiskdevice nextstgpool=tapepool
       copystgpool=onsitecopypool copycontinue=yes
   ```

   This basic configuration is the same as that shown in Figure 21 on page 252.
2. Schedule or issue the following commands to ensure that all the files are backed up

   ```
   backup stgpool diskpool onsitecopypool
   backup stgpool tapepool onsitecopypool
   ```
3. To create the storage pool backup volumes that will be moved offsite, 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 offsite 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 will be available in DISKPOOL in preparation for the next storage operations:

   ```
   migrate stgpool diskpool
   ```

Keeping a Client's Files Together: Collocation

With collocation enabled, the server attempts to keep files belonging to a group of client nodes, a single client node, or client file space on a minimal number of sequential access storage volumes. You can set *collocation* for each sequential access storage pool when you define or update the pool.

By using collocation, you can reduce 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. Figure 27 on page 259 shows an example of collocation by client node with three clients, each having a separate volume containing that client's data.

Figure 27. Example of Collocation Enabled

Figure 28 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 28. Example of Collocation Enabled

When collocation is disabled, the server attempts to use all available space on each volume before selecting a new volume. While this process provides better utilization of individual volumes, user files can become scattered across many volumes. Figure 29 on page 260 shows an example of collocation disabled, with three clients sharing space on single volume.

Figure 29. 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 is no collocation.

The following sections give more detail on collocation:

"The Effects of Collocation on Operations"
"How the Server Selects Volumes with Collocation Enabled" on page 261
"How the Server Selects Volumes with Collocation Disabled" on page 263
"Turning Collocation On or Off" on page 264
"Collocation on Copy Storage Pools" on page 264
"Planning for and Enabling Collocation" on page 265

The Effects of Collocation on Operations

Table 18 summarizes the effects of collocation on operations.

Table 18. Effect of Collocation on Operations

Operation	Collocation Enabled	Collocation Disabled
Backing up, archiving, or migrating client files	More media mounts to collocate files.	Usually fewer media mounts are required.
Restoring, retrieving or recalling client files	Large numbers of files can be restored, retrieved, or recalled more quickly because files are located on fewer volumes.	Multiple mounts of media may be required for a single user because files may be spread across multiple volumes. More than one user's files can be stored on the same sequential access storage volume. For example, if two users attempt to recover a file that resides on the same volume, the second user will be forced to wait until the first user's files are recovered.

Table 18. Effect of Collocation on Operations (continued)

Operation	Collocation Enabled	Collocation Disabled
Storing data on tape	The server attempts to use all available tape volumes to separate user files before it uses all available space on every tape volume.	The server attempts to use all available space on each tape volume before using another tape volume.
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 handle because volumes are not fully used.	More mount operations required during restore, retrieve, and recall of client files.

During the following server operations, all the data belonging to a collocation group, a single client node, or a single client file space is moved or copied by one process:

1. Moving data from random access and sequential access volumes
2. Moving node data from sequential access volumes
3. Backing up a random access or sequential access storage pool
4. Restoring a sequential access storage pool
5. Reclamation of a sequential access storage pool or offsite volumes
6. Migration from a random access storage pool.

For example, if data is collocated by group, all data for all nodes belonging to the same collocation group is migrated by the same process.

If collocation is by node or file space, nodes or file spaces are selected for migration based on the amount of data to be migrated. The node or file space with the most data is migrated first. If collocation is by group, all nodes in the storage pool are first evaluated to determine which node has the most data. The node with the most data is migrated first along with all the data for all the nodes belonging to that collocation group regardless of the amount of data in the nodes' file spaces or whether the low migration threshold has been reached.

One reason to collocate by group is that individual client nodes often do not have sufficient data to fill high-capacity tape volumes. Collocating data by groups of nodes can reduce unused tape capacity by putting more collocated data on individual tapes. In addition, because all data belonging to all nodes in the same collocation group are migrated by the same process, collocation by group can reduce the number of times a volume containing data to be migrated needs to be mounted. Collocation by group can also minimize database scanning and reduce tape passes during data transfer from one sequential access storage pool to another.

How the Server Selects Volumes with Collocation Enabled

Table 19 on page 262 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 19. How the Server Selects Volumes When Collocation is Enabled

Volume Selection Order	When collocation is by group	When collocation is by node	When collocation is by file space
1	A volume that already contains files from the collocation group to which the client belongs	A volume that already contains files from the same client node	A volume that already contains files from the same file space of that client node
2	An empty predefined volume	An empty predefined volume	An empty predefined volume
3	An empty scratch volume	An empty scratch volume	An empty scratch volume
4	A volume with the most available free space among volumes that already contain data	A volume with the most available free space among volumes that already contain data	A volume containing data from the same client node
5	Not applicable	Not applicable	A volume with the most available free space among volumes that already contain data

When the server needs to continue to store data on a second volume, it uses the following selection order to acquire additional space:

1. An empty predefined volume
2. An empty scratch volume
3. A volume with the most available free space among volumes that already contain data
4. Any available volume in the storage pool

When collocation is by client node or file space, the server attempts to provide the best use of individual volumes while minimizing the mixing of files from different clients or file spaces on volumes. This is depicted in Figure 30 on page 263, 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.

Notes:

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 30. Using All Available Sequential Access Storage Volumes with Collocation Enabled at the Group or File Space Level

When collocation is by group, the server attempts to collocate data from nodes belonging to the same collocation group. As shown in the Figure 31, data for the following groups of nodes has been collocated:

- Group 1 consists of nodes A, B, and C
- Group 2 consists of nodes D and E
- Group 3 consists of nodes F, G, H, and I

Whenever possible, the Tivoli Storage Manager server collocates data belonging to a group of nodes on a single tape, as represented by Group 2 in the figure. Data for a single node can also be spread across several tapes associated with a group (Group 1 and 2). If the nodes in the collocation group have multiple file spaces, the server does not attempt to collocate those file spaces.

Figure 31. Using All Available Sequential Access Storage Volumes with Collocation Enabled at the Group Level

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 32 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 32. Using All Available Space on Sequential Volumes with Collocation Disabled

Turning Collocation On or Off

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.

Note: 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 on Copy Storage Pools

Using collocation on copy storage pools requires special consideration.

Primary and copy storage pools perform different recovery roles. Normally you use primary storage pools to recover data to clients directly. You use copy storage pools to recover data to the primary storage pools. In a disaster where both clients and the server are lost, the copy storage pool volumes will probably be used directly to recover clients. The types of recovery scenarios that concern you the most will help you to determine whether to use collocation on your copy storage pools.

You may also want to consider that collocation on copy storage pools will result in more partially filled volumes and potentially unnecessary offsite reclamation activity. Collocation typically results in partially filled volumes when you collocate by node or by file space. (Partially filled volumes are less prevalent, however, when you collocate by group.) Partially filled volumes might be acceptable for primary storage pools because the volumes remain available and can be filled

during the next migration process. However, this may be unacceptable for copy storage pools because the storage pool backups are usually taken offsite immediately. If you use collocation for copy storage pools, you will have to decide among the following:

- Taking more partially filled volumes offsite, thereby increasing the reclamation activity when the reclamation threshold is lowered or reached.
- 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 in order to utilize as much tape capacity as possible

With collocation disabled for a copy storage pool, typically there will be only a few partially filled volumes after storage pool backups to the copy storage pool are complete.

Consider carefully before using collocation for copy storage pools. Even if you use collocation for your primary storage pools, you may want to disable collocation for copy storage pools. Collocation on copy storage pools may be desirable when you have few clients, but each of them has large amounts of incremental backup data each day.

For more information about collocation, see "Keeping a Client's Files Together: Collocation" on page 258.

Planning for and Enabling Collocation

Tivoli Storage Manager provides the following collocation options for the DEFINE STGPOOL and UPDATE STGPOOL commands:

No Specifies that collocation is disabled.

Group Specifies that collocation is enabled at the group level for client nodes. The server attempts to put data for nodes that belong to the same collocation group on as few volumes as possible.

Node Specifies that collocation is enabled at the client node level. The server attempts to put data for one node on as few volumes as possible.

Filespace
Specifies that collocation is enabled at the file space level for client nodes. The server attempts to put data for one node and file space on as few volumes as possible.

When deciding whether and how to collocate data, do the following:

1. Familiarize yourself with the potential advantages and disadvantages of collocation, in general. For a summary of effects of collocation on operations, see Table 18 on page 260.
2. If the decision is to collocate, determine how data should be organized, whether by client node, group of client nodes, or file space. If the decision is to collocate by group, you need to decide how to group nodes:
   - If the goal is to save space, you may wish to group small nodes together to better utilize 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 multisession No Query Restore operation.

- If the goal is to departmentalize data, then you can group nodes by department.

3. If collocation by group is the desired result, you need to do the following:
   a. Define collocation groups using the DEFINE COLLOCGROUP command.
   b. Add client nodes to the collocation groups using the DEFINE COLLOCGROUPMEMBER command.

   The following query commands are available to help in collocating groups:

   QUERY COLLOCGROUP
   Displays the collocation groups defined on the server.

   QUERY NODE
   Displays the collocation group, if any, to which a node belongs.

   QUERY NODEDATA
   Displays information about the data for one or more nodes in a sequential access storage pool.

   QUERY STGPOOL
   Displays information about the location of client data in a sequential access storage pool and the amount of space a node occupies in a volume.

   For more information about these commands, refer to the *Administrator's Reference*.

   You can also use Tivoli Storage Manager server scripts or PERL scripts to display information that can be useful in defining collocation groups.
4. Specify how data is to be collocated in a storage pool using the COLLOCATE parameter on the DEFINE STGPOOL or UPDATE STGPOOL command.
5. If you decide later that you want to delete members of a collocation group, you can use the DELETE COLLOCMEMBER command. You can also update the description of a collocation group using the UPDATE COLLOCGROUP command and delete entire collocation groups using the DELETE COLLOCGROUP command.

Tip: If you use collocation, but want to reduce the number of media mounts and use space on sequential volumes more efficiently, you can do the following:

- Define a storage pool hierarchy and policy to require that backed-up, archived, or space-managed files are stored initially in disk storage pools.

  When files are migrated from a disk storage pool, the server attempts to migrate all files belonging to the client node or collocation group that is using the most disk space in the storage pool. This process works well with the collocation option because the server tries to place all of the files from a given client on the same sequential access storage volume.
- Use scratch volumes for sequential access storage pools to allow the server to select new volumes for collocation.
- Specify the client option COLLOCATEBYFILESPEC to limit the number of tapes to which objects associated with one file specification are written. This collocation option makes collocation by the server more efficient; it does not override collocation by file space or collocation by node. For general information about client options, see "Managing Client Option Files" on page 340. For details about the COLLOCATEBYFILESPEC option, refer to the *Backup-Archive Clients Installation and User's Guide*.

Reclaiming Space in Sequential Access Storage Pools

Space on a sequential volume becomes reclaimable as files expire or are deleted from the volume. For example, files become obsolete because of aging or limits on the number of versions of a file. 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. See the following sections:

"How IBM Tivoli Storage Manager Reclamation Works" on page 267

"Choosing a Reclamation Threshold" on page 269

"Lowering the Reclamation Threshold" on page 270

"Starting Reclamation Manually or in a Schedule" on page 270

"Specifying the Number of Reclamation Processes" on page 270

"Reclaiming Volumes in a Storage Pool with One Drive" on page 271

"Reclamation of Tape Volumes with High Capacity" on page 272

"Reclamation of Volumes with the Device Type of SERVER" on page 272

"Reclamation for Copy Storage Pools" on page 273

"How Collocation Affects Reclamation" on page 276

Note: Storage pools defined with the NETAPPDUMP, the CELERRADUMP or the NDMPDUMP data format cannot be reclaimed.

How IBM Tivoli Storage Manager Reclamation Works

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. You can set a reclamation threshold for each sequential access storage pool when you define or update the pool.

During space reclamation, the server copies files that remain on eligible volumes to other volumes. For example, Figure 33 on page 268 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.

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 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 or are deleted by the client. The server removes unused space from expired or deleted logical files 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 During Data Movement" on page 297 for details.

Figure 33. 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 "Reclamation of Volumes with the Device Type of SERVER" on page 272 for details of how the server reclaims these types of volumes.

Volumes in a copy storage pool are reclaimed in the same manner as a primary storage pool except for the following:

- *Offsite* volumes are handled differently.
- The server copies active files from the candidate volume only to other volumes in the *same* storage pool.

See "Reclamation for Copy Storage Pools" on page 273 for details.

Note: If you receive an error message during the reclamation process, refer to *IBM Tivoli Storage Manager Messages*. which can provide useful information for diagnosing and fixing problems. For example, suppose you receive the following message:

```
08/19/03 06:53:10  ANR1175W Volume CNN292 contains files which
                   could not be reclaimed.
```

IBM Tivoli Storage Manager Messages contains the following information about ANR1175W:

Explanation:
After reclamation of the indicated volume, it still contains one or more files that could not be reclaimed. These files were skipped because of read errors, or because the files are marked damaged.

System Action:
The volume is marked unavailable so it will not be chosen again for reclamation.

User Response:

- Use the UPDATE VOLUME command to set the volume access to readonly.
- Try using the MOVE DATA command to manually reclaim the volume.
- If files still exist on the volume, audit the volume using AUDIT VOLUME FIX=YES.
- Try using the MOVE DATA command again.
- Use the RESTORE VOLUME command to restore files marked damaged.

Choosing a Reclamation Threshold

The reclamation threshold indicates how much reclaimable space a volume must have before the server reclaims the volume. 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 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.

Lowering the Reclamation Threshold

If you have been running with a high reclamation threshold and decide you now need to reclaim volumes, you may want to lower the threshold in several steps. Lowering the threshold in steps ensures that volumes with the most reclaimable space are reclaimed first. For example, if you had 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 onsite volume is being reclaimed, the server uses the new threshold setting when the process begins to reclaim the next volume. If offsite volumes are being reclaimed, the server does not use the new threshold setting during the process that is running (because all eligible offsite volumes are reclaimed at the same time).

Use the CANCEL PROCESS command to stop a reclamation process.

Starting Reclamation Manually or in a Schedule

To gain more control over how and when the reclamation process occurs, you can use the RECLAIM STGPOOL command. You can also specify the maximum amount of time a reclamation process will take before it is automatically cancelled. 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, 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*.

Specifying the Number of Reclamation Processes

You can specify one or more reclamation processes for each primary sequential access storage pool or copy storage pool using the RECLAIMPROCESS parameter on the DEFINE STGPOOL and UPDATE STGPOOL commands. Multiple reclamation processes run concurrently, allowing you to make better use of your available tape drives or FILE volumes.

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 handle 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 "Mount Limit" on page 200.

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 for mount points or drives to become available. The amount of time the processes will wait is the value specified for the MOUNTWAIT parameter on the DEFINE DEVCLASS commands, if no 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 276.

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 enable volume reclamation for a storage pool that has only one mount point, you can define a *reclamation storage pool* for the server to use when reclaiming volumes. 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
```

Notes:

1. You can specify multiple concurrent reclamation processes for a primary storage pool with one drive by using the RECLAIMSTGPOOL parameter. If multiple concurrent processing is not desired, specify a value of 1 for the RECLAIMPROCESS parameter on the DEFINE STGPOOL or UPDATE STGPOOL commands.
2. 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

Reclamation of 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 of a type that are relatively slow at positioning tapes. To help reduce overall process time, consider doing the following:

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 Volumes with the Device Type of SERVER

When virtual volumes (volumes with the device type of SERVER) in a primary storage pool are reclaimed, the client data stored on those volumes is sent across the network between the source server and the target server. As a result, the reclamation process can tie up your network resources. To control when reclamation starts for these volumes, consider setting the reclamation threshold to 100% for any primary storage pool that uses virtual volumes. Lower the reclamation threshold at a time when your network is less busy, so that the server can reclaim volumes.

For virtual volumes in a copy storage pool, the server reclaims a volume as follows:

1. The source server determines which files on the volume are still valid.
2. The source server obtains these valid files from a primary storage pool, or if necessary, from an onsite volume (not a virtual volume) in another copy storage pool.
3. The source server writes the files to one or more new virtual volumes in the copy storage pool and updates its database.
4. The server issues a message indicating that the volume was reclaimed.

Note: You can specify multiple concurrent reclamation processes for a primary storage pool with a device type of SERVER. However, running multiple concurrent processes for this type of storage pool can tie up network resources because the data is sent across the network between the source server and target server. Therefore, if you want to run multiple concurrent processes, do so when the network is less busy. If multiple concurrent processing is not desired, specify a value of 1 for the RECLAIMPROCESS parameter on the DEFINE STGPOOL or UPDATE STGPOOL commands.

For information about using the SERVER device type, see "Using Virtual Volumes to Store Data on Another Server" on page 601.

Reclamation for Copy Storage Pools

Reclamation of primary storage pool volumes does not affect copy storage pool files.

Reclamation of volumes in copy storage pools is similar to that of primary storage pools. However, most volumes in copy storage pools may be set to an access mode of offsite, making them ineligible to be mounted. When reclamation occurs and how reclamation processing is done depends on whether the volumes are marked as offsite.

For volumes that are not offsite, reclamation usually occurs after the volume is full and then begins to empty because of file deletion. 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 offsite volumes, reclamation can occur when the percentage of unused space on the volume is greater than the reclaim parameter value. The unused space includes both space that has never been used on the volume and space that has become empty because of file deletion. During reclamation, the server copies valid files on offsite volumes from the original files in the primary storage pools. In this way, the server copies valid files on offsite volumes without having to mount these volumes. For more information, see "Reclamation of Offsite Volumes" on page 274.

Reclamation of copy storage pool volumes should be done periodically to allow reuse of partially filled volumes that are offsite. Reclamation can be done automatically by setting the reclamation threshold for the copy storage pool to less than 100%. However, you need to consider controlling when reclamation occurs because of how offsite volumes are treated. For more information, see "Controlling When Reclamation Occurs for Offsite Volumes" on page 274.

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 offsite 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. Doing so will enable you to make better use of your available tape drives or FILE volumes. The principles underlying multiple concurrent reclamation processes for copy storage 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 "Specifying the Number of Reclamation Processes" on page 270.

Reclamation of Offsite Volumes

As for volumes with other access values, 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. The default reclamation threshold for copy storage 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 may 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 and 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.

   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 Backup Volumes Onsite" on page 702.

Controlling When Reclamation Occurs for Offsite Volumes

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 offsite 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 offsite. The reclamation process would cause a new volume to be created with the same files on it. The volume you take offsite 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 offsite.

One way to resolve this situation is to keep partially filled volumes onsite until they fill up. However, this would mean a small amount of your data would be without an offsite copy for another day.

If you send copy storage pool volumes offsite, it is recommended you control copy storage 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 offsite 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 offsite volumes each day. You may choose to perform offsite reclamation on a less frequent basis. For example, suppose you ship copy storage pool volumes to and from your offsite storage location once a week. You can run reclamation for the copy storage pool weekly, so that as offsite volumes become empty they are sent back for reuse.

When you do perform reclamation for offsite volumes, the following sequence is recommended:

1. Back up your primary storage pools to copy storage pools.
2. Turn on reclamation for copy storage pools by lowering the reclamation threshold below 100%.
3. When reclamation processing completes, turn off reclamation for copy storage pools by raising the reclamation threshold to 100%.
4. Mark any newly created copy storage pool volumes as offsite and then move them to the offsite location.

This sequence ensures that the files on the new copy storage pool volumes are sent offsite, and are not inadvertently kept onsite because of reclamation.

Using Storage on Another Server for Copy Storage Pools: Another resolution to this problem of partially filled volumes is to use storage on another Tivoli Storage Manager server (device type of SERVER) for storage pool backups. If the other server is at a different site, the copy storage pool volumes are already offsite, with no moving of physical volumes between the sites. See "Using Virtual Volumes to Store Data on Another Server" on page 601 for more information.

Limiting the Number of Offsite 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 offsite volumes to be reclaimed. When determining the value for the OFFSITERECLAIMLIMIT, consider using the statistical information in the message issued at the end of the offsite volume reclamation operation.

Alternatively, you can use the following Tivoli Storage Manager SQL SELECT command to obtain the statistical information from the SUMMARY table for the offsite volume reclamation operation:

```
select * from summary where activity='OFFSITE RECLAMATION'
```

The statistical information includes the following:

- The number of offsite volumes that were processed

- The number of parallel processes that were used
- The total amount of time required for the processing

The order in which offsite volumes are reclaimed is based on the amount of unused space in a volume. (Unused space includes both space that has never been used on the volume and space that has become empty because of file deletion.) Volumes with the largest amount of unused space are reclaimed first.

For example, suppose a copy storage pool contains three volumes: VOL1, VOL2, and VOL3. VOL1 has the largest amount of unused space, and VOL3 has the least amount of unused space. Suppose further that the percentage of unused space in each of the three volumes is greater than the value of the RECLAIM parameter. If you do not specify a value for the OFFSITERECLAIMLIMIT parameter, all three volumes will be reclaimed when the reclamation runs. If you specify a value of 2, only VOL1 and VOL2 will be reclaimed when the reclamation runs. If you specify a value of 1, only VOL1 will be reclaimed.

Delaying Reuse of Reclaimed Volumes

You should delay the reuse of any reclaimed volumes in copy storage pools for as long as you keep your oldest database backup. Delaying reuse may help you to recover data under certain conditions during recovery from a disaster. For more information on delaying volume reuse, see "Delaying Reuse of Volumes for Recovery Purposes" on page 651.

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. Therefore, 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

Multiple concurrent reclamation processing does not affect collocation. 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 "Specifying the Number of Reclamation Processes" on page 270.

See also "Reclamation of Tape Volumes with High Capacity" on page 272.

Estimating Space Needs for Storage Pools

This section provides guidelines for estimating the initial storage space required for your installation. You have the following default random access (disk) storage pools available at installation:

- BACKUPPOOL for backed-up files
- ARCHIVEPOOL for archived files
- SPACEMGPOOL for files migrated from client nodes (space-managed files)

You can add space to these storage pools by adding volumes, or you can define additional storage pools.

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 "Requesting Information on the Use of Storage Space" on page 291.

Estimating Space Needs in Random Access Storage Pools

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 "Choosing Appropriate Migration Threshold Values" on page 242 for recommendations on setting migration thresholds.

Estimating Space for Backed-Up Files in a Random Access Storage Pool

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 a Random Access Storage Pool

Estimating the amount of storage space for archived files is more difficult, because the number of archived files generated by users is not necessarily related to the amount of data stored on their workstations.

To estimate the total amount of space needed for all archived files in a single random access (disk) storage pool, determine what percentage of user files are typically archived.

Work with policy administrators to calculate this percentage based on the number and type of archive copy groups defined. For example, if policy administrators have defined archive copy groups for only half of the policy domains in your enterprise, then estimate that you need less than 50% of the amount of space you have defined for backed-up files.

Because additional storage space can be added at any time, you can start with a modest amount of storage space and increase the space by adding storage volumes to the archive storage pool, as required.

Estimating Space Needs in Sequential Access Storage Pools

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 267 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 a Client's Files Together: Collocation" on page 258 for information about using collocation for your storage pools.
- The type of storage devices and sequential volumes supported at your installation

Monitoring Storage Pools and Volumes

Any administrator can query for information about a storage pool by viewing a standard or a detailed report. Use these reports to monitor storage pool usage, including:

- Whether you need to add space to your disk and sequential access storage pools
- The status of the process of migrating data from one to storage pool to the next storage pool in the storage hierarchy
- 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 may need to define more volumes for disk storage pools, or add more volumes for sequential access storage pools such as tape. For more information on maintaining a supply of volumes in libraries, see "Managing the Volume Inventory" on page 176.

Estimated Capacity and Percent Utilization

To obtain a standard report that lists the estimated capacity and utilization percentage of all storage pools defined to the system, you can issue the following command:

```
query stgpool
```

Figure 34 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 34. Information about Storage Pools

Estimated Capacity

Specifies the space available in the storage pool in megabytes (M) or gigabytes (G).

For a disk storage pool, this value reflects the total amount of available space in the storage pool, including any volumes that are varied offline.

For a sequential access storage pool, this value is an estimate of the total amount of available space on all volumes in the storage pool. The total includes volumes with any access mode (read-write, unavailable, read-only, offsite, or destroyed). The total includes scratch volumes that the storage pool can acquire only when the storage pool is using at least one scratch volume for data.

Volumes in a sequential access storage pool, unlike those in a disk storage pool, do not contain a precisely known amount of space. Data is written to a volume as necessary until the end of the volume is reached. For this reason, the estimated capacity is truly an *estimate* of the amount of available space in a sequential access storage pool.

Pct Util

Specifies, as a percentage, the space used in each storage pool.

For disk storage pools, this value reflects the total number of disk blocks currently allocated by Tivoli Storage Manager. Space is allocated for backed-up, archived, or space-managed files that are eligible for server migration, cached files that are copies of server-migrated files, and files that reside on any volumes that are varied offline.

Note: The value for Pct Util can be higher than the value for Pct Migr if you query for storage pool information while a client transaction (such as a backup) is in progress. The value for Pct Util is determined by the amount of space actually allocated (while the transaction is in progress). The value for Pct Migr represents only the space occupied by *committed* files. At the end of the transaction, Pct Util and Pct Migr become synchronized.

For sequential access storage pools, this value is the percentage of the total bytes of storage available that are currently being used to store active data (data that is not expired). Because the server can only estimate the available capacity of a sequential access storage pool, this percentage also reflects an estimate of the actual utilization of the storage pool.

Space Trigger Utilization and Scratch Volumes

To obtain a detailed report that lists not only the estimated capacity and utilization percentage of storage pools, but also the space-trigger and scratch-volume utilization, 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. 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.

Example: Monitoring the Capacity of a Backup Storage Pool

Figure 34 on page 279 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.

Monitoring the Use of Storage Pool Volumes

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, such as the following:
  - 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

Getting General Information about Storage Pool Volumes

To request general information about all volumes defined to the server, enter:

```
query volume
```

Figure 35 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
                                                      (MB)
------------------------ -----------  ----------  ---------  -----  --------
/dev/raixvol1            AIXPOOL1     DISK            240.0   26.3  On-Line
/dev/raixvol2            AIXPOOL2     DISK            240.0   36.9  On-Line
/dev/rdosvol1            DOSPOOL1     DISK            240.0   72.2  On-Line
/dev/rdosvol2            DOSPOOL2     DISK            240.0   74.1  On-Line
/dev/ros2vol1            OS2POOL1     DISK            240.0   55.7  On-Line
/dev/ros2vol2            OS2POOL2     DISK            240.0   51.0  On-Line
WREN00                   TAPEPOOL     TAPE8MM       2,472.0    0.0  Filling
WREN01                   TAPEPOOL     TAPE8MM       2,472.0    2.2  Filling
```

Figure 35. 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 36 shows the output of this detailed query. Table 20 gives some suggestions on how you can use the information.

```
                   Volume Name: WREN01
             Storage Pool Name: TAPEPOOL
             Device Class Name: TAPE8MM
       Estimated Capacity (MB): 2,472.0
                      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 36. Detailed Information for a Storage Pool Volume

Table 20. 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.

Table 20. Using the Detailed Report for a Volume (continued)

Task	Fields and Description
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 a Storage Pool Volume" on page 671 for information about auditing a volume.
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 Files from One Volume to Another Volume" on page 294. 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 a Storage Pool Volume" on page 671 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.

Table 20. Using the Detailed Report for a Volume (continued)

Task	Fields and Description
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, offsite volumes in copy storage 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 234 and "Requesting Information on the Use of Storage Space" on page 291.

Getting 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 do the following:

- Request the server to correct any inconsistencies (AUDIT VOLUME command)
- Move files from one volume to other volumes
- Delete a volume from a storage pool

Because the server tracks the contents of a storage volume through its database, the server does not need to access the requested volume to determine its contents.

The report generated by a QUERY CONTENT command shows the contents of a volume. 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.

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.

Viewing a Standard Report on the Contents of a Volume: 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 37 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 37. 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.

Viewing a Detailed Report on the Contents of a Volume: To display detailed information about the files stored on volume VOL1, enter:

```
query content vol1 format=detailed
```

Figure 38 on page 286 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 38. Viewing a Detailed Report of the Contents of a Volume

Finding the Volumes Used by a Client Node

You can use the server's SELECT command to find the sequential volumes used by a client node. Use SELECT to perform an SQL query of 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

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 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, including files on volumes that are varied offline. Cached data are excluded in the Pct Migr value.

  For sequential access storage pools, this value is the percentage of the total volumes in the storage pool that actually contain data at the moment. For example, assume a storage pool has four explicitly defined volumes, and a maximum scratch value of six volumes. If only two volumes actually contain data at the moment, then Pct Migr will be 20%.

  This field is blank for copy storage 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.)

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.)

Next Storage Pool
: Specifies the primary storage pool destination to which data is migrated. (This field is blank for copy storage pools.)

Example: Monitoring the Migration of Data Between Storage Pools

Figure 34 on page 279 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 39 on page 288 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 39. Information on Backup Storage Pools

You can query the server to monitor the migration process by entering:

```
query process
```

A message similar to Figure 40 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 40. Information on the Migration Process

When migration is finished, the server displays the following message:

```
ANR1101I Migration ended for storage pool BACKUPPOOL.
```

Handling Problems during the Migration Process

A problem can occur that causes the migration process to be suspended. For example, there may not be sufficient space in the storage pool to which data is being migrated. When migration is suspended, the process might be retried.

At this point, a system administrator can:

- Cancel the migration process. See “Canceling the Migration Process” for additional information.
- End the migration process by changing the attributes of the storage pool from which data is being migrated. See “Ending the Migration Process by Changing Storage Pool Characteristics” on page 289 for additional information.
- Provide additional space. See “Providing Additional Space for the Migration Process” on page 289 for additional information.

The server attempts to restart the migration process every 60 seconds for several minutes and if not successful will terminate the migration process.

Canceling the Migration Process

To stop server migration when a problem occurs or when you need the resources the process is using, you can cancel the migration.

First determine the identification number of the migration process by entering:

```
query process
```

A message similar to Figure 41 is displayed:

```
 Process Process Description      Status
  Number
-------- ------------------------ --------------------------------------------
       1 Migration                ANR1113W Migration suspended for storage pool
                                   BACKUPPOOL - insufficient space in
                                   subordinate storage pool.
```

Figure 41. Getting the Identification Number of the Migration Process

Then you can cancel the migration process by entering:

```
cancel process 1
```

Ending the Migration Process by Changing Storage Pool Characteristics

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 will end.) To stop the repeated attempts at restart, you can change some characteristics of the storage pool from which data is being migrated. Depending on your environment, you can:

- Set higher migration thresholds for the storage pool from which data is being migrated. The higher threshold means the storage pool must have more migratable data before migration starts. This change delays migration.

  In the example in "Example: Monitoring the Migration of Data Between Storage Pools" on page 287, 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 the Migration of Data Between Storage Pools" on page 287, you could add volumes to the disk storage pool BACKUPPOOL to increase its storage capacity.

  Note: Do this only if you received an out-of-space message for the storage pool to which data is being migrated.

Providing Additional Space for the Migration Process

A migration process can be suspended because of insufficient space in the storage pool to which data is being migrated. To allow the migration process to complete, you can provide additional storage volumes for that storage pool.

In the example in "Example: Monitoring the Migration of Data Between Storage Pools" on page 287, you could 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

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.

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. For example, to request a detailed report for BACKUPPOOL, enter:

```
query stgpool backuppool format=detailed
```

Figure 42 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):
         Continue Copy on Error?:
                        CRC Data: No
                Reclamation Type:
```

Figure 42. 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.

Requesting Information on the Use of Storage Space

Task	Required Privilege Class
Query the server for information about server storage	Any administrator

Any administrator can request information about server storage occupancy. Use the QUERY OCCUPANCY command for reports with information broken out by node or file space. Use this report to determine the amount of space used by:

- Client node and file space
- Storage pool or device class
- Type of data (backup, archive, or space-managed)

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.
- 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 may contain empty space that had been used by logical files that are now expired or deleted. 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.

Amount of Space Used by Client Node

Any administrator can request information about the space used by each client node and file space:

- How much data has been backed up, archived, or migrated to server storage
- How many of the files that are in server storage have been backed up to a copy storage pool
- The amount of storage space being used

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
```

Remember that file space names are case-sensitive and must be entered exactly as they are known to the server. Use the QUERY FILESPACE command to determine the correct capitalization. For more information, see “Managing File Spaces” on page 329.

Figure 43 on page 292 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 43. 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*.

Amount of Space Used by Storage Pool or Device Class

You can monitor the amount of space being used by an individual storage pool, a group of storage pools, or storage pools categorized by a particular device class. Creating occupancy reports on a regular basis can help you with capacity planning.

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 44 on page 293 displays a report on the occupancy of tape storage pools assigned to the TAPECLASS device class.

```
Node Name        Type  Filespace    Storage      Number of    Physical   Logical
                       Name         Pool Name        Files       Space     Space
                                                                Occupied  Occupied
                                                                    (MB)      (MB)
---------------  ----  -----------  -----------  ---------  ----------  --------
CAROL            Arch  OS2C         ARCHTAPE           5         .92        .89
CAROL            Bkup  OS2C         BACKTAPE          21        1.02       1.02
PEASE            Arch  /home/peas-  ARCHTAPE         492       18.40      18.40
                        e/dir
PEASE            Bkup  /home/peas-  BACKTAPE          33        7.60       7.38
                        e/dir
PEASE            Bkup  /home/peas-  BACKTAPE           2         .80        .80
                        e/dir1
TOMC             Arch  /home/tomc   ARCHTAPE         573      20.85      19.27
                        /driver5
TOMC             Bkup  /home        BACKTAPE          13        2.02       1.88
```

Figure 44. A Report on the Occupancy of Storage Pools by Device Class

Note: For archived data, you may 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.

Amount of Space Used by Backed-Up, Archived, or 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. See "Estimating Space Needs for Storage Pools" on page 277 and "Estimating Space for Archived Files in a Random Access Storage Pool" on page 278 for information about planning storage space.

To request a report about backup versions stored in the disk storage pool named BACKUPPOOL, for example, enter:

```
query occupancy stgpool=backuppool type=backup
```

Figure 45 displays a report on the amount of server storage used for backed-up files.

```
Node Name        Type  Filespace    Storage      Number of    Physical   Logical
                       Name         Pool Name        Files       Space     Space
                                                                Occupied  Occupied
                                                                    (MB)      (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 45. 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 45:

1. Add the number of megabytes of space occupied by backup versions.

   In this example, backup versions occupy 92.22 MB of space in BACKUPPOOL.
2. Add the number of files stored in the storage pool.

   In this example, 1760 backup versions reside in BACKUPPOOL.

3. Divide the space occupied by the number of files to determine the average size of each file backed up to the BACKUPPOOL.

   In this example, the average size of each workstation file backed up to BACKUPPOOL is about 0.05 MB, or approximately 50 KB.

You can use this average to estimate the capacity required for additional storage pools that are defined to the server.

Moving Files from One Volume to Another Volume

You can move files from one volume to another volume in the same or a different storage pool using the MOVE DATA command. The volumes can be onsite volumes or offsite volumes. During normal operations, you do not need to move data. You might need to move data in some situations, for example, when you need to salvage any readable data from a damaged Tivoli Storage Manager volume.

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.

Note:

- Files in a copy storage pool do not move when primary files are moved.
- You can only move data for volumes belonging to a storage pool with DATAFORMAT=NATIVE or DATAFORMAT=NONBLOCK.

Task	Required Privilege Class
Move files from a volume in any storage pool to an available volume in any storage pool	System or unrestricted storage
Move files from one volume to an available volume in any storage pool to which you are authorized	Restricted storage

Moving Data to Other Volumes in the Same Storage Pool

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 304 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 "Moving Data to Another Storage Pool."

Moving Data to Another 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.

Note: Data cannot be moved from a primary storage pool to a copy storage pool. Data in a copy storage pool cannot be moved to any other 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.

Moving Data from an Offsite Volume in a Copy Storage Pool

You can move data from offsite volumes without bringing the volumes onsite. Processing of the MOVE DATA command for primary storage pool volumes does not affect copy storage pool files.

Processing of the MOVE DATA command for volumes in copy storage pools is similar to that of primary storage pools, with the following exceptions:

- Most volumes in copy storage pools may 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.
- 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.

When you move files from a volume marked as offsite, the server does the following:

1. Determines which files are still active on the volume from which you are moving data
2. Obtains these 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

Procedure for Moving Data

1. Before you move files from a volume, complete the following steps:
   - 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 230 for information about updating the access mode of a storage pool volume.

- Ensure sufficient space is available on volumes within the specified destination storage pool by:
  a. 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 281 for information about requesting information about a storage volume.
  b. Querying the specified destination storage pool to ensure there is sufficient capacity to store the files being moved. See "Monitoring Space Available in a Storage Pool" on page 279 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 229 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.

  See "Requesting Information about a Device Class" on page 211 for requesting information about the mount limit value for the device class.
- If you are moving files from a tape volume to a tape storage pool, ensure that the two tape drives required are available.

2. Move the data using the MOVE DATA command.

   For example, to move the files stored in the /dev/vol3 volume to any available volume in the STGTMP1 storage pool, enter:

   ```
   move data /dev/vol3 stgpool=stgtmp1
   ```

   When you move data from a volume, the server starts a background process and sends informational messages, such as:

   ```
   ANR1140I Move Data process started for volume /dev/vol3
   (process ID 32).
   ```

   The command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

   Note:

   - A volume may not be totally empty after a move data operation completes. For example, the server may be unable to relocate one or more files to another volume because of input/output errors on the device or because errors were found in the file. You can delete the volume with DISCARDDATA=YES to delete the volume and any remaining files. The server then deletes the remaining files that had I/O or other errors.
   - You can only move data for volumes belonging to a storage pool with DATAFORMAT=NATIVE or DATAFORMAT=NONBLOCK.

Requesting Information about the Data Movement Process

To request information on the data movement process, enter:

```
query process
```

Figure 46 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 46. Information on the Data Movement Process

Reclaiming Space in Aggregates During Data Movement

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 only for specific volumes. To remove empty space in a specific volume and reconstruct an aggregate, you can issue the MOVE DATA command which will by default will remove the empty space occupied by deleted files in an aggregate.

Monitoring the Movement of Data between Volumes

You can query the server for volume information to monitor the movement of data between volumes. 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
                                                     (MB)
---------------     -----------   ----------     ---------    -----    --------
/dev/vol3           BACKUPPOOL    DISK                15.0     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
                                                     (MB)
----------------    -----------   ----------     ---------    -----    --------
VOL1                STGTMP1       8500DEV          4,944.0      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
                                                     (MB)
----------------    ----------    ----------     ---------    -----    --------
/dev/vol3           BACKUPPOOL    DISK                15.0      0.0    On-Line
```

Moving Data for a Client Node

You can move data located in a sequential-access storage pool for one or more nodes, or for a single node with selected file spaces, by using the MOVE NODEDATA command. For this command the data can be located on either a primary or copy storage pool. When the source storage pool is a primary storage pool, you can move data to other volumes within the same pool or to another primary storage pool. When the source storage pool is a copy storage pool, data can only be moved to other volumes within that storage pool.

Notes:

1. You can only move data by node if the data resides in a storage pool whose data format is NATIVE or NONBLOCK.
2. If you are moving files within the same storage pool, there must be volumes available that do not contain the data you are moving. That is, the server cannot use a destination volume containing data that will need to be moved.

Task	Required Privilege Class
Move data by node	System, unrestricted storage or restricted storage

Moving Data for All File Spaces for One or More Nodes

Moving data for all file spaces on one or more nodes 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

Note: You should 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 for Selected File Spaces for One Node

Moving data for selected file spaces for a single node 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
```

Requesting Information about the Data Movement Process

To request information on the data movement process, enter:

```
query process
```

Figure 47 on page 300 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 47. Information on the Data Movement Process

Preventing Incomplete Data Movement Operations

There are various reasons why an incomplete MOVE NODEDATA operation can occur. The following are the most common reasons:

- Files have been marked as damaged in the source storage pool. For more information regarding how to deal with files marked as damaged see "Correcting Damaged Files" on page 678.
- 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 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 while a MOVE NODEDATA is running for that copy pool
  - Storage of files from a client directly into the storage pool

Renaming a Storage Pool

You can rename a storage pool. You may need to do this when distributing policy using enterprise configuration. See "Setting Up a Managed Server" on page 578.

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 may 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 need to change the destination in the copy groups and management classes. You then activate the policy set with the changed destinations.

Defining a Copy Storage Pool

Use a copy storage pool to back up one or more primary storage pools. See Table 22 on page 302 and "Backing Up Storage Pools" on page 648 for more information. When you define a copy storage pool, be prepared to provide some or all of the information in Table 21.

Note: To back up a primary storage pool the DATAFORMAT must be NATIVE or NONBLOCK.

Table 21. Information for Defining a Copy Storage Pool

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. This is a required parameter. Updating a storage pool cannot change whether the pool is a primary or copy 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. Files can be moved within the volumes of the copy storage 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. See "Maintaining a Supply of Scratch Volumes in an Automated Library" on page 182.
Collocation	When collocation is enabled, the server attempts to keep all files belonging to a group of client nodes, a single client node, or a client file space on a minimal number of sequential access storage volumes. See "Collocation on Copy Storage Pools" on page 264.

Table 21. Information for Defining a Copy Storage Pool (continued)

Information	Explanation
Reclamation threshold	Specifies when to initiate reclamation of volumes in the copy storage 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 offsite storage pool volumes and virtual volumes. When a copy storage pool volume that is offsite becomes eligible for reclamation, the reclamation process attempts 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. See "Reclamation for Copy Storage Pools" on page 273 and "Reclamation of Volumes with the Device Type of SERVER" on page 272 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 "Delaying Reuse of Reclaimed Volumes" on page 276.
Offsite reclaim limit	Specifies the number of offsite volumes to be reclaimed during reclamation for a storage pool. See "Reclamation of Offsite Volumes" on page 274.
Reclamation processes	Specifies the number of concurrent processes to use for reclaiming the volumes in a storage pool. See "Reclamation for Copy Storage Pools" on page 273.

Example: Defining a Copy Storage Pool

Assume you need to maintain copies of the files stored in BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL (default disk storage pools) for disaster recovery purposes. You want to create a copy storage pool named DISASTER-RECOVERY. You decide to use only scratch tapes in the new pool, setting the maximum number of scratch volumes to an appropriate value. You enter the following command:

```
define stgpool disaster-recovery tapeclass pooltype=copy
maxscratch=100
```

To store data in the new storage pool, you must back up the primary storage pools (BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL) to the DISASTER-RECOVERY pool. See "Backing Up Storage Pools" on page 648.

Comparing Primary and Copy Storage Pools

Table 22 compares the characteristics of primary and copy storage pools.

Table 22. Comparing Primary and Copy Storage Pools

Characteristic	Primary storage pool	Copy storage pool
Destination for backed-up or archived files (specified in backup or archive copy groups)	Yes	No
Destination for space-managed files (specified in the management class)	Yes	No

Table 22. Comparing Primary and Copy Storage Pools (continued)

Characteristic	Primary storage pool	Copy storage pool
Offsite access mode for volumes	No	Yes, except for volumes with device type SERVER
Destroyed access mode for volumes	Yes	No
Random access storage volumes	Yes	No
Sequential access storage volumes	Yes	Yes
Contents	Client files (backup versions, archived files, space-managed files)	Copies of files that are stored in primary storage pools
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 offsite, data is copied from the original files in primary storage pools. • If volumes are offsite, you cannot move data in those volumes by node.
Collocation	Yes (sequential access storage pools only)	Yes
Reclamation	Yes (sequential access storage pools only)	Yes Virtual volumes (volumes with device type SERVER) and offsite volumes are handled differently. For details, see "Reclamation of Volumes with the Device Type of SERVER" on page 272 and "Reclamation of Offsite Volumes" on page 274.
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

Deleting a Storage Pool

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 a Storage Pool Volume with Data" on page 305 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 279.

  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 quotes) to remove this storage pool from the storage hierarchy definition

  See "Defining or Updating Primary Storage Pools" on page 220 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.

Deleting Storage Pool Volumes

You can delete volumes, and optionally the client files they contain, from either primary or copy storage pools.

If files that are not cached are deleted from a primary storage pool volume, any copies of these files in copy storage pools will also be deleted.

Files in a copy storage 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.
- The primary file is deleted because of:
  – Policy-based file expiration
  – File space deletion
  – Deletion of the primary storage pool volume

Tip: If you are deleting many volumes, delete the volumes one at a time. Concurrently deleting many volumes can adversely affect server performance.

Task	Required Privilege Class
Delete volumes from any storage pool	System or unrestricted storage
Delete volumes from storage pools over which they have authority	Restricted storage

Deleting an Empty Storage Pool Volume

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)
```

After you respond yes, the server generates a background process to delete the volume.

The command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

Deleting a Storage Pool Volume with Data

To prevent you from accidentally deleting 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.

Notes:

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.

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 Files from One Volume to Another Volume" on page 294 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

© Copyright IBM Corp. 1993, 2005

Chapter 11. Adding Client Nodes

When the IBM Tivoli Storage Manager server is installed, the IBM Tivoli Storage Manager backup-archive client and the administrative client are installed on the same machine as the server by default. However, many installations of IBM Tivoli Storage Manager include remote clients, and application clients on other machines, often running on different operating systems.

The server views its registered clients as nodes that require services and resources from the server. The term *nodes* in this chapter indicates the following type of clients and servers that you can register as client nodes:

- Tivoli Storage Manager backup-archive client
- IBM System Storage Archive Manager application 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 *Backup-Archive Clients Installation and User's Guide*. Administrators can perform the following activities when managing nodes:

Tasks:
Tasks:
"Installing Client Node Software" on page 310
"Accepting Default Closed Registration or Enabling Open Registration" on page 310
"Registering Nodes with the Server" on page 310
"Connecting Nodes with the Server" on page 313
Concepts:
"Overview of Clients and Servers as Nodes" on page 310
"Comparing Network-Attached Nodes to Local Nodes" on page 315

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

© Copyright IBM Corp. 1993, 2005

Overview of Clients and Servers as Nodes

Each backup-archive client, 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 machine 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 machine. For example, a Structured Query Language (SQL) server machine can contain both a Tivoli Storage Manager for SQL server application client for database and transaction log backups, and a Tivoli Storage Manager backup-archive client for file system backups.

Installing Client Node Software

Administrators can install backup-archive clients, application clients, or Tivoli Storage Manager for Space Management clients by using any of the following methods:

- Installing directly from the CD-ROM
- Installing by transferring installable files from the CD-ROM to a target machine
- Installing by creating client software images and installing the images

For more information about installing:

- Client software, refer to *Backup-Archive Clients Installation and User's Guide*.
- IBM System Storage Archive Manager application client software, refer to the application client documentation for your particular client.

Use the procedures in this chapter to configure a node after it has been installed.

Registering Nodes with the Server

Administrators can register Tivoli Storage Manager clients, application clients, and HSM clients as client nodes.

When a node is registered, Tivoli Storage Manager automatically creates an administrative user ID with client owner authority over the node. You can use this administrative user ID to access the Web backup-archive client from remote locations through a Web browser. If an administrative user ID already exists with the same name, an administrative user ID is not automatically defined. For more information, see "Overview of Remote Access to Web Backup-Archive Clients" on page 325.

Note: To connect to a Web backup-archive client directly from a supported Web browser or from a hyperlink in the Web administrative Enterprise Console, you must specify the node's URL and port number during the registration process or later update the node with this information.

Accepting Default Closed Registration or Enabling Open Registration

Before a user can request Tivoli Storage Manager services, the node must be registered with the server.

Closed registration is the default. The administrator must register client nodes when registration is set to closed.

Open registration allows the client nodes to register their node names, passwords, and compression options. On UNIX and Linux systems, only the root user can register a client node with the server.

With either registration mode, by default, an administrative user ID with client owner authority is created over the node.

Note: Changes to the registration process do not affect existing registered client nodes.

Closed Registration

To add a node with closed registration, an administrator uses the REGISTER NODE command to register the node and specify the initial password. The administrator can also specify the following optional parameters:

- Contact information.
- The name of the policy domain to which the node is assigned.
- Whether the node compresses its files before sending them to the server for backup and archive.
- Whether the node can delete backups and archives from server storage.
- The name of a client option set to be used by the node.
- Whether to force a node to change or reset the password.
- The type of node being registered.
- The URL address used to administer the client node.
- The maximum number of mount points the node can use.
- Whether the client node keeps a mount point for an entire session.
- The transfer path used when the node sends data.
- The transfer path used when data is read for a client.
- Whether the server or client node initiates sessions.
- The IP address of the node.
- The low level address of the node.

Open Registration

To add a node with open registration, the server prompts the user for a node name, password, and contact information the first time the user attempts to connect to the server. With open registration, the server automatically assigns the node to the STANDARD policy domain. The server by default allows users to delete archive copies, but not backups stored in server storage.

You can enable open registration by entering the following command from an administrative client command line:

```
set registration open
```

For examples and a list of open registration defaults, refer to the *Administrator's Reference*.

To change the defaults for a registered node, use the UPDATE NODE command.

Node Compression Considerations

When you enable compression, it reduces network utilization and saves server storage, but causes additional central processing unit (CPU) overhead to the node. Data compression is recommended only when there is insufficient network capacity.

Attention: Use either client compression or drive compression, but not both. For details, see "Using Data Compression" on page 213.

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 application program interface (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.

Registering Nodes with Client Options Sets

Administrators can use client options sets in conjunction with the client options file to register nodes with the server. Client option sets are considered advanced implementation and are discussed in "Managing Client Option Files" on page 340. You can specify an option set for a node when you register or update the node. For example:

```
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.

Registering a Network-attached Storage File Server as a Node

To include a NAS file server as a node that Tivoli Storage Manager can back up and restore with NDMP operations, you can register the file server as a NAS node. Data that is backed up from the NAS file server will be associated with the NAS node name.

The REGISTER NODE and UPDATE NODE commands have a default parameter of TYPE=CLIENT. To register a NAS file server as a node, you must specify the TYPE=NAS parameter. For example, to register a NAS file server with a node name of NASXYZ and a password of PW4PW, enter the following:

```
register node nasxyz pw4pw type=nas
```

You must use this same node name when you later define the corresponding data mover name. For more information, see Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143.

Registering a Source Server as a Node on a Target Server

A virtual volume is a volume that appears to be a sequential media volume on a source server. The volume is actually stored as an archive file on a target server.

To use virtual volumes, register the source server as a client node on the target server.

The REGISTER NODE and UPDATE NODE commands have a default parameter of TYPE=CLIENT. To register a source server as a node, you must specify the TYPE=SERVER parameter. For more information, see "Using Virtual Volumes to Store Data on Another Server" on page 601.

Registering an Application Programming Interface to the Server

Workstation users can request Tivoli Storage Manager services by using an application that uses the Tivoli Storage Manager application programming interface (API). An administrator uses the REGISTER NODE command to register the workstation as a node.

Understanding How to Set the Compression Option

For applications that use the Tivoli Storage Manager API, compression can be determined by:

- 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*.

Understanding How to Set the File Deletion Option

For applications that use the Tivoli Storage Manager API, the file deletion option can be set by:

- 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. On the Windows platform, you can use a wizard to work with the client options file.

Note: 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

Required Client Options

Each node requires a client options file. Each client options file must contain the network address of the Tivoli Storage Manager server and other communication options that allow the node to communicate with the server. Figure 48 shows the contents of a client options file that is configured to connect to the server by using TCP/IP. The communication options specified in the client options file satisfy the minimum requirements for the node to connect with the server.

Figure 48. Client Options File

Non–Required Client Options

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 clients, options are located in three options files: client systems options file, client user options file, and include-exclude options file. Clients on other platforms use a single options file.

Methods for Creating or Updating a Client Options File

There are several methods for creating or updating client options files. The available methods depend on the client platform.

Using a Text Editor

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

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.

Note: 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 machine (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 49.

Figure 49. Stand-alone Environment

Figure 50 on page 316 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 50. 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 from the Administrative Command Line Client

The administrator can register nodes by using the REGISTER NODE command. For more information, refer to *Administrator's Reference*.

Enabling Open Registration

The default registration mode at installation is closed. To change the default to open so users can register their own client nodes, enter:

```
set registration open
```

Configuring the Client Options File to Connect with the Server

Edit the client options file (dsm.opt) in the client directory using a text editor.

Example: Register Three Client Nodes Using the Administrative Command Line

You want to register three workstations from the engineering department and assign them to the ENGPOLDOM policy domain. Before you can assign client nodes to a policy domain, the policy domain must exist. To define a policy domain, see Chapter 13, "Implementing Policies for Client Data," on page 357.

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. For this example, you 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
```

Next, issue the MACRO command. For information on the MACRO command, see the *Administrator's Reference*.

```
macro regeng.mac
```

Chapter 12. Managing Client Nodes

This chapter contains information about managing client nodes that have been installed and configured. For information about installing and configuring client nodes, see Chapter 11, "Adding Client Nodes," on page 309.

Note: The IBM Tivoli Storage Manager server views its registered clients, application clients, and source servers as nodes. The term *nodes* in this chapter refers to the following type of clients and servers as client nodes:

- IBM System Storage Archive Manager application clients
- IBM Tivoli Storage Manager backup-archive clients
- IBM Tivoli Storage Manager source servers registered as nodes on a target server
- Network-attached storage (NAS) file servers using NDMP support

Administrators can manage client nodes and control their access to the server. See the following sections for more information:

Tasks:
"Managing Nodes" on page 318
"Managing Client Access Authority Levels" on page 327
"Managing File Spaces" on page 329
"Managing Client Option Files" on page 340
"Managing IBM Tivoli Storage Manager Sessions" on page 343
"Managing IBM Tivoli Storage Manager Security" on page 348
Concepts:
"Client Nodes and File Spaces" on page 329

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Managing Client Node Registration Techniques

By default, IBM Tivoli Storage Manager provides closed registration as the technique for registering client nodes. Administrators can modify the default with the SET REGISTRATION command. For more information about open and closed registration, see "Accepting Default Closed Registration or Enabling Open Registration" on page 310.

© Copyright IBM Corp. 1993, 2005

Managing Nodes

From the perspective of the server, each client and application client is a node requiring IBM Tivoli Storage Manager services. For information, see "Client Nodes and File Spaces" on page 329. Client nodes can be local or remote to the server. For information, see "Comparing Network-Attached Nodes to Local Nodes" on page 315.

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

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. Since every firewall is different, the firewall administrator may 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 UDPATE 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 storage agent. When a firewall terminates a valid session, unpredictable problems can occur which make processes and sessions appear to be hung on communication I/O. Configuring IBM Tivoli Storage Manager components with known ports facilitates excluding IBM Tivoli Storage Manager sessions from timeout restrictions.

Client-initiated Sessions

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. To enable administrative clients to communicate with a server across a firewall, open the TCP/IP port for the server on the TCPADMINPORT option in the dsmserv.opt file. The default TCP/IP port is the

TCPPORT value. When authentication is turned on, the information that is sent over the wire is encrypted. See the *Backup-Archive Clients Installation and User's Guide* for more information.

Note: 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.

If you select the CLIENTORSERVER option of the SESSIONINITIATION server parameter, the client may start sessions with the server. Or, server-prompted scheduling may be used to prompt the client to connect to the server.

Server-initiated Sessions

To limit the start of backup-archive client sessions to the IBM Tivoli Storage Manager server, you must specify this on the server, and also synchronize the information in the client option file. In either the REGISTER NODE or UPDATE NODE command, select the SERVERONLY option of the SESSIONINITIATION parameter. Provide the HLADDRESS and LLADDRESS client node addresses.

For example,

```
register node fran secretpw hladdress=9.11.521.125 lladdress=1501
 sessioninitiation=serveronly
```

The HLADDRESS specifies the IP address of the client node, and is used whenever the server contacts the client. The LLADDRESS specifies the low level address of the client node and is used whenever the server contacts the client. The client node listens for sessions from the server on the LLADDRESS port number.

If SESSIONINITIATION=SERVERONLY for a node defined on the IBM Tivoli Storage Manager server, the client must have SESSIONINITIATION=SERVERONLY in its option file. In addition, the TCP/IP address of the client must correspond to the information supplied with the HLADDRESS server parameter. Finally, TCPCLIENTPORT in the client option file must correspond to the information supplied with the LLADDRESS server parameter, or the server will not know how to contact the client.

Notes:

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 Services (DNS) to determine the name of client nodes. If your DNS system is not correctly configured, there may be delays or failures in looking up names. The DNSLOOKUP option is available to restrict usage of DNS services. It can be changed dynamically through the SETOPT DNSLOOKUP command.

Table 23. 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 or UPDATE NODE command	SESSIONINITIATION= SERVERONLY	client option file

Table 23. Server-Initiated Sessions (continued)

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
HLADDRESS	REGISTER or UPDATE NODE command	TCP/IP address	TCP/IP address
LLADDRESS	REGISTER or UPDATE NODE command	TCPCLIENTPORT	client option file

Updating Client Node Information

You can use 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.

For example, update client node TOMC to prevent it from deleting archived files from storage pools by entering:

```
update node tomc archdelete=no
```

Renaming Client Nodes

You can rename a client node with the RENAME NODE command. You may need to rename a client node if the workstation network name or host name changes. For example, with UNIX and Linux clients, users define their node name based on the value returned by the HOSTNAME command. When users access the server, their IBM Tivoli Storage Manager user IDs match the host name of their workstations. If the host name changes, you can update a client node user ID to match the new host name.

For example, to rename CAROLH to ENGNODE, enter:

```
rename node carolh engnode
```

ENGNODE retains the contact information and access to backup and archive data that belonged to CAROLH. All files backed up or archived by CAROLH now belong to ENGNODE.

Locking and Unlocking Client Nodes

You can prevent client nodes from accessing the server with the LOCK NODE command. This will prevent client nodes from performing functions such as either backup and restore or archive and retrieve.

You can restore a locked node's access to the server with the UNLOCK NODE command.

For example, to prevent client node MAB from accessing the server, enter:

```
lock node mab
```

To let client node MAB access the server again, enter:

```
unlock node mab
```

See also "Disabling or Enabling Access to the Server" on page 346.

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 have been deleted (see "Deleting File Spaces" on page 339), you can delete the node.

For example, to remove client node DEBBYG, enter:

1. Delete the DEBBYG file space by entering:
   ```
   delete filespace debbyg * type=any
   ```
2. Delete the DEBBYG node by entering:
   ```
   remove node debbyg
   ```

Note: Before you can delete a 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.

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 a 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 machine responsible for performing the backup may change over time, such as with a cluster. Consolidating shared data from multiple machines 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 HSM clients can be directed to backup or restore data on behalf of another node (the target node).

Distributed Workloads to Reduce Backup and Restore Time

Concurrent operations can reduce backup and restore times in environments such as IBM 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 file system into smaller units for backup purposes and not have password coordination issues.

The following is an example of how scheduling would work where workload is distributed, for example in the DB2 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 machines. 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.

Here is how the administrator defines the schedule that does an EEE backup on behalf of NODE_Z:

```
DEFINE SCHEDULE STANDARD BACKUP-SCHED ACTION=<tbd by EEE>
OPTIONS='-ASNODENAME=NODE_Z'
```

Define the association of the schedule to the three nodes:

```
DEFINE ASSOCIATION STANDARD BACKUP-SCHED NODE_A
```

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 machine. A target node can be a logical entity, meaning no physical machine corresponds to the node. Or, it can be a predefined node which corresponds to a physical machine.

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.

Attention:

- 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.

Shared Access Sample Configuration

The following is an example of setting up proxy node authority for shared access. In the example, client agent nodes NODE_1, NODE_2, and NODE_3 all share the same GPFS file system. 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.

1. Define four nodes on the server: NODE_1, NODE_2, NODE_3, and NODE_GPFS.

   ```
   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.

   ```
   grant proxynode target=node_gpfs agent=node_1,node_2,node_3
   ```

3. Define the node name and asnode name for each of the machines 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.

   ```
   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 with the following:

   ```
   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:

   ```
   dsmc schedule
   ```

Displaying Information about Client Nodes

You can display information about client nodes. For example, as a policy administrator, you might query the server about all client nodes assigned to the policy domains for which you have authority. Or you might query the server for detailed information about one client node.

Displaying Information about Client Nodes Assigned to Specific Policy Domains

You can display information about client nodes assigned to specific policy domains. For example, to view information about client nodes that are assigned to STANDARD and ENGPOLDOM policy domains, enter:

```
query node * domain=standard,engpoldom
```

The output from that command may display similar to the following:

```
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      SUN         ENGPOLDOM               <1            1    No
               SOLARIS
```

Displaying Information about a Specific Client Node

You can view information about specific client nodes. For example, to review the registration parameters defined for client node JOE, enter:

```
query node joe format=detailed
```

The resulting report may appear similar to the following:

```
                    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
```

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 3.7.0 server or above, an identical administrative user ID is created at the same time. This user ID has client owner authority over the node by default.

A Web backup-archive client can be accessed from the Administration Center interface. This allows an administrator with the proper authority to perform backup, archive, restore, and retrieve operations on any machine that is running the Web backup-archive client. See the Administration Center for more information.

You can establish access to a Web backup-archive client for help desk personnel that do not have system or policy privileges by granting those users client access authority to the nodes they need to manage. Help desk personnel can then perform activities on behalf of the client node such as backup and restore operations.

A native backup-archive client can log on to IBM Tivoli Storage Manager using their node name and password, or administrative user ID and password. The

administrative user ID password is managed independently from the password that is generated with the ***passwordaccess generate*** client option. The client must have the option ***passwordaccess generate*** specified in their client option file to enable use of the Web backup-archive client.

To use the Web backup-archive client from your web browser, you specify the URL and port number of the IBM Tivoli Storage Manager backup-archive client machine running the Web client. The browser you use to connect to a Web backup-archive client must be Microsoft® Internet Explorer 5.0 or Netscape 4.7 or later. The browser must have the Java Runtime Environment (JRE) 1.3.1, which includes the Java Plug-in software. The JRE is available at http://java.sun.com/getjava.

A Tivoli Storage Manager Version 5.3 or later Web backup-archive client is required in order to hyperlink from the new Java-based Tivoli Storage Manager administrative client to a Tivoli Storage Manager client machine. If you attempt to hyperlink from the Java-based Tivoli Storage Manager administrative client to an earlier version of the Tivoli Storage Manager Web backup-archive client, you will have to re-enter your credentials.

During node registration, you have the option of granting client owner or client access authority to an existing administrative user ID. You can also prevent the server from creating an administrative user ID at registration. If an administrative user ID already exists with the same name as the node being registered, the server registers the node but does not automatically create an administrative user ID. This process also applies if your site uses open registration.

For more information about installing and configuring the Web backup-archive client, refer to *Backup-Archive Clients Installation and User's Guide*.

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 describes 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 machine, you can delete file spaces or archive data.

The user ID with client owner authority can also access the data from another machine using the –NODENAME 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 machine using the –NODENAME 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. See "Example: Setting up Help Desk Access to Client Machines in a Specific Policy Domain" on page 328 for more information.

Granting Client Authority

To grant client *access* authority to administrator FRED for the LABCLIENT node, issue:

```
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.

To grant client *owner* authority to ADMIN1 for the STUDENT1 node, issue:

```
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 use 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. For example, you want to register client node DESK2, issue:

```
register node desk2 pass2dsk
```

The following shows the output from 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. For example, you want to register DESK2 without creating an administrative user ID with client owner authority by default. Issue the following:

```
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. For example, to give client owner authority to the HELPADMIN user ID when registering the NEWCLIENT node, enter:

```
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 Machines in a Specific Policy Domain

You want to set up help desk access for user HELP1 to the client nodes in the FINANCE domain. You want to grant HELP1 client access authority to the FINANCE domain without having to grant system or policy privileges.

The client nodes have been previously set up as follows:

- Installed and configured. The URL and port numbers were specified during the REGISTER NODE process.
- Assigned to the FINANCE policy domain.
- Started the Client Acceptor service.
- Specified ***passwordaccess generate*** option in their client option files.

The help desk person, using HELP1 user ID, has a Web browser with Java Runtime Environment (JRE) 1.3.1.

1. Register an administrative user ID of HELP1.

   ```
   register admin help1 05x23 contact="M. Smith, Help Desk x0001"
   ```

2. Grant the HELP1 administrative user ID client access authority to all clients in the FINANCE domain. With client access authority, HELP1 can perform backup and restore operations for clients in the FINANCE domain. Client nodes in the FINANCE domain are Dave, Sara, and Joe.

```
grant authority help1 class=node authority=access domains=finance
```

The following is output 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 machine Sara:

   ```
   http://sara.machine.name:1581
   ```

   A Java applet is started, and the client hub window is displayed in the main window of the Web browser. When HELP1 accesses the backup function from the client hub, the IBM Tivoli Storage Manager login screen is displayed in a separate Java applet window. HELP1 authenticates with the administrative user ID and password. HELP1 can perform a backup for Sara.

For information about what functions are not supported on the Web backup-archive client, refer to *Backup-Archive Clients Installation and User's Guide*.

Managing File Spaces

A *file space name* identifies a group of files that are stored as a logical unit in server storage. Administrators manage file spaces in which IBM Tivoli Storage Manager stores each client node's data. See "Client Nodes and File Spaces" for more information.

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

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 machine 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 machine as in the case of a SQL server machine 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 machine. 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.

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*.

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 and Macintosh OS X 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 set-up. 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. Once 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.

Note: Once 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 only on Windows and Macintosh OS X. 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.

Note: The remainder of this section will refer to these clients as Unicode-enabled clients, users of Windows and Macintosh operating systems, or clients.

It is strongly recommended that users of Windows and Macintosh 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*.

See the following sections:

"Reasons for Migrating Clients to Unicode-Enabled File Spaces"
"Migrating Clients to Unicode-Enabled File Spaces" on page 332
"Querying Unicode-enabled File Spaces" on page 338
"Unicode-enabled Clients and Existing Backup Sets" on page 338

Reasons for Migrating Clients to Unicode-Enabled File Spaces

Without IBM Tivoli Storage Manager support for storing Unicode-enabled file spaces, some clients have experienced 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. See "Migrating Clients to Unicode-Enabled File Spaces" on page 332.

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.

Note: 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.

Migrating Clients to Unicode-Enabled File Spaces

To allow clients with existing data to migrate to Unicode-enabled file spaces, IBM Tivoli Storage Manager provides an automatic rename function for file spaces. When enabled, IBM Tivoli Storage Manager uses the rename function when it recognizes that a file space that is not Unicode-enabled in server storage matches the name of a file space on a client. The existing file space in server storage is renamed, so that the file space in the current operation is then treated as a new, Unicode-enabled file space. For example, if the operation is an incremental backup at the file space level, the entire file space is then backed up to the server as a Unicode-enabled file space.

The following example shows how this process works when automatic renaming is enabled from the server, for an existing client node that has file spaces in server storage.

1. The administrator updates a client node definition by issuing an UPDATE NODE command with the parameter, AUTOFSRENAME YES.
2. The client processes an incremental back up.
3. IBM Tivoli Storage Manager processes the back up as follows:
   a. Renames the existing file space (_OLD)
   b. Creates a new Unicode-enabled file space
   c. Processes the back up in the current operation to the new Unicode-enabled file space

Attention: If you force the file space renaming for all clients at the same time, backups can contend for network and storage resources, and storage pools can run out of storage space.

Before you allow automatic renaming of file spaces for Unicode-enabled IBM Tivoli Storage Manager clients, read the following sections.

"Options for Automatically Renaming File Spaces"
"The Rules for Automatically Renaming File Spaces" on page 334
"Planning for Unicode Versions of Existing Client File Spaces" on page 334
"How Clients are Affected by the Migration to Unicode" on page 336
"Example of a Migration Process" on page 337

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.

Note: The setting for AUTOFSRENAME affects only clients that are Unicode-enabled.

You have these options:

- Do not allow existing file spaces to be renamed, so that Unicode-enabled file spaces are not created (AUTOFSRENAME=NO, the default).

  IBM Tivoli Storage Manager does not automatically rename client file spaces when the client system upgrades to the Unicode-enabled IBM Tivoli Storage Manager client. This setting can help an administrator control how many clients' file spaces can be renamed at one time. The administrator can determine how many Unicode-enabled clients exist by using the QUERY NODE FORMAT=DETAILED command. The output displays the client level. A Unicode-enabled client is on a Windows and Macintosh OS X system at IBM Tivoli Storage Manager Version 4.2.0 or higher.
- Automatically rename existing file spaces, forcing the creation of Unicode-enabled file spaces in place of the renamed file spaces (AUTOFSRENAME=YES).

  IBM Tivoli Storage Manager automatically renames client file spaces in server storage when the client upgrades to the Unicode-enabled client and runs one of the following operations: archive, selective backup, full incremental backup, or partial incremental backup. IBM Tivoli Storage Manager automatically renames the file spaces that are specified in the current operation and creates new, Unicode-enabled file spaces where files and directories are stored to complete the operation. Other file spaces that are not specified in the current operation are not affected by the rename. This means a client can have mixed file spaces. See "The Rules for Automatically Renaming File Spaces" on page 334 for how the new name is constructed.

  Attention: If you force the file space renaming for all clients at the same time, client operations can contend for network and storage resources, and storage pools can run out of storage space.
- Allow clients to choose whether to rename files spaces, in effect choosing whether new Unicode-enabled file spaces are created (AUTOFSRENAME=CLIENT).

  If you use this value for a client node, the client can set its AUTOFSRENAME option in its options file. The client option determines whether file spaces are renamed (YES or NO), or whether the user is prompted for renaming at the time of an IBM Tivoli Storage Manager operation (PROMPT).

  The default value for the client option is PROMPT. When the option is set for prompting, the client is presented with a choice about renaming file spaces. When a client that has existing file spaces on server storage upgrades to the Unicode-enabled client, and the client runs an IBM Tivoli Storage Manager operation with the server, the user is asked to choose whether to rename the file spaces that are involved in the current operation.

  The client is prompted only once about renaming a particular file space.

  If the client does not choose to rename the file space, the administrator can later rename the file space so that a new Unicode-enabled file space is created the next time the client processes an archive, selective backup, full incremental backup, or partial incremental backup.

Attention: There is no prompt for operations that run with the client scheduler. If the client is running the scheduler and the client AUTOFSRENAME option is set to PROMPT, there is no prompt and the file space is not renamed. This allows a client session to run unattended. The prompt appears during the next interactive session on the client.

The following table summarizes what occurs with different parameter and option settings.

Table 24. Effects of AUTOFSRENAME Settings

Parameter on the server (for each client)	Option on the client	Result for file spaces	Is the file space renamed?
Yes	Yes, No, Prompt	Renamed	Yes
No	Yes, No, Prompt	Not renamed	No
Client	Yes	Renamed	Yes
	No	Not renamed	Yes
	Prompt	Command-line or GUI: The user receives a one-time only prompt about renaming	Depends on the response from the user (yes or no)
		Client Scheduler: Not renamed (prompt appears during the next command-line or GUI session)	No

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
Original file space	\\maria\c$
Renamed file space	\\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	
Original file space	\\maria\c$	Other existing file spaces: \\maria\c$_OLD \\maria\c$_OLD1
Renamed file space	\\maria\c$_OLD2	

If the new name for the file space exceeds the limit of 64 characters, the file space name is truncated on the right before the suffix _OLD is added.

Planning for Unicode Versions of Existing Client File Spaces: You need to consider the following factors in your planning:

- After clients with existing file spaces start to create Unicode-enabled file spaces, they will still need to have access to the renamed file spaces that are not Unicode-enabled for some period of time.

- Your storage pool and database space requirements can double if you allow all clients to create Unicode-enabled file spaces in addition to their existing file spaces that are not Unicode-enabled.
- Because the initial backups after migration are complete backups, it can also greatly increase the time required to finish backup operations.

To minimize problems, you need to plan the storage of Unicode-enabled file spaces for clients that already have existing file spaces in server storage.

1. Determine which clients need to migrate.

   Clients that have had problems with backing up files because their file spaces contain names of directories or files that cannot be converted to the server's code page should have the highest priority. Balance that with clients that are most critical to your operations. If you have a large number of clients that need to become Unicode-enabled, you can control the migration of the clients.

   Change the rename option for a few clients at a time to keep control of storage space usage and processing time. Also consider staging migration for clients that have a large amount of data backed up.
2. Allow for increased backup time and network resource usage when the Unicode-enabled file spaces are first created in server storage.

   Based on the number of clients and the amount of data those clients have, consider whether you need to stage the migration. Staging the migration means setting the AUTOFSRENAME parameter to YES or CLIENT for only a small number of clients every day.

   Note: If you set the AUTOFSRENAME parameter to CLIENT, be sure to have the clients (that run the client scheduler) set their option to AUTOFSRENAME YES. This ensures the file spaces are renamed.
3. Check the current storage usage for the clients that need to become Unicode-enabled.

   You can use the QUERY OCCUPANCY command to display information on how much space each client is currently using. Initially, clients will need only the amount of space used by active files. Therefore, you need to estimate how much of the current space is used by copies (different versions of the same file). Migration will result in a complete backup at the next incremental backup, so clients will need space for that backup, plus for any other extra versions that they will keep. Therefore, the amount of storage required also depends on policy (see the next step). Your IBM Tivoli Storage Manager policy specifies how files are backed up, archived, migrated from client node storage, and managed in server storage.
4. Understand how your IBM Tivoli Storage Manager policies affect the storage that will be needed.

   If your policies expire files based only on the number of versions (Versions Data Exists), storage space required for each client will eventually double, until you delete the old file spaces.

   If your policies expire files based only on age (Retain Extra Versions), storage space required for each client will increase initially, but will not double.

   If your policies use both the number of versions and their age, each client will need less than double their current usage.
5. Estimate the effect on the database size.

   The database size depends on the number of files in server storage, as well as the number of versions of those files. As Unicode-enabled file spaces are backed up, the original file spaces that were renamed remain. Therefore, the

server requires additional space in the database to store information about the increased number of file spaces and files.

See "Estimating and Monitoring Database and Recovery Log Space Requirements" on page 494.

6. Arrange for the additional storage pool space, including space in copy storage pools, based on your estimate from step 3 on page 335 and 4 on page 335.
7. Check the server database space that is available and compare with your estimate from step 5 on page 335.
8. Ensure that you have a full database backup *before* you proceed with migration of Unicode-enabled file spaces. See "Backing Up the Database" on page 652.
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.

How Clients are Affected by the Migration to Unicode: 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.

  Note: 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.

  Note: 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:

  1. Sue had been backing up her file space, \\sue-node\d$.
  2. Sue upgrades the IBM Tivoli Storage Manager client on her system to the Unicode-enabled IBM Tivoli Storage Manager client.
  3. Sue performs a selective backup of the file HILITE.TXT.

4. 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$. This new Unicode-enabled file space contains only the HILITE.TXT file.
5. All other directories and files in Sue's file system will be backed up on the next full incremental backup. If Sue needs to restore a file before the next full incremental backup, she can restore the file from the \\sue-node\d$_OLD file space.

Refer to the *Backup-Archive Clients Installation and User's Guide* for more information.

Example of a Migration Process: This section gives one possible sequence for migrating clients. Assumptions for this scenario are:

- The IBM Tivoli Storage Manager server database has been backed up.
- The latest server software has been installed. This installation has also performed an upgrade to the server database.
- Clients have installed the latest software.
- A few clients are file servers. Most clients are workstations used by individuals.
- Clients generally run scheduled incremental backups every night.

The following is a possible migration process:

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, enter 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. (See "The Rules for Automatically Renaming File Spaces" on page 334.)
   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.
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. See "Managing the Renamed File Spaces."

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:

```
query filespace
```

```
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. See "Displaying Information about File Spaces" for details.

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. For information about restoring backup sets, see "Restoring Backup Sets from a Backup-Archive Client" on page 414.

Displaying Information about File Spaces

You can display file space information to:

- Identify file spaces defined to each client node, so that you can delete each file space from the server before removing the client node from the server
- Identify file spaces that are Unicode-enabled and identify their file space ID (FSID)

- Monitor the space used on workstation's disks
- Monitor whether backups are completing successfully for the file space
- Determine the date and time of the last backup

You display file space information by identifying the client node name and file space name.

Note: File space names are case-sensitive and must be entered exactly as known to the server.

For example, to view information about file spaces defined for client node JOE, enter:

```
query filespace joe *
```

The following figure shows the output from this command.

```
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
```

When you display file space information in detailed format, the Filespace Name field may display file space names as "...". This indicates to the administrator that a file space does exist but could not be converted to the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space name or file name may display with a combination of invalid characters or blank spaces. Refer to *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. For more information see, "Moving Data for a Client Node" on page 298.

Deleting File Spaces

You can delete a client node from a server, but first you must delete all of that client's data from server storage by deleting any file spaces that belong to the node.

Administrators may want to delete a file space in the following cases:

- Users are not authorized to delete backed-up or archived files in storage pools.

  For example, client node PEASE no longer needs archived files in file space */home/pease/dir2*. However, he does not have the authority to delete those files. You can delete them by entering:

  ```
  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. See "Accepting Default Closed Registration or Enabling Open Registration" on page 310 for information on allowing users to delete files in storage pools.

- You want to remove a client node from the server.

  You must delete a user's files from storage pools before you can remove a client node. For example, to delete all file spaces belonging to client node DEBBYG, enter:

  ```
  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. See "Deleting Client Nodes" on page 321 for more details.

- You want to delete a specific user's files.

  For client nodes that support multiple users, such as UNIX or Linux, a file owner name is associated with each file on the server. The owner name is the user ID of the operating system, such as the UNIX Linux user ID. When you delete a file space belonging to a specific owner, only files that have the specified owner name in the file space are deleted.

When a node has more than one file space and you issue a DELETE FILESPACE command for only one file space, a QUERY FILESPACE command for the node during the delete process shows no file spaces. When the delete process ends, you can view the remaining file spaces with the QUERY FILESPACE command. If data retention protection is enabled, the only files which will be deleted from the file space are those which have met the retention criterion. The file space will not be deleted if one or more files within the file space cannot be deleted.

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. For more information about client option files, see "Connecting Nodes with the Server" on page 313.

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. See the following sections:

- "Creating Client Option Sets on the Server"
- "Managing Client Option Sets" on page 342

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 file described in "Connecting Nodes with the Server" on page 313.

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 *Backup-Archive Clients Installation and User's Guide* for detailed information about individual client options.

To create a client option set and have the clients use the option set, do the following:

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.

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. For example:

```
define cloptset engbackup description='Backup options for eng. dept.'
```

Note: 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. The following example shows how to add a client option in the ENGBACKUP option set.

```
define clientopt engbackup schedlogretention 5
```

For a list of valid client options you can specify, refer to *Administrator's Reference*.

The server automatically assigns sequence numbers to the specified options, or you can choose to specify the sequence number for order of processing. This is helpful if you have defined more than one of the same option as in the following example.

```
define clientopt engbackup inclexcl "include d:\admin"
define clientopt engbackup inclexcl "include d:\payroll"
```

A sequence number of 0 is assigned to the option `include d:\admin`. A sequence number of 1 is assigned to the option `include d:\payroll`. If you want to specifically process one option before another, include the sequence parameter as follows:

```
define clientopt engbackup inclexcl "include d:\admin" seqnumber=2"
define clientopt engbackup inclexcl "include d:\payroll" seqnumber=1"
```

The options are processed starting with the highest sequence number.

Any include-exclude statements in the server client option set have priority over the include-exclude statements in the local client options file. The server include-exclude statements are always enforced and placed at the bottom of the include-exclude list and evaluated before the client include-exclude statements. If the server option set has several include-exclude statements, the statements are processed starting with the highest sequence number. The client can use the QUERY INCLEXCL command to view the include-exclude statements in the order they are processed. QUERY INCLEXCL also displays the source of each include-exclude statement. For more information on the processing of the include-exclude statements see "The Include-Exclude List" on page 368 and also 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 effect 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
```

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 as follows:

```
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 5 days.

Managing Client Option Sets

Administrators can perform the following activities to manage client option sets:

Task	Required Privilege Class
Updating the sequence number for a client option	System or unrestricted policy
Deleting an option from a client option set	System, unrestricted policy, or restricted policy
Copying a client option set	System, unrestricted policy, or restricted policy
Displaying client option set information	Any administrator
Updating the client option set description	System, unrestricted policy, or restricted policy
Deleting a client option set	System, unrestricted policy, or restricted policy

Updating the Sequence Number for a Client Option

You can update the sequence number for a client option to change its processing order. This is helpful if you have more than one of the same option, for example several INCLUDE options.

The following example shows how to change the sequence number for the DATEFORMAT option from 0 to 9:

```
update clientopt engbackup dateformat 0 9
```

Deleting an Option from a Client Option Set

You can remove an option that is defined in a client option set. The following example shows how to remove the SCHEDMODE polling option from the financeschd option set:

```
delete clientopt financeschd schedmode
```

Copying a Client Option Set

You can copy an existing client option to another option set. The following example shows how to copy the engbackup option set to financeschd option set:

```
copy cloptset engbackup financeschd
```

Requesting Information about a Client Option Set

To display information about the contents of a client option set, issue the following command:

```
query cloptset financeschd
```

Updating the Description for a Client Option Set

You can update the description for a client option set. The following example shows how to update the description for the engbackup option set:

```
update clopset engbackup description='Scheduling information'
```

Deleting 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 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. See "Managing Client Restartable Restore Sessions" on page 346 for more information.

Tivoli Storage Manager can hold a client restore session in DSMC loop mode until one of these conditions is met:

- The device class MOUNTRETENTION limit is satisfied.
- The client IDLETIMEOUT period is satisfied.
- The loop session ends.

Administrators can perform the following activities when managing IBM Tivoli Storage Manager sessions:

Task	Required Privilege Class
Displaying information about client sessions	Any administrator
Canceling a client session	System or operator
Disabling or enabling a client session	System or operator
Freeing links for client connections	Administrator with root authority

Displaying Information about IBM Tivoli Storage Manager Sessions

Each client session is assigned a unique session number. To display information about client sessions, enter:

```
query session
```

Figure 51 shows a sample client session report.

```
  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 51. Information about Client Sessions

You can determine the state of the server by examining the *session state* and *wait 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 one of the following:

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. For more information, see "Reclaiming Space in Sequential Access Storage Pools" on page 267.

Note: 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 IDLETIMEOUT 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.

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 machine is not responding or as a prerequisite to halting the server. Administrators can display a session number with the QUERY SESSION command as described in "Displaying Information about IBM Tivoli Storage Manager Sessions" on page 343.

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. For details, see "Server Session States" on page 344.

If the session you cancel is currently waiting for a media mount, the mount request is automatically canceled. If a volume associated with the client session is currently being mounted by an *automated* library, the cancel may not take effect until the mount is complete.

For example, to cancel a session for client MARIE:

1. Query client sessions to determine the session number as shown Figure 51 on page 343. The example report displays MARIE's session number 6.
2. Cancel node MARIE's session by entering:

```
cancel session 6
```

If you want to cancel all backup and archive sessions, enter:

```
cancel session all
```

When a Client Session is Automatically Canceled

Client sessions can be automatically canceled 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

Task	Required Privilege Class
Disabling and enabling client node access to the server	System or operator
Displaying server status	Any administrator

You can prevent clients from establishing sessions with the server by using the DISABLE SESSIONS command. This command does not cancel sessions currently in progress or system processes such as migration and reclamation. For example, to disable client node access to the server, enter:

```
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 entering:

```
enable sessions
```

You can issue the QUERY STATUS command to determine if the server is enabled or disabled.

See also "Locking and Unlocking Client Nodes" on page 320.

Managing Client Restartable Restore Sessions

Some large restore operations may invoke a special type of restore operation called client restartable restore sessions. These special sessions allow users to restart the restore session from where it left off if the session was interrupted. IBM Tivoli Storage Manager identifies client restartable restore sessions by displaying message ANS1247I on the client machine when the sessions start. These restore sessions can be restarted as long as the restore interval has not expired.

Following restore operations 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 is 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 *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. For example, to determine which client nodes have eligible restartable restore sessions, enter:

```
query restore
```

Restartable restore sessions have a negative session number.

Canceling a Client Restartable Restore Session

When a client restore session is in a restartable state, the file space is locked in server storage and no files can be moved from sequential volumes. This prevents the data from being migrated, moved, reclaimed, or backed up by another operation. These sessions will automatically expire when the specified restore interval has passed.

An administrator can cancel a restartable restore session that is in an active or restartable state. If the restore session is active, any outstanding mount requests related to the active session are automatically canceled. When a restartable restore session is canceled with the CANCEL RESTORE command, it cannot be restarted from the point of interruption. A restartable restore session always has a negative session number.

To cancel a restartable restore session, you must specify the session number. For example:

```
cancel restore -1
```

Interrupting an Active Client Restartable Restore Session

An administrator can interrupt an active restartable restore session and have the option to later restart the session from its point of interruption by canceling the session.

```
cancel session -2
```

Managing IBM Tivoli Storage Manager Security

Administrators can perform the following activities to manage IBM Tivoli Storage Manager security.

Tasks:
"Managing Access to the Server and Clients" on page 350
"Managing IBM Tivoli Storage Manager Administrators" on page 351
"Managing Levels of Administrative Authority" on page 353
"Managing Passwords and Login Procedures" on page 354
Concepts:
"The Server Console"
"Administrative Authority and Privilege Classes"

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 off.

Administrative Authority and Privilege Classes

After administrators are registered, they can perform a limited set of tasks. By default, administrators can request command-line help and issue queries.

To perform other tasks, administrators must be granted authority by being assigned one or more administrative privilege classes. Privilege classes determine the authority level for an administrator. Figure 52 on page 349 illustrates the privilege classes. An administrator with system privilege class can perform any task with the server. Administrators with policy, storage, operator, analyst, or node privileges can perform subsets of tasks.

Note: 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 *Administrator's Reference* for details on server options.

Figure 52. Administrative Privilege Classes

Table 25 summarizes the privilege classes, and gives examples of how to set privilege classes. For more information, see "Managing Levels of Administrative Authority" on page 353.

Table 25. 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 25. 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. (See "Overview of Remote Access to Web Backup-Archive Clients" on page 325.)
Analyst `grant authority marysmith classes=analyst`	Reset the counters that track IBM Tivoli Storage Manager server statistics.

Managing Access to the Server and Clients

An administrator can control access to the server and clients by a number of methods. See Table 26 for a summary.

Table 26. Managing Access

Task	Details
Allow a new administrator to access the server	1. "Registering Administrators" on page 351 2. "Granting Authority to Administrators" on page 353
Modify authority for registered administrators	"Managing Levels of Administrative Authority" on page 353
Give a user authority to access a client remotely	"Managing Client Access Authority Levels" on page 327
Give an administrator authority to create a backup set for a client node	"Generating Client Backup Sets on the Server" on page 413
Prevent administrators from accessing the server	"Locking and Unlocking Administrators from the Server" on page 352
Prevent new sessions with the server, but allow current sessions to complete	"Disabling or Enabling Access to the Server" on page 346
Prevent clients from accessing the server	"Locking and Unlocking Client Nodes" on page 320
Change whether passwords are required to access IBM Tivoli Storage Manager	"Disabling the Default Password Authentication" on page 355
Change requirements for passwords	• "Modifying the Default Password Expiration Period" on page 354 • "Setting a Limit for Invalid Password Attempts" on page 355 • "Setting a Minimum Length for a Password" on page 355
Prevent clients from initiating sessions within a firewall **Note:** For information on connecting with IBM Tivoli Storage Manager across a firewall, refer to the *Installation Guide.*	"Server-initiated Sessions" on page 319

Managing IBM Tivoli Storage Manager Administrators

The administrator is responsible for registering other administrators, granting levels of authority to administrators, renaming or removing administrators, and for locking and unlocking administrators from the server.

Task	Required Privilege Class
Registering an administrator	System
Granting administrative authority	System
Updating information about other administrators	System
Updating information about yourself	Any administrator
Displaying information about administrators	Any administrator
Renaming an administrator user ID	System
Removing administrators	System
Locking or unlocking administrators from the server	System

Registering Administrators

An administrator registers other administrators with the REGISTER ADMIN command.

To register an administrator with a user ID of DAVEHIL, the password *birds*, and a password expiration period of 120 days, enter the REGISTER ADMIN command:

```
register admin davehil birds passexp=120 contact='backup team'
```

Updating Information about Other Administrators

An administrator can reset another administrator's password with the UPDATE ADMINISTRATOR command. For example, administrator DAVEHIL changes his password to *ganymede*, by issuing the following command:

```
update admin davehil ganymede
```

Note: The SERVER_CONSOLE administrator's ID and contact information cannot be updated.

Renaming an Administrator

You can rename an administrator ID when an employee wants to be identified by a new ID, or you want to assign an existing administrator ID to another person. You cannot rename an administrator ID to one that already exists on the system.

For example, if administrator HOLLAND leaves your organization, you can assign administrative privilege classes to another user by completing the following steps:

1. Assign HOLLAND's user ID to WAYNESMITH by issuing the RENAME ADMIN command:

   ```
   rename admin holland waynesmith
   ```

   By renaming the administrator's ID, you remove HOLLAND as a registered administrator from the server. In addition, you register WAYNESMITH as an administrator with the password, contact information, and administrative privilege classes previously assigned to HOLLAND.

2. Change the password to prevent the previous administrator from accessing the server by entering:

```
update admin waynesmith new_password contact="development"
```

Note: The administrator SERVER_CONSOLE cannot be renamed. See "The Server Console" on page 348.

Removing Administrators

You can remove administrators from the server so that they no longer have access to administrator functions. For example, to remove registered administrator ID SMITH, enter:

```
remove admin smith
```

Notes:

1. You cannot remove the last system administrator from the system.
2. You cannot remove the administrator SERVER_CONSOLE. See "The Server Console" on page 348 for more information.

Displaying Information about Administrators

Any administrator can query the server to display administrator information. You can restrict the query to all administrators authorized with a specific privilege class.

For example, to query the system for a detailed report on administrator ID DAVEHIL, issue the QUERY ADMIN command:

```
query admin davehil format=detailed
```

Figure 53 displays a detailed report.

```
        Administrator Name: DAVEHIL
      Last Access Date/Time: 2002.09.04 17.10.52
    Days Since Last Access: <1
    Password Set Date/Time: 2002.09.04 17.10.52
   Days Since Password Set: 26
     Invalid Sign-on Count: 0
                   Locked?: No
                   Contact:
          System Privilege: Yes
          Policy Privilege: **Included with system privilege**
         Storage Privilege: **Included with system privilege**
         Analyst Privilege: **Included with system privilege**
        Operator Privilege: **Included with system privilege**
   Client Access Privilege: **Included with system privilege**
    Client Owner Privilege: **Included with system privilege**
    Registration Date/Time: 05/09/2002 23:54:20
 Registering Administrator: SERVER_CONSOLE
          Managing profile:
Password Expiration Period:  90 Day (s)
```

Figure 53. A Detailed Administrator Report

Locking and Unlocking Administrators from the Server

You can lock out other administrators to temporarily prevent them from accessing IBM Tivoli Storage Manager by using the LOCK ADMIN command.

For example, administrator MARYSMITH takes a leave of absence from your business. You can lock her out by entering:

```
lock admin marysmith
```

When she returns, any system administrator can unlock her administrator ID by entering:

```
unlock admin marysmith
```

MARYSMITH can now access the server to complete administrative tasks.

You cannot lock or unlock the SERVER_CONSOLE ID from the server. See "The Server Console" on page 348 for details.

Managing Levels of Administrative Authority

A privilege class is a level of authority granted to an administrator. The privilege class determines which administrative tasks the administrator can perform. See "Administrative Authority and Privilege Classes" on page 348 and *Administrator's Reference* about the activities that administrators can perform with each privilege class.

You can perform the following activities to manage levels of authority:

Task	Required Privilege Class
Granting a level of authority to an administrator	System
Modifying the level of authority for an administrator	System

Granting Authority to Administrators

You can grant authority with the GRANT AUTHORITY command. For example, to grant to administrator JONES restricted policy privilege for the domain ENGPOLDOM, enter the following command:

```
grant authority jones domains=engpoldom
```

Extending Authority for Administrators

You can grant and extend authority with the GRANT AUTHORITY command. If an ID already has some level of authority, granting additional authority adds to any existing privilege classes; it does not override those classes.

For example, JONES has restricted policy privilege for policy domain ENGPOLDOM. Enter the following command to extend JONES' authority to policy domain MKTPOLDOM and add operator privilege:

```
grant authority jones domains=mktpoldom classes=operator
```

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 enter the following command:

```
grant authority holland stgpools=tape*
```

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, enter:

```
grant authority holland stgpools=tapepool4
```

Reducing Authority for Administrators

You can revoke part of an administrator's authority with the REVOKE AUTHORITY command. For example, rather than revoking all of the privilege classes for administrator JONES, you want to revoke only the operator authority and the policy authority to policy domain MKTPOLDOM. You enter the following command:

```
revoke authority jones classes=operator domains=mktpoldom
```

JONES still has policy privilege to the ENGPOLDOM policy domain.

Reducing Privilege Classes

You can reduce an administrator's authority simply by revoking one or more privilege classes and granting one or more other classes.

For example, administrator HOGAN has system authority. To reduce authority for HOGAN to the operator privilege class, do the following:

1. Revoke the system privilege class by entering:
   ```
   revoke authority hogan classes=system
   ```
2. Grant operator privilege class by entering:
   ```
   grant authority hogan classes=operator
   ```

Revoking Authority for Administrators

You can revoke an administrator's authority with 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 enter:

```
revoke authority jones
```

Managing Passwords and Login Procedures

By default, IBM Tivoli Storage Manager requires authorized administrators and nodes to identify themselves to the server with a password.

Administrators can perform the following activities to manage passwords and login procedures:

Task	Required Privilege Class
Modifying the default timeout period for the Administration Center	System
Modifying the default password expiration period	System
Setting the limit for invalid password attempts	System
Setting the minimum length for passwords	System
Disabling the default password authentication	System

Modifying the Default Password Expiration Period

By default, the server sets a password expiration of 90 days. The expiration period begins when an administrator or client node is first registered to the server. If a user password is not changed within this period, the server prompts the user to change the password the next time the user tries to access the server.

To set the password expiration period for selected administrators or client nodes, you must specify the administrator or node names with the ADMIN or NODE parameter with the SET PASSEXP command. If you set the expiration period only

for selected users, you may set the expiration period from 0–9999 days. A value of 0 means that user's password never expires. For example, to set the expiration period of client node LARRY to 120 days, issue the following command:

```
set passexp 120 node=larry
```

Once you have explicitly set a password expiration for a node or administrator, it is not modified if you later set a password expiration for all users. You can use the RESET PASSEXP command to reset the password expiration period to the common expiration period. Use the QUERY STATUS command to display the common password expiration period, which at installation is set to 90 days.

Setting a Limit for Invalid Password Attempts

By default, IBM Tivoli Storage Manager does not check the number of times a user attempts to log in with an invalid password. You can set a limit on consecutive invalid password attempts for all client nodes. When the limit is exceeded, the server locks the node. The following example sets a system-wide limit of three consecutive invalid password attempts:

```
set invalidpwlimit 3
```

The default value at installation is 0. A value of 0 means that invalid password attempts are not checked. You can set the value from 0 to 9999 attempts.

If you initially set a limit of 4 and then change the limit to a lower number, some clients may fail verification during the next login attempt.

After a client node has been locked, only a storage administrator with proper authority can unlock the node. For information about unlocking a client or administrator node, see "Locking and Unlocking Client Nodes" on page 320 and "Locking and Unlocking Administrators from the Server" on page 352.

An administrator can also force a client to change their password on the next login by specifying the FORCEPWRESET=YES parameter on the UPDATE NODE or UPDATE ADMIN command. For more information, refer to *Administrator's Reference.*

Setting a Minimum Length for a Password

By default, IBM Tivoli Storage Manager does not check the length of a password. The administrator can specify a minimum password length that is required for IBM Tivoli Storage Manager passwords. The following example shows how to set the minimum password length to eight characters:

```
set minpwlength 8
```

The default value at installation is 0. A value of 0 means that password length is not checked. You can set the length value from 0 to 64.

Disabling the Default Password Authentication

By default, the server automatically sets password authentication on. With password authentication set to on, all users must enter a password when accessing the server. To allow administrators and client nodes to access the server without entering a password, issue the following command:

```
set authentication off
```

Attention: Setting password authentication off reduces data security.

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. See "Basic Policy Planning" on page 358 for information about the standard policy.

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. See "File Expiration and Expiration Processing" on page 361 and "Running Expiration Processing to Delete Expired Files" on page 391 for details.

You may need more flexibility in your policies than the standard policy provides. To accommodate individual user's needs, you may fine tune the STANDARD policy (see "Getting Users Started" on page 360 for details), or create your own policies (see "Creating Your Own Policies" on page 376 for details). 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 (see "Setting Policy to Enable Point-in-Time Restore for Clients" on page 404 for more information).

Policy can be distributed from a configuration manager to managed servers. See Chapter 21, "Working with a Network of IBM Tivoli Storage Manager Servers," on page 563 for more information on distributing configurations.

See the following sections:

Concepts:
"Basic Policy Planning" on page 358
"The Standard Policy" on page 359
"File Expiration and Expiration Processing" on page 361
"Client Operations Controlled by Policy" on page 362
"The Parts of a Policy" on page 364
"More on Management Classes" on page 367
"How IBM Tivoli Storage Manager Selects Files for Policy Operations" on page 372
"How Client Migration Works with Backup and Archive" on page 376

© Copyright IBM Corp. 1993, 2005

Tasks:
“Getting Users Started” on page 360
“Changing Policy” on page 360
“Creating Your Own Policies” on page 376
“Defining and Updating a Policy Domain” on page 378
“Defining and Updating a Policy Set” on page 379
“Defining and Updating a Management Class” on page 380
“Defining and Updating a Backup Copy Group” on page 381
“Defining and Updating an Archive Copy Group” on page 387
“Assigning a Default Management Class” on page 389
“Validating and Activating a Policy Set” on page 389
“Assigning Client Nodes to a Policy Domain” on page 391
“Running Expiration Processing to Delete Expired Files” on page 391
“Protection and Expiration of Archive Data” on page 393
“Policy Configuration Scenarios” on page 399
“Configuring Policy for Direct-to-Tape Backups” on page 399
“Configuring Policy for Tivoli Storage Manager Application Clients” on page 400
“Policy for Logical Volume Backups” on page 400
“Configuring Policy for NDMP Operations” on page 401
“Configuring Policy for LAN-free Data Movement” on page 402
“Policy for IBM Tivoli Storage Manager Servers as Clients” on page 404
“Setting Policy to Enable Point-in-Time Restore for Clients” on page 404
“Distributing Policy Using Enterprise Configuration” on page 405
“Querying Policy” on page 405
“Deleting Policy” on page 408

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see “Managing Servers with the New Administration Center” on page 36.

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 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.

See "The Standard Policy" for more details about the standard policy.

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. See "File Expiration and Expiration Processing" on page 361 and "Running Expiration Processing to Delete Expired Files" on page 391 for details.

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. See "The Parts of a Policy" on page 364 for details. The attributes of the default policy are as follows:

Table 27. 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

Table 27. Summary of Default Policy (continued)

Policy	Object where the policy is set
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
Space Management (HSM) Policy	
Client files are not space-managed (there are no HSM clients).	STANDARD management class, SPACEMGTECHNIQUE parameter

Getting Users Started

When you register a client node, the default is to assign the node to the STANDARD policy domain. If users register their own workstations during open registration, they are also assigned to the STANDARD policy domain.

To help users take advantage of IBM Tivoli Storage Manager, you can further tune the policy environment by doing the following:

- Define sets of client options for the different groups of users. See "Creating Client Option Sets on the Server" on page 340 for details.
- 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. See "Creating Client Option Sets on the Server" on page 340 for details.
  - 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. See also "The Include-Exclude List" on page 368.
- Automate incremental backup procedures by defining schedules for each policy domain. Then associate schedules with client nodes in each policy domain. For information on schedules, see Chapter 15, "Scheduling Operations for Client Nodes," on page 433.

Changing Policy

Some types of clients and situations require policy changes. For example, if you need to direct client data to storage pools different from the default storage pools, you need to change policy. Other situations may also require policy changes. See "Policy Configuration Scenarios" on page 399 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. Do the following:

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. For details, see "Defining and Updating a Management Class" on page 380, "Defining and Updating a Backup Copy Group" on page 381, and "Defining and Updating an Archive Copy Group" on page 387.
3. Validate the policy set. See "Validating a Policy Set" on page 389 for details.
4. Activate the policy set. The contents of your new policy set becomes the ACTIVE policy set. See "Activating a Policy Set" on page 390 for details.

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 (client software at Version 4.2 and later)
- 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

Notes:

1. A base file is not eligible for expiration until all of its dependent subfiles have been expired. For details, see "Expiration Processing of Base Files and Subfiles" on page 420.
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. See "Deletion Hold" on page 394 for more information.

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. See "Reclaiming Space in Sequential Access Storage Pools" on page 267 and "Running Expiration Processing to Delete Expired Files" on page 391 for more information.

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. See "Managing Client Restartable Restore Sessions" on page 346 for information about restartable restore sessions.

Client Operations Controlled by Policy

IBM Tivoli Storage Manager policies govern the following client operations, which are discussed in this section:

- "Backup and Restore" on page 362
- "Archive and Retrieve"
- "Client Migration and Recall" on page 363

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, see "Policy for Logical Volume Backups" on page 400.

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. See "Setting Policy to Enable Point-in-Time Restore for Clients" on page 404 for details on the requirements.

Archive and Retrieve

To preserve files for later use or for records retention, a user with a backup-archive client can archive files, subdirectories, and directories on media controlled by the server. When users archive files, they can choose to have the backup-archive client erase the original files from their workstation after the client archives the files.

When a user retrieves a file, the server sends a copy of the file to the client node. The archived file remains in server storage.

Client Migration and Recall

When the Tivoli Storage Manager for Space Management product is on the workstation, a user can migrate files from workstation storage to server storage and recall those files as needed. Tivoli Storage Manager for Space Management frees space for new data and makes more efficient use of your storage resources. The installed Tivoli Storage Manager for Space Management product is also called the space manager client or the HSM client.

Files that are migrated and recalled with the HSM client are called *space-managed* files.

For details about using Tivoli Storage Manager for Space Management, see *IBM Tivoli Storage Manager for Space Management for UNIX: User's Guide*.

Migration

When a file is migrated to the server, it is replaced on the client node with a small stub file of the same name as the original file. The stub file contains data needed to locate the migrated file on server storage.

Tivoli Storage Manager for Space Management provides selective and automatic migration. Selective migration lets users migrate files by name. The two types of automatic migration are:

Threshold
: If space usage exceeds a high threshold set at the client node, migration begins and continues until usage drops to the low threshold also set at the client node.

Demand
: If an out-of-space condition occurs for a client node, migration begins and continues until usage drops to the low threshold.

To prepare for efficient automatic migration, Tivoli Storage Manager for Space Management copies a percentage of user files from the client node to the IBM Tivoli Storage Manager server. The *premigration* process occurs whenever Tivoli Storage Manager for Space Management completes an automatic migration. The next time free space is needed at the client node, the files that have been premigrated 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.

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 54 shows the parts of a policy and the relationships among the parts. You may refer to "Example: Sample Policy Objects" on page 377.

Figure 54. 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:

- 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, see "Running Expiration Processing to Delete Expired Files" on page 391 for details

Archive copy group
Controls the archive processing of files associated with the management class. An archive copy group determines the following:

- 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, see "Running Expiration Processing to Delete Expired Files" on page 391 for details

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.

See "More on Management Classes" on page 367 for details.

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

Relationships among Clients, Storage, and Policy

Figure 55 on page 366 summarizes the relationships among the physical device environment, IBM Tivoli Storage Manager storage and policy objects, and clients. The numbers in the following list correspond to the numbers in the figure.

Figure 55. How Clients, Server Storage, and Policy Work Together

1 When clients are registered, they are associated with a policy domain. Within the policy domain are the policy set, management class, and copy groups.

2, **3**
When a client backs up, archives, or migrates a file, it is bound to a management class. A management class and the backup and archive copy groups within it specify where files are stored and how they are managed when they are backed up, archived, or migrated from the client.

4, **5**
Storage pools are the destinations for backed-up, archived, or

space-managed files. Copy groups specify storage pools for backed-up or archived files. Management classes specify storage pools for space-managed files.

Storage pools are mapped to device classes, which represent devices. The storage pool contains volumes of the type indicated by the associated device class. For example, a storage pool that is mapped to a device class with a device type of 8MM contains only 8mm tapes.

6 Files that are initially stored on disk storage pools can migrate to tape storage pools if the pools are set up in a storage hierarchy.

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. See "Registering Nodes with the Server" on page 310 for information on registering client nodes and assigning them to policy domains.

The following sections give you more information about management classes and how they work with other parts of IBM Tivoli Storage Manager:

- "Contents of a Management Class"
- "Default Management Classes" on page 368
- "The Include-Exclude List" on page 368
- "How Files and Directories Are Associated with a Management Class" on page 370

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, which 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. See "How Files and Directories Are Associated with a Management Class" on page 370.
- 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. See "How Files and Directories Are Associated with a Management Class" on page 370.
- 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 do the following:

- 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.

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 56 shows an example of an include-exclude list. The statements in this example list do the following:

- Excludes certain files or directories from backup, archive, and client migration operations

  Line 1 in Figure 56 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 56 means that the files in the /home/ssteiner directory are excluded. The include statement that follows on line 3, however, means that the /home/ssteiner/options.scr file is eligible for backup and client migration.
- Binds a file to a specific management class

  Line 4 in Figure 56 means that all files and subdirectories belonging to the /home/ssteiner/driver5 directory are managed by the policy defined in the MCENGBK2 management class.

```
exclude /.../core
exclude /home/ssteiner/*
include /home/ssteiner/options.scr
include /home/ssteiner/driver5/.../* mcengbk2
```

Figure 56. 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 /home/ssteiner directory. 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 information on defining client option sets and assigning a client option set to a client, see "Creating Client Option Sets on the Server" on page 340.

For detailed information on the include and exclude options, see the user's guide for the appropriate client.

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.

  Note: 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).
- 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:
  - 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.

  Note: 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.
- 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 at Version 4.2 or later 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 details.

Effects of Changing a Management Class

A file remains bound to a management class even if the attributes of the management class or its copy groups change. The following scenario illustrates this process:

1. A file named REPORT.TXT is bound to the default management class that contains a backup copy group specifying that up to three backup versions can be retained in server storage.
2. During the next week, three backup versions of REPORT.TXT are stored in server storage. The active and two inactive backup versions are bound to the default management class.
3. The administrator assigns a new default management class that contains a backup copy group specifying only up to two backup versions.
4. The administrator then activates the policy set, and the new default management class takes effect.
5. REPORT.TXT is backed up again, bringing the number of versions to four. The server determines that according to the new backup copy group only two versions are to be retained. Therefore, the server marks the two oldest versions for deletion (expired).
6. Expiration processing occurs (see "Running Expiration Processing to Delete Expired Files" on page 391 for details). 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.

Rebinding Files to Management Classes

Rebinding is the process of associating a file or a logical volume image with a new management class.

Backup Versions: The server rebinds backup versions of files and logical volume images in the following cases:

- The user changes the management class specified in the include-exclude list and does a backup.
- An administrator activates a policy set in the same policy domain as the client node, and the policy set does not contain a management class with the same name as the management class to which a file is currently bound.
- An administrator assigns a client node to a different policy domain, and the active policy set in that policy domain does not have a management class with the same name.

Backup versions of a directory can be rebound when the user specifies a different management class using the DIRMC option in the client option file, and when the directory gets backed up.

If a file is bound to a management class that no longer exists, the server uses the default management class to manage the backup versions. When the user does another backup, the server rebinds the file and any backup versions to the default management class. If the default management class does not have a backup copy group, the server uses the backup retention grace period specified for the policy domain.

Archive Copies: Archive copies are never rebound because each archive operation creates a different archive copy. Archive copies remain bound to the management class name specified when the user archived them.

If the management class to which an archive copy is bound no longer exists or no longer contains an archive copy group, the server uses the default management class. If you later change or replace the default management class, the server uses the updated default management class to manage the archive copy.

If the default management class does not contain an archive copy group, the server uses the archive retention grace period specified for the policy domain.

How IBM Tivoli Storage Manager Selects Files for Policy Operations

This section describes how IBM Tivoli Storage Manager selects files for the following operations:

- Full and partial incremental backups
- Selective backup
- Logical volume backup
- Archive
- 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 should 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. See "Setting Policy to Enable Point-in-Time Restore for Clients" on page 404 for more information.

Full Incremental Backup

When a user requests a full incremental backup, IBM Tivoli Storage Manager performs the following steps to determine eligibility:

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.

   Note: 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, IBM Tivoli Storage Manager performs the following steps to determine eligibility:

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, IBM Tivoli Storage Manager performs the following steps to determine eligibility:

1. Checks the file against any include or exclude statements contained in the user include-exclude list:
   - Files that are not excluded are eligible for backup. If a management class is specified with the INCLUDE option, IBM Tivoli Storage Manager uses that management class.
   - If no include-exclude list exists, the files selected are eligible for backup, and IBM Tivoli Storage Manager uses the default management class.
2. Checks the management class of each included file:
   - If the management class contains a backup copy group and the serialization requirement is met, the file is backed up. Serialization specifies how files are handled if they are modified while being backed up and what happens if modification occurs.
   - If the management class does not contain a backup copy group, the file is not eligible for backup.

An important characteristic of selective backup is that a file is backed up without regard for whether the file has changed. This result may not always be what you want. For example, suppose a management class specifies to keep three backup versions of a file. If the client uses incremental backup, the file is backed up only when it changes, and the three versions in storage will be at different levels. If the client uses selective backup, the file is backed up regardless of whether it has changed. If the client uses selective backup on the file three times without changing the file, the three versions of the file in server storage are identical. Earlier, different versions are lost.

Logical Volume Backup

When a user requests a logical volume backup, IBM Tivoli Storage Manager performs the following steps to determine eligibility:

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, IBM Tivoli Storage Manager performs the following steps to determine eligibility:

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.

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. For details on how to generate backup sets see "Creating and Using Client Backup Sets" on page 412.

Automatic Migration from a Client Node

A file is eligible for automatic migration from an HSM client if it meets all of the following criteria:

- It resides on a node on which the root user has added and activated hierarchical storage management. It must also reside in a local file system to which the root user has added space management, and not in the root (/) or /tmp file system.
- It is not excluded from migration in the include-exclude list.
- It meets management class requirements for migration:
  - The file is not a character special file, a block special file, a FIFO special file (that is, a named pipe file) or a directory.
  - The file is assigned to a management class that calls for space management.

- The management class calls for automatic migration after a specified number of days, and that time has elapsed.
- A backup version of the file exists if the management class requires it.
- The file is larger than the stub file that would replace it (plus one byte) or the file system block size, whichever is larger.

How Client Migration Works with Backup and Archive

As an administrator, you can define a management class that specifies automatic migration from the client under certain conditions. For example, if the file has not been accessed for at least 30 days and a backup version exists, the file is migrated. You can also define a management class that allows users to selectively migrate whether or not a backup version exists. Users can also choose to archive files that have been migrated. IBM Tivoli Storage Manager does the following:

- If the file is backed up or archived to the server to which it was migrated, the server copies the file from the migration storage pool to the backup or archive storage pool. For a tape-to-tape operation, each storage pool must have a tape drive.
- If the file is backed up or archived to a different server, IBM Tivoli Storage Manager accesses the file by using the migrate-on-close recall mode. The file resides on the client node only until the server stores the backup version or the archived copy in a storage pool.

When a client restores a backup version of a migrated file, the server deletes the migrated copy of the file from server storage the next time reconciliation is run.

When a client archives a file that is migrated and does not specify that the file is to be erased after it is archived, the migrated copy of the file remains in server storage. When a client archives a file that is migrated and specifies that the file is to be erased, the server deletes the migrated file from server storage the next time reconciliation is run.

IBM Tivoli Storage Manager's default management class specifies that a backup version of a file must exist before the file is eligible for migration.

Creating Your Own Policies

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

Task	Required Privilege Class
Assign a default management class to a nonactive policy set in any policy domain	System or unrestricted policy
Assign a default management class to a nonactive policy set in policy domains over which you have authority	Restricted policy
Validate and activate policy sets in any policy domain	System or unrestricted policy
Validate and activate policy sets in policy domains over which you have authority	Restricted policy
Start inventory expiration processing	System

You can create your own policies in one of two ways:

- Define the parts of a policy and specify each attribute
- Copy existing policy parts and update only those attributes that you want to change

The following table shows that an advantage of copying existing policy parts is that some associated parts are copied in a single operation.

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

Figure 57 on page 378 shows the policies for an engineering department. This example is used throughout the rest of this chapter.

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 57. An Example of Policy Objects Defined for an Engineering Department

The sections that follow describe the tasks involved in creating new policies for your installation. Do the tasks in the following order:

Tasks:
"Defining and Updating a Policy Domain"
"Defining and Updating a Policy Set" on page 379
"Defining and Updating a Management Class" on page 380
"Defining and Updating a Backup Copy Group" on page 381
"Defining and Updating an Archive Copy Group" on page 387
"Assigning a Default Management Class" on page 389
"Activating a Policy Set" on page 390
"Running Expiration Processing to Delete Expired Files" on page 391.

Defining and Updating a Policy Domain

When you update or define a policy domain, you specify:

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 occurs:
- 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.

Example: Defining a Policy Domain

To create a new policy domain you can do one of the following:
- Copy an existing policy domain and update the new domain
- Define a new policy domain from scratch

Note: When you copy an existing domain, you also copy any associated policy sets, management classes, and copy groups.

For example, to copy and update, follow this procedure:
1. Copy the STANDARD policy domain to the ENGPOLDOM policy domain by entering:
   ```
   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 2 years by entering:
   ```
   update domain engpoldom description='Engineering Policy Domain'
   backretention=90 archretention=730
   ```

Defining and Updating a Policy Set

When you define or update a policy set, specify:

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. See "Activating a Policy Set" on page 390.

Example: Defining a Policy Set

An administrator needs to develop new policies based on the existing STANDARD policy set. To create the TEST policy set in the ENGPOLDOM policy domain, the administrator performs the following steps:

1. Copy the STANDARD policy set and name the new policy set TEST:

   ```
   copy policyset engpoldom standard test
   ```

   Note: When you copy an existing policy set, you also copy any associated management classes and copy groups.

2. Update the description of the policy set named TEST:

   ```
   update policyset engpoldom test
   description='Policy set for testing'
   ```

Defining and Updating a Management Class

When you define or update a management class, specify:

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 Tivoli Storage Manager for Space Management clients (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.

Note: You cannot specify a copy storage pool as a destination.

Example: Define a New Management Class

Create a new management class by following these steps:

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

Tasks:
"Where to Store Backed-Up Files"
"How to Handle Files That Are Modified During Backup"
"How Frequently Files Can Be Backed Up" on page 382
"How Many Backup Versions to Retain and For How Long" on page 383

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:

- 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.

Note: You cannot specify a copy storage pool.

How to Handle Files That Are Modified During Backup

You can use the SERIALIZATION attribute on the DEFINE COPYGROUP command to specify how files are handled 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 handle 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 usuable. 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.

How Frequently Files Can Be Backed Up

You can specify how frequently files can be backed up with two parameters:

Frequency

The frequency is the minimum number of days that must elapse between full incremental backups.

Note: For Windows this attribute is ignored during a journal-based backup.

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.

How Many Backup Versions to Retain and For How Long

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.

Note: A base file is not eligible for expiration until all its dependent subfiles have been expired. For details, see "Enabling Clients to Use Subfile Backup" on page 418.

For example, see Table 28 and Figure 58. 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 29 on page 385 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 28. 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 58. Active and Inactive Versions of REPORT.TXT

Table 29. 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. See “Setting Policy to Enable Point-in-Time Restore for Clients” on page 404 and “Policy for Logical Volume Backups” on page 400 for more information.

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. See “Setting Policy to Enable Point-in-Time Restore for Clients” on page 404 and “Policy for Logical Volume Backups” on page 400 for more information.

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. See “Setting Policy to Enable Point-in-Time Restore for Clients” on page 404 for more information.

Retain Only Version

If NOLIMIT is specified, the last version is retained forever unless a user or administrator deletes the file from server storage.

Example: Define a Backup Copy Group

Define a backup copy group belonging to the MCENG management class in the STANDARD policy set belonging to the ENGPOLDOM policy domain. This new copy group must do the following:

- 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`)

To define the backup copy group, enter:

```
define copygroup engpoldom standard mceng standard
destination=engback1 serialization=static
verexists=5 verdeleted=4 retextra=90 retonly=600
```

Defining and Updating an Archive Copy Group

To define or update an archive copy group, specify:

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. See "Using Virtual Volumes to Store Data on Another Server" on page 601 for information about registering servers as clients to another server.

Note: You cannot specify a copy storage pool as a destination.

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.

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.

When to initiate retention time

Specifies when the retention time noted in the RETVER parameter is initiated. Using the RETINIT parameter in the assigned archive copy group, you can initiate retention time by specifying either the storage of an object or another event.

There can be only be one event which begins retention for each object. For example, a business action such as the closure of a bank account is considered an event.

The server ignores a notification for an object for which retention has already been initiated either at the time the object was created or via a previous event notification. To notify the server of an event which initiates retention of an object, the client must be the owner of the object.

The minimum amount of days to retain an object

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. The value of this parameter is not displayed if you specified CREATION in RETINIT.

After you have defined an archive copy group using the RETINIT=EVENT and RETMIN=n parameters, 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.

It is possible to place a deletion hold on an object that was stored with RETINIT=EVENT for which the event has not been signalled. If the event is signalled while the deletion hold is in effect, the retention period will be initiated but the object will not be deleted. 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. See "Deletion Hold" on page 394 for more information about deletion holds.

Example: Define an Archive Copy Group

Define an archive copy group belonging to the MCENG class that:

- Allows users to archive a file if it is not in use (`serialization=static`)
- Retains the archive copy for 730 days (`retver=730`)
- Stores 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. See "Default Management Classes" on page 368 for suggestions about the content of default management classes.

Example: Assign a Default Management Class

To assign the STANDARD management class as the default management class for the TEST policy set in the ENGPOLDOM policy domain, enter:

```
assign defmgmtclass engpoldom standard standard
```

The STANDARD management class was copied from the STANDARD policy set to the TEST policy set (see "Example: Defining a Policy Set" on page 380). Before the new default management class takes effect, you must activate the policy set.

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.	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. See "How Files and Directories Are Associated with a Management Class" on page 370 for details. 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. See "Defining and Updating a Policy Domain" on page 378 for details.
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. See "Defining and Updating a Policy Domain" on page 378
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 HSM users.

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
- 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. See "Changing Policy" on page 360 for details.

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.

See "Data Retention Protection" on page 393 for more details about data retention protection.

Example: Validating and Activating a Policy Set

Validating and activating the STANDARD policy set in the ENGPOLDOM policy domain is a two-step process:

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.

For example, to assign the client node APPCLIENT1 to the ENGPOLDOM policy domain, enter the following command:

```
update node appclient1 domain=engpoldom
```

To create a new client node, NEWUSER, and assign it to the ENGPOLDOM policy domain, enter 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.

Notes:

1. A base file is not eligible for expiration until all of its dependent subfiles have been expired. For details, see "Expiration Processing of Base Files and Subfiles" on page 420.
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. See "Deletion Hold" on page 394 for more information.

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 you start the server. After that, the server runs expiration processing at the interval you specified with the option, measured from the start time of the server.

Using Commands and Scheduling to Control Expiration Processing

You can manually start expiration processing by issuing the following command:

```
expire inventory
```

Expiration processing then deletes expired files from the database. You can schedule this command by using the DEFINE SCHEDULE command. If you schedule the EXPIRE INVENTORY command, set the expiration interval to 0 (zero) in the server options so that the server does not run expiration processing when you start the server.

You can control how long the expiration process runs by using the DURATION parameter with the EXPIRE INVENTORY command.

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 EXPQUIET server option, or by using the QUIET=YES parameter with the EXPIRE INVENTORY command.. 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 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 Backup Volumes Onsite" on page 702 for more information.

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 will not be deleted from the Tivoli Storage Manager server until policy-based retention requirements for that object have been satisfied. It is enforced at the server level via the SET ARCHIVERETENTIONPROTECTION command. See the *Administrator's Reference* for more information.

Retention protection can only be activated on a new server that does not already have stored objects (backup, archive, or space-managed). Activating retention protection applies to all archive objects subsequently stored on that server. After retention protection has been set, the server cannot store backup objects, space-managed objects, or backupsets. Retention protection cannot be added for an object that was previously stored on a Tivoli Storage Manager server. Once an object is stored with retention protection, retention protection cannot be removed.

Retention protection is based on the retention criterion for each object, which is determined by the RETVER parameter of the archive copy group of the management class to which the object is bound. If an object uses event-based retention, the object will not expire until whatever comes later: either the date the object was archived plus the number of days in the RETMIN parameter or the date the event was signalled plus the number of days specified in the RETVER parameter. On servers which have retention protection enabled, the following operations will not delete objects whose retention criterion has not been satisfied:

- Requests from the client to delete an archive object
- DELETE FILESPACE (from either a client or administrative command)
- DELETE VOLUME DISCARDDATA=YES
- AUDIT VOLUME FIX=YES

Note: A cached copy of data can be deleted, but data in primary and copy storage pools can only be marked damaged and are never deleted.

If your server has data retention protection activated, the following 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 once it is imported.
- You cannot import data to a retention protected server.

For servers which have data retention protection enabled, consider using mirroring for the Tivoli Storage Manager database and log. You should also consider using roll forward log mode and performing frequent database backups. This way if a database restore is needed, the database can be brought back to the current point in time without loss of data. See Chapter 23, "Protecting and Recovering Your Server," on page 639 for more information.

Deletion Hold

If a hold is placed on an object via the client API, the object will not be 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, but the hold status will not 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 and copy storage pools can only be marked damaged and are never deleted.

Protecting Data Using the SnapLock Feature

The Network Appliance SnapLock licensed feature helps meet federal regulatory requirements for archived data. The SnapLock feature allows an application such as Tivoli Storage Manager to set a retention date for files and commit a file to a WORM (write once, read many) state. Data stored with a retention date cannot be deleted from the file system until the retention period has expired. The SnapLock feature can only be used by Tivoli Storage Manager servers which have data retention protection enabled. See "Data Retention Protection" on page 393 for more on data retention protection.

Data archived in data retention protection servers and stored to Network Appliance 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 Tivoli Storage Manager FILE volume, via the SnapLock interface, to a date in the future. This date is calculated by using the retention parameters (RETVER and RETMIN) 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 Tivoli Storage Manager FILE volumes are referred to as WORM FILE

volumes. Once a retention date has been set, the WORM FILE volume can not be deleted until the retention date has passed. IBM System Storage Archive Manager combined with WORM FILE volume reclamation ensures protection for the life of the data.

Storage pools can be managed in two ways: either by threshold or based on data retention period. The storage pool parameter, RECLAMATIONTYPE, 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
```

This indicates that the storage pools are managed by threshold.

Tivoli Storage Manager servers which have data retention protection enabled via the IBM System Storage Archive Manager product, and have access to a Network Appliance 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 will be managed by a retention date. When a Tivoli Storage Manager 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 SnapLock management section in the Network Appliance *Data ONTAP Storage Management Guide* for details on the SnapLock filer. Note this is Network Appliance documentation.

Attention: It is not recommended that you use this feature to protect data with a retention period of less than three months.

Reclamation and the SnapLock Feature

It is recommended that the Network Appliance default retention period be set to 30 days to match the WORM FILE default reclamation period. Tivoli Storage Manager will reclaim any remaining data on a WORM FILE volume just prior to the retention date expiration. The reclamation of a WORM FILE volume to another WORM FILE volume prior to the retention date expiration ensures that data is always protected by the SnapLock feature. Since this protection is at a Tivoli Storage Manager volume level, the data on the volumes is free to 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 also the retention period stored with the physical file on the SnapLock volume. Should a Tivoli Storage Manager administrator issue a command in an attempt to delete the data, Tivoli Storage Manager would fail the command. Or, should someone attempt to delete the file through a series of network file system calls, the SnapLock feature would prevent 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. In some cases, the retention of individual files may exceed the retention time for the WORM FILE volume they were stored on, necessitating transfer to

another volume to ensure that the files are stored on WORM media. Some objects on the volume may need to be retained longer than other objects on the volume because:

- 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 may be 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 Tivoli Storage Manager 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, it cannot be updated to THRESHOLD. During the definition of the SnapLock Storage pool, a check is made to ensure that the directories specified in the device class are actually 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 will fail. If a device class is updated to contain additional directories and there are SnapLock storage pools assigned to it, the same check will be made to ensure all directories are SnapLock WORM volumes. See the Administrator's Reference for a detailed description of the DEFINE STGPOOL command.

There are three retention periods available in the Network Appliance 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 Network Appliance SnapLock volumes based on the values in the copy group for the data being archived. The Network Appliance filer should not conflict with the ability of the Tivoli Storage Manager server to set the retention period. The following are the Tivoli Storage Manager recommendations for retention periods in the Network Appliance 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 Network Appliance SnapLock filer for WORM FILE storage) for the data retention period. The copy group is the one in use storing data on Network Appliance 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 Network Appliance SnapLock volumes will be largely defeated in that no volume will be able to be reused for thirty years.

With the Network Appliance 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 Network Appliance filer for that volume is set to the END RECLAIM PERIOD date. This means the Network Appliance 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 Network Appliance 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 once 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.

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 Network Appliance WORM volumes. The Tivoli Storage Manager server allows this type of movement, but if data is moved from a WORM FILE volume to another type of media, the data may no longer be protected from inadvertent or malicious deletion. If this data is on WORM volumes to meet data retention and protection requirements for certain legal purposes and is moved to other media, the data may no longer meet those requirements. You should configure your storage pools so this type of data is kept in storage pools which consist of SnapLock WORM volumes during the entire data retention period.

Set up SnapLock Volumes as Tivoli Storage Manager WORM FILE Volumes

When defining or updating configurations that involve SnapLock storage pools, you should ensure that the storage pools selected for the NEXTSTGPOOL, RECLAIMSTGPOOL, and COPYSTGPOOLS parameters have the RECLAMATIONTYPE=SNAPLOCK option specified. By configuring the storage pools in this way, you ensure that your data will be properly protected. If you define a next, reclaim, or copy storage pool without selecting the RECLAMATIONTYPE=SNAPLOCK option, you will not have a protected storage pool. The command will succeed but a warning message will be issued.

Perform the following steps to set up a SnapLock volume for use as a Tivoli Storage Manager WORM FILE volume.

1. Install and set up SnapLock on the Network Appliance filer. See Network Appliance documentation for more information.

2. Properly configure the minimum, maximum, and default retention periods as advised in "Retention Periods" on page 395. 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 Application Programming Interface to archive your objects into the SnapLock storage pool. This feature is not available on standard Tivoli Storage Manager backup-archive clients.

Policy Configuration Scenarios

This section includes recommendations for some cases for which policy changes may be needed.

- "Configuring Policy for Direct-to-Tape Backups"
- "Configuring Policy for Tivoli Storage Manager Application Clients" on page 400
- "Policy for Logical Volume Backups" on page 400
- "Configuring Policy for NDMP Operations" on page 401
- "Configuring Policy for LAN-free Data Movement" on page 402
- "Policy for IBM Tivoli Storage Manager Servers as Clients" on page 404
- "Setting Policy to Enable Point-in-Time Restore for Clients" on page 404

Configuring Policy for Direct-to-Tape Backups

The server default policy enables client nodes to back up data to disk storage pools on the server. As an alternative, you may configure a policy to store client data directly in tape storage pools to reduce contention for disk resources. If you back up directly to tape, the number of clients that can back up data at the same time is equal to the number of drives available to the storage pool (through the mount limit of the device class). For example, if you have one drive, only one client at a time can back up data.

The direct-to-tape backup eliminates the need to migrate data from disk to tape. However, performance of tape drives is often lower when backing up directly to tape than when backing up to disk and then migrating to tape. Backing up data directly to tape usually means more starting and stopping of the tape drive. Backing up to disk then migrating to tape usually means the tape drive moves more continuously, meaning better performance.

At the server command line, you may define a new policy domain that enables client nodes to back up or archive data directly to tape storage pools. For example, you may define a policy domain named DIR2TAPE with the following steps:

1. Copy the default policy domain STANDARD as a template:

   ```
   copy domain standard dir2tape
   ```

   This command creates the DIR2TAPE policy domain that contains a default policy set, management class, backup and archive copy group, each named STANDARD.
2. Update the backup or archive copy group in the DIR2TAPE policy domain to specify the destination to be a tape storage pool. For example, to use a tape storage pool named TAPEPOOL for backup, enter the following command:

   ```
   update copygroup dir2tape standard standard destination=tapepool
   ```

   To use a tape storage pool named TAPEPOOL for archive, enter 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, enter the following command:

   ```
   update node tapeuser1 domain=dir2tape
   ```

Configuring Policy for Tivoli Storage Manager Application Clients

The Tivoli Storage Manager application clients using the server to store data may require that you configure policy to make the most efficient use of server storage. See the user's guide for each application client for policy requirements.

Some of the application clients include a time stamp in each database backup. Because the default policy for the server keeps one backup version of each unique file, database backups managed by default policy are never deleted because each backup is uniquely named with its time stamp. To ensure that the server deletes backups as required, configure policy as recommended in the user's guide for the application client.

Policy for Logical Volume Backups

Consider defining a management class specifically for logical volume backups. To enable clients to restore a logical volume and then reconcile the results of any file backup operations since the logical volume backup was made, you must set up management classes with the backup copy group set up differently from the STANDARD. The Versions Data Exists, Versions Data Deleted, and Retain Extra Versions parameters work together to determine over what time period a client can restore a logical volume image and reconcile later file backups. Also, you may have server storage constraints that require you to control the number of backup versions allowed for logical volumes. The server handles logical volume backups the same as regular incremental or selective backups. Logical volume backups differ from selective, incremental, or archive operations in that each file space that is backed up is treated as a single large file.

Backups of logical volumes are intended to help speed the restoration of a machine. 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 30 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 30. 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 NAS file server as a node, using 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 Network Appliance 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. See Chapter 7, "Using NDMP for Operations with NAS File Servers," on page 143 for details.

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. For more information about client option sets see "Creating Client Option Sets on the Server" on page 340.

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.

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 machine. The storage agent transfers data between the client and the storage device. See *IBM Tivoli Storage Manager Storage Agent User's Guide* for details. See the Web site for details on clients that support the feature: www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

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. ("Configuring IBM Tivoli Storage Manager for LAN-free Data Movement" on page 135 describes how to configure the devices and define the storage pool.) 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:

- "Define a New Policy Domain" on page 403
- "Define a New Management Class in an Existing Policy Domain" on page 404

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. Do the following:

1. Create the policy domain for the clients. For example, to define a policy domain that is named SANCLIENTS, enter 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, enter 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, enter 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. See "Configuring IBM Tivoli Storage Manager for LAN-free Data Movement" on page 135 for details.
   - Accept the default settings for all remaining parameters.

   For example, to define the backup copy group for the SANCLIENTMC management class, enter 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. See "Configuring IBM Tivoli Storage Manager for LAN-free Data Movement" on page 135 for details.
   - Accept the default settings for all remaining parameters.

   For example, to define the archive copy group for the SANCLIENTMC management class, enter 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, enter 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, enter the following command:

   ```
   update node sanclient1 domain=sanclients
   ```

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 IBM 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. See "Using Virtual Volumes to Store Data on Another Server" on page 601 for more information.

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.

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.

See "Setting Up an Enterprise Configuration" on page 575 for details.

Querying Policy

Task	Required Privilege Class
Query any policy domain, policy set, management class, or copy group	Any administrator

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.

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

To request information about backup copy groups (the default) in the ENGPOLDOM engineering policy domain, enter:

```
query copygroup engpoldom * *
```

The following 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 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

To request information about management classes in the ENGPOLDOM engineering policy domain, enter:

```
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

To request information about policy sets in the ENGPOLDOM engineering policy domain, enter:

```
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

To request information about a policy domain (for example, to determine if any client nodes are registered to that policy domain), enter:

```
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. See "Protection and Expiration of Archive Data" on page 393 for limitations on deleting archive objects.

You cannot delete the ACTIVE policy set or objects that are part of that policy set.

Task	Required Privilege Class
Delete policy domains	System
Delete any policy sets, management classes, or copy groups	System or unrestricted policy
Delete policy sets, management classes, or copy groups that belong to policy domains over which you have authority	Restricted policy

You can delete the policy objects named STANDARD that come with the server. However, all STANDARD policy objects are restored whenever you reinstall the server. If you reinstall the server after you delete the STANDARD policy objects, the server issues messages during processing of a subsequent DSMSERV AUDITDB command. The messages may include the following statement: "An instance count does not agree with actual data." The DSMSERV AUDITDB command corrects this problem by restoring the STANDARD policy objects. If necessary, you can later delete the restored STANDARD policy objects.

Deleting Copy Groups

You can delete a backup or archive copy group if it does not belong to a management class in the ACTIVE policy set.

For example, to delete the backup and archive copy groups belonging to the MCENG and STANDARD management classes in the STANDARD policy set, enter:

```
delete copygroup engpoldom standard mceng type=backup
delete copygroup engpoldom standard standard type=backup
delete copygroup engpoldom standard mceng type=archive
delete copygroup engpoldom standard standard type=archive
```

Deleting Management Classes

You can delete a management class if it does not belong to the ACTIVE policy set.

For example, to delete the MCENG and STANDARD management classes from the STANDARD policy set, enter:

```
delete mgmtclass engpoldom standard mceng
delete mgmtclass engpoldom standard standard
```

When you delete a management class from a policy set, the server deletes the management class and all copy groups that belong to the management class in the specified policy domain.

Deleting Policy Sets

Authorized administrators can delete any policy set other than the ACTIVE policy set. For example, to delete the TEST policy set from the ENGPOLDOM policy domain, enter:

```
delete policyset engpoldom test
```

When you delete a policy set, the server deletes all management classes and copy groups that belong to the policy set within the specified policy domain.

The ACTIVE policy set in a policy domain cannot be deleted. You can replace the contents of the ACTIVE policy set by activating a different policy set. Otherwise, the only way to remove the ACTIVE policy set is to delete the policy domain that contains the policy set.

Deleting Policy Domains

You can delete a policy domain only if the domain has no client nodes registered to it. To determine if any client nodes are registered to a policy domain, issue the QUERY DOMAIN or the QUERY NODE command. Move any client nodes to another policy domain, or delete the nodes.

For example, to delete the STANDARD policy domain, perform the following steps:

1. Request a list of all client nodes assigned to the STANDARD policy domain by entering:

   ```
   query node * domain=standard
   ```

2. If client nodes are assigned to the policy domain, remove them in one of the following ways:
   - 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, as described in "How Files and Directories Are Associated with a Management Class" on page 370.
   - 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 entering:

   ```
   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.

Chapter 14. Managing Data for Client Nodes

This chapter contains information to help you generate backup sets and enable subfile backups for client nodes. You can also use data validation for client nodes so that any data corruption is identified when data is sent over the network between the client and server.

See the following sections for more information:

Tasks:
Tasks:
"Validating a Node's Data During a Client Session" on page 412
"Generating Client Backup Sets on the Server" on page 413
"Restoring Backup Sets from a Backup-Archive Client" on page 414
"Moving Backup Sets to Other Servers" on page 415
"Managing Client Backup Sets" on page 415
"Enabling Clients to Use Subfile Backup" on page 418
"Optimizing Restore Operations for Clients" on page 420
"Managing Performance for Archive Operations" on page 427
"Managing Storage Usage for Archives" on page 429
Concepts:
"Performance Considerations for Data Validation" on page 412
"Creating and Using Client Backup Sets" on page 412
"Archive Operations Overview" on page 426

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:

- Tivoli Storage Manager client nodes at Version 5.1 or higher.
- Tivoli Storage Manager storage agents at Version 5.1 or higher. For details, refer to the storage agent user's guide for your particular operating system.

© Copyright IBM Corp. 1993, 2005

This section provides information about data validation between a node and the server.

See “Choosing Where to Enable Data Validation” on page 673 to help you determine where to enable data validation.

Performance Considerations for Data Validation

Consider the impact on performance when you decide whether data validation is necessary for all nodes or some nodes. Data validation impacts performance because additional CPU overhead is required on both the client and server to calculate and compare CRC values.

This type of validation is independent from validating data written to a storage pool volume. See “Data Validation During Audit Volume Processing” on page 673.

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
```

Creating and Using Client Backup Sets

A *backup set* is a collection of backed-up data from one client, stored and managed as a single object on specific media in server storage. The server creates copies of active versions of a client’s backed up objects that are within the one or more file spaces specified with the GENERATE BACKUPSET command, and consolidates them onto sequential media. Currently, the backup object types supported for backup sets include directories and files only. The process is also called *instant archive*.

The media may be directly readable by a device such as a CD-ROM, JAZ, or ZIP drive attached to a client’s machine.

Administrators can generate multiple copies of backup sets that correspond to some point-in-time. The backup sets can be retained for various time periods. This is an efficient way to create long-term storage of periodic backups, without requiring the data to be sent over the network again.

While an administrator can generate a backup set from any client's backed up files, backup sets can only be used by a backup-archive client. You cannot generate a backup set for a NAS node.

See the following sections for details:

- "Generating Client Backup Sets on the Server"
- "Restoring Backup Sets from a Backup-Archive Client" on page 414
- "Moving Backup Sets to Other Servers" on page 415
- "Managing Client Backup Sets" on page 415

Generating Client Backup Sets on the Server

Task	Required Privilege Class
Generate a backup set	System or restricted policy over the domain to which the node is assigned

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.

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.

See the following sections:

- "Choosing Media for Generating the Backup Set"
- "Selecting a Name for the Backup Set" on page 414
- "Setting a Retention Period for the Backup Set" on page 414
- "Example: Generating a Client Backup Set" on page 414

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.

Consider the following 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 machines. 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 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. For more information, see "Configuring Removable File Devices" on page 129.

Using Scratch Media: 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.

Selecting a Name for the Backup Set

The server adds a unique suffix to the name you specify for the backup set. For example, if you name the backup set *mybackupset*, the server adds a unique extension, such as 3099, to the name. This allows you to create backup sets with the same name without overwriting previous backup sets.

To later display information about this backup set, you can include a wildcard character with the name, such as *mybackupset**, or you can specify the fully qualified name, such as *mybackupset.3099*.

Setting a Retention Period for the Backup Set

You can set the retention period, specified as a number of days, to retain the backup set on the server. You can specify a number between zero and 30,000 days. Backup sets are retained on the server for 365 days if you do not specify a value. The server uses the retention period to determine when to expire the volumes on which the backup set resides.

Example: Generating a Client Backup Set

Generate a backup set on CD-ROM that the client can later use to restore the data. Because CD-ROM is a read-only medium, you must create a backup set on a device that the server can write to before you create the backup set on the CD-ROM. Do not exceed the CD-ROM data capacity. Use the following steps to generate a backup set on a CD-ROM:

1. Define a device class whose device type is FILE. Name the device class CDFILE:

   ```
   define devclass cdfile devtype=file maxcapacity=640M dir=/backupset
   ```

2. Create a backup set with a device type of FILE for client node JOHNSON. Name the backup set PROJECT and retain it for 90 days.

   ```
   generate backupset johnson project devclass=file
   volumes=BK1,BK2,BK3 retention=90
   ```

   Note: 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 415.

Restoring Backup Sets from a Backup-Archive Client

Backup-archive client nodes can restore their backup sets in either of two ways:

- Directly from the server.
- Using a device attached to the client's machine that will read the media in which the backup set is stored.

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 *Using the Backup-Archive Client* guide for your particular operating system.

Moving Backup Sets to Other Servers

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.

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 a common device type. The level of the server defining the backup set must be equal to or greater than the level of the server that generated the backup set.

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.

Using the example described in "Example: Generating a Client Backup Set" on page 414, you can make the backup set that was copied to the CD-ROM available to another server by entering:

```
define backupset johnson project devclass=cdrom volumes=BK1,BK2,BK3
description="backup set copied to CD-ROM"
```

Notes:

1. Devclass=cdrom specifies a device class of type REMOVABLEFILE that points to your CD-ROM drive. CD-ROMs have a maximum capacity of 650MB.
2. Volumes=BK1,BK2,BK3 specifies the names of the volumes containing the backup set. The volume label of these CD-ROMs must match the name of the file on the volume exactly.

Managing Client Backup Sets

You can update, query, and delete backup sets.

Task	Required Privilege Class
Update the retention period assigned to a backup set	System or restricted policy over the domain to which the node is assigned
Display information about backup sets	Any administrator
Display information about backup set contents	System or restricted policy over the domain to which the node is assigned

Task	Required Privilege Class
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.

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. For example, to update the retention period assigned to backup set named ENGDATA.3099, belonging to client node JANE, to 120 days, enter:

```
update backupset jane engdata.3099 retention=120
```

Displaying Backup Set Information

To view information about backup sets, you can use the QUERY BACKUPSET command. The output that is displayed lists information such as the name of the client node whose data is contained in the backup set as well as the description of the backup set, assuming one has been used.

The following figure shows the report that is displayed after you enter:

```
query backupset
```

```
        Node Name: JANE
  Backup Set Name: MYBACKUPSET.3099
        Date/Time: 09/04/2002 16:17:47
 Retention Period: 60
Device Class Name: DCFILE
      Description:
```

Displaying Information about Backup Set Volumes

A client's backup set can reside on more than one volume. The server records the information about the volumes used for the backup set in the volume history file. Volume history includes information such as the date and time the backup set was generated, the device class to which the backup set was written, and the command used to generate the backup set. If a backup set spans several volumes, the server displays the command used to generate the backup set only with the first volume.

You can view this information when you use the QUERY VOLHISTORY command, with BACKUPSET specified as the volume type.

The following example shows how this particular client's backup set resides on three volumes, and the command used to generate the backup set is displayed with the first volume.

```
       Date/Time: 09/04/2002 07:34:06 PM
     Volume Type: BACKUPSET
   Backup Series:
Backup Operation:
      Volume Seq: 1
    Device Class: FILE
     Volume Name: 01334846.ost
 Volume Location:
         Command: gen backupset client57 testbs /home dev=file scratch=yes
                   ret=2 desc="Client57 backupset"

       Date/Time: 09/04/2002 07:34:06 PM
     Volume Type: BACKUPSET
   Backup Series:
Backup Operation:
      Volume Seq: 2
    Device Class: FILE
     Volume Name: 01334849.ost
 Volume Location:
         Command:


       Date/Time: 09/04/2002 07:34:06 PM
     Volume Type: BACKUPSET
   Backup Series:
Backup Operation:
      Volume Seq: 3
    Device Class: FILE
     Volume Name: 01334850.ost
 Volume Location:
         Command:
```

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. The following figure shows the report that is displayed after you enter:

```
query backupsetcontents jane engdata.3099
```

```
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                     ...        ......
```

How File Space and File Names May be Displayed: File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (??), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

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. 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.

To delete a backup set named ENGDATA.3099, belonging to client node JANE, created before 11:59 p.m. on March 18, 1999, enter:

```
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
```

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 419.

Example of 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 handles backups of file CUST.TXT.

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.

Setting Up Clients to Use Subfile Backup

To enable subfile backup, do the following:

- ***On the server:*** You must set up the server to allow clients to back up subfiles. Use the SET SUBFILE command.

  ```
  set subfile client
  ```

- ***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 340 for how to set up and use client option sets.

  See *IBM Tivoli Storage Manager for Windows: Backup-Archive Clients Installation and User's Guide* for more information about the options.

Managing Subfile Backups

The following sections describe how 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 391.

Adding Subfiles to Backup Sets

When a subfile is added to a backup set, Tivoli Storage Manager includes its corresponding base file with the backup set. If the base file and dependent subfiles are stored on separate volumes when a backup set is created, additional volume mounts may be required to create the backup set.

Deleting Base Files

If a base file is deleted as a result of processing a DELETE VOLUME command, the server recognizes its dependent subfiles and deletes them from the server as well. Subfiles without the corresponding base file are incomplete and useless to the user.

Optimizing Restore Operations for Clients

The progressive incremental backup that is the Tivoli Storage Manager standard results in operations that are optimized for the restore of individual files or small numbers of files. Progressive incremental backup minimizes tape usage, reduces network traffic during backup operations, and eliminates the storage and tracking of multiple copies of the same data. Progressive incremental backup may reduce the impact to client applications during backup. For a level of performance that is balanced across both backup and restore operations, the best method is usually using progressive incremental backup with collocation set on in the storage pool.

If restore performance is more important than a balance between backup and restore operations, you can optimize based on your goals for restore performance. When you optimize for restore, there are often costs in tape usage and backup performance. To balance the costs against the need for optimized restore operations, you should do the following:

Identify systems that are most critical to your business
Consider where your most important data resides, what is most critical to restore, and what needs the fastest restore. Identify which systems and applications you need to focus on optimizing for restore.

Identify your goals
Identify your goals and order the goals by priority. Some goals to consider are:

- Disaster recovery or recovery from hardware crashes, requiring file system restores
- Recovery from loss or deletion of individual files or groups of files
- Recovery for database applications
- Point-in-time recovery of groups of files

The importance of each goal can vary for the different client systems that you identified as being most critical.

For more background on restore operations for clients, see "Concepts for Client Restore Operations" on page 423.

Environment Considerations

Performance depends on the environment, including:

- Network characteristics
- Storage hardware, including the types of tape drives used and the availability of snapshot functions
- Time constraints for backup and restore operations

Consider using disk to store data that requires quick restoration. For data that is less critical, store the data to disk, then allow or force the data to migrate to tape later.

One of the most important variables in restore performance is the layout of the data that is to be restored across single or multiple tape volumes. Major causes of performance problems are excessive tape mounts and the need to skip over expired or inactive data on a tape. After a long series of incremental backups, perhaps over years, the active data for a single file space can be spread across many tape volumes. A single tape volume can have active data mixed with inactive and expired data. See the following sections, which discuss ways to control the placement of data, such as:

- Use collocation in storage pools.
- Limit the number of inactive versions of data through policy.
- Use the MOVE DATA or MOVE NODEDATA commands.

Restoring Entire File Systems

Using a file system image backup optimizes restore operations when an entire file system needs to be restored, such as in disaster recovery or recovery from a hardware failure. Restoring from an image backup minimizes concurrent mounts of tapes and positioning within a tape during the restore operation. Consider the following as aids to file system restore operations:

- Perform image backups frequently. More frequent image backups give better point-in-time granularity, but will cost in terms of tape usage, disruption to the client system during backup, and greater network bandwidth needed.

  A guideline is to perform an image backup when more than 20% of the data in the file system has changed.
- Combine image backups with progressive incremental backups for the file system. This allows for full restore to an arbitrary point-in-time.
- To minimize disruption to the client during backup, use either hardware-based or software-based snapshot techniques for the file system.

The capability for image backup is not available for all clients at this time. If image backup is not available for the client and full file system restore is a priority, consider using selective backup to force a full file system backup at regular intervals.

Restoring Parts of File Systems

Progressive incremental backups optimize restore operations for small numbers of files or groups of files. These backups also make optimal use of network bandwidth for backup operations, and may minimize elapsed backup time and tape usage. If you want to optimize for restoring a file or a group of files, or for a system on which an image backup cannot be made, consider the following additional methods:

- Use collocation by group, by a single client node, or by client file space for primary sequential pools that clients back up to. For large file spaces for which restore performance is critical, consider creating mount points on the client system. This would allow collocation of data below the file space level.

  See "Keeping a Client's Files Together: Collocation" on page 258 for more information about collocation.
- Specify the client option COLLOCATEBYFILESPEC. This can help limit the number of tapes written to by objects from one file specification. See the Administrator's Reference for more information about client options.
- Use the MOVE NODEDATA command to consolidate critical data in tape storage pools, even in storage pools that have collocation set on. It may be important to consolidate data for certain nodes, file spaces, and data types more often than for others. If you do not use collocation or are limited by tape quantity, you may want to do this more often. The rate of data turnover is also something to consider.

  Use the RECONSTRUCT parameter on the command to remove unused space in file aggregates when the aggregates are moved.

  Use the command for staging data to disk when the lead time for a restore request allows it.

  The effectiveness of the command in optimizing for restore might be reduced if a large number of versions are kept.
- Create backup sets that can be taken to the client system and used to restore from directly. This is effective if there is sufficient lead time prior to the restore, and can save network bandwidth.

  Creation of backup sets can also be done periodically when resources are available, for example on weekends.
- Use progressive incremental backups, but periodically force a full backup.

  Some users have found it effective to define multiple Tivoli Storage Manager client nodes on a system. One client node performs the incremental backups and uses policies which retain multiple versions. Another client node performs either full backups or incremental backups with collocation, but uses policies that retain a single version. One node can be used for restoring older versions of individual files, and the other client node can be used for restoring a complete file system or directory tree to the latest version.
- Create multiple storage pool hierarchies for clients with different priorities. For the most critical data, use of only disk storage might be the best choice. Using different storage hierarchies also allows you to set collocation differently in the different hierarchies.
- Minimize the number of versions you keep. This reduces the amount of time spent positioning a tape during a restore operation. An alternative would be to perform full backups.
- Consider storage media characteristics, for example, the type of tape drive you use. Use full file system backups if the tape drives you use are relatively slow at positioning operations.

Restoring Databases for Applications

Doing more frequent full backups leads to faster restores for databases. For some database products, you can use multiple sessions to restore, restore just the database, or restore just the logs for the database. Optimal techniques for specific Tivoli Storage Manager application clients are documented in "IBM Tivoli Storage Manager Publications" on page xiii.

Restoring Files to a Point in Time

If you need the ability to restore files to a point in time, consider setting policy to keep a large number of versions (by setting VEREXISTS=NOLIMIT and VERDELETED=NOLIMIT in the backup copy group). Keeping a large number of versions is not essential for restoring to a point in time, but by increasing the number of versions that are kept, it may be possible to restore from an earlier point in time and still find the versions corresponding to that time. If you also schedule incremental backups regularly, you will have greater granularity in restoring to a discrete point in time. However, keeping a large number of versions can degrade the performance of restore operations, as described in "Restoring Parts of File Systems" on page 421. Setting policy to keep a large number of versions also has costs in terms of database space and storage pool space. It may have overall performance implications.

If you cannot afford the resource costs of keeping the large numbers of file versions and need the ability to restore to a point in time, consider using backup sets, exporting the client data, or using archive. Using backup sets, exporting client data, or archiving files gives you the capability to restore to the point in time when the backup set was generated, the export was performed, or the archive was created. Keep in mind that when you need to restore the data, your selection is limited to the time at which you created the backup set, export, or archive.

Note: If you use the archive function, you should create an archive monthly or yearly. Archive should not be used as a primary backup method because frequent archives with large amounts of data can affect server performance.

Concepts for Client Restore Operations

The following sections contain background information on client restore operations:

"No Query Restore Processes"
"Backup and Restore Using Multiple Commands" on page 424
"Restore Using Multiple Sessions on Clients" on page 425
"Controlling Resource Utilization by a Client" on page 425
"Managing Server Storage to Optimize Restore Operations" on page 426

No Query Restore Processes

The client uses two different methods for restore operations: Standard restore (also called classic restore), and *no-query restore*.

Standard restore requires more interaction between the client and the server, and multiple processes cannot be used for the restore. The no-query restore requires less interaction between the client and the server, and the client can use multiple sessions for the restore process. The no-query restore process is useful when restoring large file systems on a client with limited memory because it avoids some processing that can affect the performance of other client applications. In addition, it can achieve a high degree of parallelism by restoring with multiple sessions from the server and storage agent simultaneously. The no-query restore operation, however, can take much longer to complete than the standard restore operation. For example, in cases where only specific directories or files within a directory are to be restored, it may be faster to use a classic restore.

The method is called no-query restore because the client sends a single restore request to the server instead of querying the server for each object to be restored. The server returns the files and directories to the client without further action by

the client. The client accepts the data coming from the server and restores it to the destination named on the restore command.

The no-query restore process is used by the client only when the restore request meets both of the following criteria:

- You enter the restore command with a source file specification that has an unrestricted wildcard.

  An example of a source file specification with an unrestricted wildcard is:

  `/home/mydocs/2002/*`

  An example of a source file specification with a restricted wildcard is:

  `/home/mydocs/2002/sales.*`
- You do *not* specify any of the following client options:

  inactive
  latest
  pick
  fromdate
  todate

To force the use of classic restore, use ?* in the source file specification rather than *. For example:

`/home/mydocs/2002/?*`

For more information about restore processes, see *Backup-Archive Clients Installation and User's Guide*.

Backup and Restore Using Multiple Commands

Another method which can aid in both the backup and restore of client nodes with critical data is to manage the backup process through multiple commands instead of multiple sessions. For example, when using multi-session backup, multiple backup sessions may be contending for the same underlying hard disk, thus causing delays. An alternative is to manage this process externally by starting multiple **dsmc** commands. Each command backs up a pre-determined number of file systems. Using this method in conjunction with collocation at the file space level can improve backup throughput and allow for parallel restore processes across the same hard drives.

You must issue multiple commands when you are restoring more than one file space. For example, when you are restoring a c: drive and a d: drive on a Windows system you must issue multiple commands.

Consider using multiple commands when you are restoring a single, large file space, and all of the following conditions are true:

- The data was backed up to a storage pool that had collocation set to FILESPACE. Files will be on multiple volumes, and the volumes can be mounted by multiple processes.
- The files are approximately evenly distributed across the different top-level directories in the file space.
- The number of top-level directories in the file space is not large.
- You can issue commands for the different top-level directories, and the commands do not overlap (so that the same file is not restored multiple times by different commands).

Issue multiple commands either by issuing the commands one after another in a single session or window, or by issuing commands at the same time from different command windows.

When you enter multiple commands to restore files from a single file space, you must specify a unique part of the file space in each restore command. Be sure that you do not use any overlapping file specifications in the commands. To display a list of the directories in a file space, use the query backup command on the client. For example:

```
dsmc query backup -dirsonly -subdir=no /usr/
```

For more information, see *Backup-Archive Clients Installation and User's Guide*.

Restore Using Multiple Sessions on Clients

To use multiple sessions, data for the client must be on multiple, sequential access volumes, or a combination of sequential access volumes and disk. The data for a client usually becomes spread out over some number of volumes over time. This occurs deliberately when collocation is not used for the storage pool where the client data is stored.

To potentially benefit from multi-session restores, consider collocating client data by group. Collocation by group can cause data for a node to be distributed on more than one volume (while keeping the group's total data on as few volumes as possible).

Restore operations themselves are not restricted on mount points. The MAXNUMMP parameter in the REGISTER NODE or UPDATE NODE command applies to restore in that it can restrict other sessions trying to backup or archive data for the same node.

Set the client option for resource utilization to one greater than the number of desired sessions (use the number of drives that you want that single client to use). See "Controlling Resource Utilization by a Client." The client option can be included in a client option set.

Issue the restore command so that it results in a no query restore process. See "No Query Restore Processes" on page 423 for details.

Controlling Resource Utilization by a Client

By setting the MAXNUMMP parameter on either the UPDATE NODE or REGISTER NODE command, you can control the number of mount points (equivalent to drives) allowed to a client.

At the client, the option for resource utilization also has an effect on how many drives (how many sessions) the client can use. The client option, resource utilization, can be included in a client option set. If the number specified in the MAXNUMMP parameter is too low and there are not enough mount points for each of the sessions, it may not be possible to achieve the benefits of the multiple sessions specified in the resource utilization client option.

- For backup operations, you might want to prevent multiple sessions if the client is backing up directly to tape, so that data is not spread among multiple volumes. Multiple sessions can be prevented at the client by using a value of 2 for the resource utilization option on the client.

- For restore operations, set the resource utilization option to one greater than the number of desired sessions. Use the number of drives that you want that single client to use.

Managing Server Storage to Optimize Restore Operations

Because how data is arranged on tapes can affect restore performance, the administrator can take actions in managing server storage to help optimize restore operations.

- Ensure that tape reclamation and expiration are run regularly so that the tape drive will not have as much expired data to skip over during restore. See "Reclaiming Space in Sequential Access Storage Pools" on page 267 and "File Expiration and Expiration Processing" on page 361.
- Reduce the number of file versions that are retained so that the tape drive will not have to skip over as much inactive data during restore. See "How Many Backup Versions to Retain and For How Long" on page 383.

Managing Archive Data

Managing archive data on the server becomes important when you have client nodes that archive large numbers (hundreds or thousands) of files every day. If you archive files using automated tools that invoke the command line client or API, there is a good chance you will encounter such extremely large numbers. If you have noticed that performance is degrading over time while performing an archive operation (adding, querying, retrieving or deleting archived files), or have a large amount of storage consumed by archives, consider using some advanced techniques that can help.

"Archive Operations Overview"
"Managing Performance for Archive Operations" on page 427
"Managing Storage Usage for Archives" on page 429
"Auditing Archive Descriptions" on page 431
"Getting Information about Archives" on page 432

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.

Managing Performance for Archive Operations

Performance for archive operations is affected by:

- How the directories and files are named
- If the directories and files are organized in archive packages
- Whether or not a node is converted
- The number of directories and files that are archived

File specification names and archive packages can influence archive operation performance. The following list describes sample files and descriptions a client has archived. Consider using the file specifications and descriptions in the table below the list for query, retrieve and delete requests.

images for June 14, 1995
`c:\images\ctscan\patient1.1995jun14`

″ ″ (a blank)
`c:\images\ctscan\patient5.1998jun14`

images for November 21, 2001
`c:\images\mri\patient1.2001nov21`

images for February 22, 1995
`c:\images\mri\patient2.1995feb22`

images for August 8, 1998
`c:\images\mri\patient5.1998aug08`

images for January 1, 2003
`c:\images\xray\patient1.2003jan01`

images for November 21, 2001
`c:\images\xray\patient2.2001nov21`

File specification	Description	Example
File Specification	**Description**	**Example**
Completely specified (no wildcards)	Not specified	`retrieve c:\images\xray\patient1.2003jan01` • To use complete file specifications with archive functions, encode information in the path and file names which allows you to completely construct the name. • If the concept of archive packages will not be used, specify -description=″ ″ (a blank) when archiving a file to replace the default description supplied by the client and reduce the number of directories maintained for the path specification.
Completely specified (no wildcards)	Specified	`query archive c:\images\mri\patient2.1995feb22 -descr="images for 1995"` **Attention:** If you initiate archive operations by using the complete file specification, organizing the archives in packages will result in a set of directory entries (given the same name) for each package.

File specification	Description	Example
Completely wildcarded	Not specified	`query archive c:\*` • All files in filespace c:\ will be displayed. • If the concept of archive packages will never be used, specify -description=" " (a blank) when archiving a file to replace the default description supplied by the client and reduce the number of directories maintained for the path specification.
Completely wildcarded	Specified	`query archive c:\* -descr="images for 1995feb22"` This example demonstrates the usage of archive packages. Even though the date information is encoded in the file names, this use of packaging allows you to efficiently retrieve all images including the Cat Scan, MRI and XRay directories for all patients for February 22, 1995.

Partially specified	N/A	A partially specified file specification includes the wildcard character. Two examples of inefficient queries are:
		`q archive c:\images\*\patient1.2001nov21`
		`q archive c:\images\mri\*.1995feb22`
		
		For more efficient archive operations, specify as much of the path and file name to the left of the wildcard character as possible. For example:
		`queries: q archive c:\images\xray\patient2.2001*`
		
		Note: Archive operation efficiency can be improved by archive packaging.

Partially specified	Not specified	To retrieve all images for Patient 5, the following request will perform inefficiently:
		`c:\images\*\patient5.2001*:`
		
		Any of the following three requests will return the images more efficiently:
		`retrieve c:\images\ctscan\patient5.2001*`
		`retrieve c:\images\mri\patient5.2001*`
		`retrieve c:\images\xray\patient5.2001*`

Partially specified	Specified	To retrieve all images taken on February 22, 1995, the following request will not perform as efficiently because there is no description.
		`c:\images\*\*.1995feb22`
		
		The following request uses archive packaging with encoded file names and will perform more efficiently.
		`retrieve c:\images\*`
		`  -descr="images for 1995feb22"`

When a node organizes its archive data into packages, the node should be converted. See the following sections:

"Conversion of a Client Node" on page 429
"Converting a Client Node" on page 429
"Reversing the Conversion of a Client Node" on page 429

If a node does not organize archive data into packages, the FROMDATE and TODATE options can be used to distinguish a set of archived files. The date options, however, are not efficient search options.

Performance during inventory expiration is affected by the number of archive entries in the server database.

Conversion of a Client Node

Through a process called conversion, the server can search more efficiently for archive packages for clients that query, retrieve, or delete archive files by using the description option. The process occurs once, automatically, for client nodes that are accessed via a GUI client or a Web backup-archive client. To get the same effect for client nodes accessed only via the command line or the API, convert the node manually. See "Converting a Client Node."

Converting a node affects how the node's archive data is organized by the server, with the effect that archive operations work more efficiently. Converting a node can increase the amount of storage space used for archives. The decision whether to convert a particular node depends on how that node is used for archiving, and how important the node is.

Note: If most of the files archived for a node have unique names, users can perform searches efficiently by searching for file names instead of descriptions. Nodes with this characteristic might not need to be converted, if users know to perform their archive searches by description.

Converting a Client Node

Nodes accessed via graphical user interface clients or Web backup-archive clients are automatically converted.

For nodes accessed only by command-line clients or the client API, use the CONVERT ARCHIVE command to convert a node. For example to convert the node LINUS, enter:

```
convert archive linus
```

Reversing the Conversion of a Client Node

Once you convert a node, you do not need to convert it again. However, conversion of a client node can be reversed. For example, you might want to reverse conversion to test the effect on performance of the client node and on database usage, with and without conversion. Reversing conversion is also a step in auditing archive descriptions for a client node ("Auditing Archive Descriptions" on page 431).

To reverse the conversion operation for the node LINUS, enter the command:

```
undo archconversion linus
```

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 two actions that users can take to minimize the storage usage:

Minimize the number of unique descriptions

Users 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 a user archives 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 a user 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.

"Minimizing Archive Storage Use"

"Reducing Archive Storage Use for Client Nodes with Many Archives" on page 431

"Preventing Archive of Directory-level Access Permissions" on page 431

Minimizing Archive Storage Use

You can minimize the storage used for new archives for a client node that is *always* accessed via the command-line interface by ensuring that the user follows these guidelines:

- The user ensures that each file that is archived by the client node has a unique name.
- The user uses 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 *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. 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, you can keep the archive package count to a minimum by following the guidelines in "Minimizing Archive Storage Use" on page 430.

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.

Note: 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.

Auditing Archive Descriptions

You can audit archive descriptions to ensure that the information that is in the server database is internally consistent. You might need to audit archive descriptions when a client node that has been converted receives error messages about archives, archive descriptions, or the expiration of archived objects.

Use the following sequence of commands to audit archive descriptions for a node:

1. UNDO ARCHCONVERSION
2. CONVERT ARCHIVE

For example, to audit archive descriptions for the converted node LINUS, enter the commands:

```
undo archconversion linus
convert archive linus
```

Getting Information about Archives

You can get information about archives for a client node, such as how many directory entries the client node has, or whether the node has been converted. For example, for the client node LINUS, enter the command:

```
update archive linus showstats
```

The server returns a message such as the following:

```
ANR0898I Node DOC has (1) filespaces with (12) directories and
(28) files. There are (12) directories with different path
specifications and (0) not referenced by files. The node is
converted (No).
```

Chapter 15. Scheduling Operations for Client Nodes

This chapter contains information about scheduling the following operations:

- Backing up and restoring client data and IBM System Storage Archive Manager application client data.
- Archiving and retrieving client data.
- Running operating system commands.
- Running macro or command files that contain operating system commands, Tivoli Storage Manager commands, or both. You can schedule a command file to run on clients or application clients.

This chapter describes the following concepts:

Concepts:
"Prerequisites to Scheduling Operations"
"Comparing IBM Tivoli Storage Manager Scheduling Across Operating Systems and Components" on page 438
"Commands for Scheduling Client Operations" on page 439

Administrators perform the following activities to schedule Tivoli Storage Manager client operations:

Tasks:
"Scheduling a Client Operation" on page 434 (task overview)
"Defining Client Schedules" on page 434
"Associating Client Nodes with Schedules" on page 435
"Starting the Scheduler on the Clients" on page 435
"Displaying Schedule Information" on page 436
"Creating Schedules for Running Command Files" on page 437
"Updating the Client Options File to Automatically Generate a New Password" on page 438

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Prerequisites to Scheduling Operations

To interact with Tivoli Storage Manager for scheduling operations, a client machine must meet the following prerequisites:

- The client node must be registered with the server. For information, see Chapter 11, "Adding Client Nodes," on page 309.

© Copyright IBM Corp. 1993, 2005

- 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 313 for more information.
- The scheduler must be started on the client machine. Refer to the *Backup-Archive Clients Installation and User's Guide* for details.

Scheduling a Client Operation

To automate client operations, you can define new schedules. To later modify, copy, and delete these schedules, see Chapter 16, "Managing Schedules for Client Nodes," on page 441.

When you define a schedule, you assign it to a specific policy domain. You can define more than one schedule for each policy domain.

To set up a client schedule on the server, perform these steps:

1. Define a schedule (DEFINE SCHEDULE command). ("Defining Client Schedules" on page 434)
2. Associate client nodes with the schedule (DEFINE ASSOCIATION command). ("Associating Client Nodes with Schedules" on page 435)
3. Ensure that the clients start the client scheduler. ("Starting the Scheduler on the Clients" on page 435)
4. Display the schedule information and check that the schedule completed successfully (QUERY SCHEDULE and QUERY EVENT commands). ("Displaying Schedule Information" on page 436)

The following sections describe how to automate a basic client operation, incremental backup.

Defining Client Schedules

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

Key information to have when scheduling operations are:

- The operation that needs to run
- The time and day when the operation needs to run
- How often the operation needs to repeat

To define a schedule for daily incremental backups, use the DEFINE SCHEDULE command. You must specify the policy domain to which the schedule belongs and the name of the schedule (the policy domain must already be defined). For example:

```
define schedule engpoldom daily_backup starttime=21:00
duration=2 durunits=hours
```

This command results in the following:

- Schedule *DAILY_BACKUP* is defined for policy domain *ENGPOLDOM*.
- The scheduled action is an incremental backup; this is the default.

- The priority for the operation is 5; this is the default. If schedules conflict, the schedule with the highest priority (lowest number) runs first.
- The schedule window begins at 9:00 p.m., and the schedule itself has 2 hours to start.
- The start window is scheduled every day; this is the default.
- The schedule never expires; this is the default.
- The schedule style is classic; this is the default.

To change the defaults, see the DEFINE SCHEDULE command in the *Administrator's Reference*.

Associating Client Nodes with Schedules

Task	Required Privilege Class
Associate client nodes with schedules	System, unrestricted policy, or restricted policy for the policy domain to which the schedule belongs

Client nodes process operations according to the schedules associated with the nodes. To associate client nodes with a schedule, use the DEFINE ASSOCIATION command. 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.

After a client schedule is defined, you can associate client nodes with it by identifying the following information:
- Policy domain to which the schedule belongs
- List of client nodes to associate with the schedule

To associate the ENGNODE client node with the WEEKLY_BACKUP schedule, both of which belong to the ENGPOLDOM policy domain, enter:

```
define association engpoldom weekly_backup engnode
```

Starting the Scheduler on the Clients

The client scheduler must be started before work scheduled by the 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 319 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

Task	Required Privilege Class
Display information about scheduled operations	Any administrator

You can display information about schedules and whether the schedules ran successfully.

Displaying Schedule Details

When you request information about schedules, the server displays the following information:

- 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

For Tivoli Storage Manager, a schedule completes successfully if the command associated with the schedule is successfully issued. The success of the issued command is independent of the success of the schedule.

You need to ask these two questions:

- Did the schedule run successfully?

To determine the success of a scheduled operation, query the server. Each scheduled client operation is called an *event*, and is tracked by the server. You can get information about projected and actual scheduled processes by using the QUERY EVENT command. You can get information about scheduled processes that did not complete successfully by using exception reporting with this command.

For example, you can issue the following command to find out which events were missed (did not start) in the ENGPOLDOM policy domain for the WEEKLY_BACKUP schedule in the previous week:

```
query event engpoldom weekly_backup begindate=-7 begintime=now
enddate=today endtime=now exceptionsonly=yes
```

For more information about managing event records, see "Managing Event Records" on page 444.

- Did the operation or commands run as a result of the schedule run successfully?

  To determine the success of the commands issued as the result of a successful schedule, you can:

  – Check the client's schedule log.

    The schedule log is a file that contains information such as the statistics about the backed-up objects, the name of the server backing up the objects, and the time and date of the next scheduled operation. By default, Tivoli Storage Manager stores the schedule log as a file called *dsmsched.log* and places the file in the directory where the Tivoli Storage Manager backup-archive client is installed. Refer to *Backup-Archive Clients Installation and User's Guide* for more information.

  – Check the server's activity log.

    Search or query the activity log for related messages. For example, search for messages that mention the client node name, within the time period that the schedule ran. For example:

    ```
    query actlog begindate=02/23/2001 enddate=02/26/2001 originator=client
    nodename=hermione
    ```

  – Issue the QUERY EVENT command with FORMAT=DETAILED, and view the Result field of the output screen. For example:

    ```
    query event nodes=joe domain2 standard begindate=02/26/2002 enddate=02/27/2002
    format=detailed
    ```

    Refer to *Backup-Archive Clients Installation and User's Guide* for an explanation of the Result field.

Creating Schedules for Running Command Files

For some clients, you may want to run a command for a different application before running a Tivoli Storage Manager backup. For example, you may want to stop a database application, back up files with Tivoli Storage Manager, and then restart the application. To do this, you can schedule the running of a command file. Application clients *require* schedules that run command files.

A command file (also known as a macro or batch file on different operating systems) is stored on the client. This file contains a sequence of commands that are intended to be run during a scheduled start date and time window. Commands can include operating system commands, the Tivoli Storage Manager client's DSMC command, and commands for other applications.

To use command files, administrators must create schedules with the ACTION=MACRO parameter. For example, you can define a schedule called DAILY_INCR that will process a command file called *c:\incr.cmd* on the client:

```
define schedule standard daily_incr description="daily incremental file"
 action=macro objects="c:\incr.cmd" starttime=18:00 duration=5
 durunits=minutes period=1 perunits=day dayofweek=any
```

Associate the client with the schedule and ensure that the scheduler is started on the client or application client directory. The schedule runs the file called *c:\incr.cmd* once a day between 6:00 p.m. and 6:05 p.m., every day of the week.

Updating the Client Options File to Automatically Generate a New Password

If the server uses password authentication, clients must use passwords. Passwords are then also required for the server to process scheduled operations for client nodes. If the 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.

Comparing IBM Tivoli Storage Manager Scheduling Across Operating Systems and Components

The Tivoli Storage Manager scheduler provides the capability for the server to process scheduled operations. The Tivoli Storage Manager scheduler is installed when the backup-archive client or application client software is installed. The scheduler can be started by using the DSMC SCHEDULE command on the client's machine. For Windows clients, the Tivoli Storage Manager scheduler service must be configured separately. After installation, you can configure the client scheduler by selecting **Utilities > Setup Wizard** from the Tivoli Storage Manager client GUI.

For more information about installing, configuring, and starting the Tivoli Storage Manager scheduler, refer to the *Backup-Archive Clients Installation and User's Guide*.

The following table compares the scheduling environment across operating systems and components:

Component Type	Operating System	Scheduling Environment
Backup-Archive Client	UNIX or Linux, platforms other than Windows	The scheduler is installed as part of the client installation.
Backup-Archive Client	Windows	The scheduler is installed and configured separately.

Component Type	Operating System	Scheduling Environment
Data Protection for Oracle (application client) Data Protection for Domino (application client)	AIX, HP-UX, Linux, and Sun Solaris	Must use the Client Acceptor daemon (CAD) to manage the scheduler.
Data Protection for Microsoft Exchange (application client) Data Protection for SQL (application client) Data Protection for Domino (application client) Data Protection for Oracle (application client)	Windows	Must define a separate scheduler service.
Data Protection for Lotus Notes™ (application client)	Windows	Uses the Lotus Notes scheduler.

Commands for Scheduling Client Operations

This section summarizes example commands that can be used for the scheduling tasks that are discussed in this chapter. Refer to the *Administrator's Reference* for server command details.

Define a classic schedule for a client:

```
define schedule engpoldom daily_backup starttime=21:00
duration=2 durunits=hours
```

Associate a client with a schedule:

```
define association engpoldom weekly_backup engnode
```

On the client workstation, start the scheduler:
On most clients:

```
> dsmc schedule
```

On Windows clients:

```
> net start "TSM Scheduler"
```

Refer to the *Backup-Archive Clients Installation and User's Guide* for details about automatically starting the scheduler and running the scheduler in the background.

Display schedule information:

```
query schedule engpoldom
```

Check to see if the schedule ran successfully:

```
query event engpoldom weekly_backup begindate=-7 begintime=now
 enddate=today endtime=now
```

Chapter 16. Managing Schedules for Client Nodes

This chapter contains information about managing and coordinating IBM Tivoli Storage Manager schedules for registered client nodes. For a description of what Tivoli Storage Manager views as client nodes, see Chapter 11, "Adding Client Nodes," on page 309. For information about the Tivoli Storage Manager scheduler and creating schedules, see Chapter 15, "Scheduling Operations for Client Nodes," on page 433.

Administrators can perform the following tasks:

Tasks:
"Managing IBM Tivoli Storage Manager Schedules" on page 441
"Managing Node Associations with Schedules" on page 443
"Managing Event Records" on page 444
"Managing the Throughput of Scheduled Operations" on page 446
"Specifying One-Time Actions for Client Nodes" on page 452

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Managing IBM Tivoli Storage Manager Schedules

You can perform the following activities to manage schedules.

Task	Required Privilege Class
Verify that the schedule ran	Any administrator
Add, copy, modify, or delete client schedules in any policy domain	System or unrestricted policy
Add, copy, modify, or delete client schedules for specific policy domains	System, unrestricted policy, or restricted policy for those domains
Display information about scheduled operations	Any administrator

Verifying that the Schedule Ran

You will want to ensure that all defined nodes completed their scheduled operations. You can check whether the schedules ran successfully by using the QUERY EVENT command. For information, see "Displaying Information about Scheduled Events" on page 445.

You can also check the log file described in "Checking the Schedule Log" on page 442.

© Copyright IBM Corp. 1993, 2005

Checking the Schedule Log

The Tivoli Storage Manager client stores detailed information about each scheduled event in a file. This file contains information such as the statistics about the backed-up objects, the name of the server to which the objects are backed up, and the time and date of the next scheduled operation.

The default name for this file is *dsmsched.log*. The file is located in the directory where the Tivoli Storage Manager backup-archive client is installed. You can override this file name and location by specifying the SCHEDLOGNAME option in the client options file. See the client user's guide for more information.

Adding New Schedules

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 434.

Copying Existing Schedules

You can create new schedules by copying existing schedules to the same policy domain or a different policy domain. The schedule description and all schedule parameter values are copied to the new schedule. You can then modify the new schedule to meet site-specific requirements.

Client node associations are not copied to the new schedule. You must associate client nodes with the new schedule before it can be used. The associations for the old schedule are not changed. For information, see "Associating Client Nodes with Schedules" on page 435.

To copy the WINTER schedule from policy domain DOMAIN1 to DOMAIN2 and name the new schedule WINTERCOPY, enter:

```
copy schedule domain1 winter domain2 wintercopy
```

Modifying Schedules

You can modify existing schedules by using the UPDATE SCHEDULE command. For example, to modify the ENGWEEKLY client schedule in the ENGPOLDOM policy domain, enter:

```
update schedule engpoldom engweekly period=5 perunits=days
```

The ENGWEEKLY schedule is updated so that the incremental backup period is now every five days.

Deleting Schedules

When you delete a schedule, Tivoli Storage Manager deletes all client node associations for that schedule. See "Associating Client Nodes with Schedules" on page 435 for more information.

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 444.

Displaying Information about Schedules

When you request information about schedules, the server displays the following information:

- Schedule name
- Policy domain name
- Type of operation to be performed
- Start date and time for the initial startup window
- Duration of the startup window
- Time period between startup windows (if using a classic schedule)
- Day (classic schedule) or days (enhanced schedule) of the week on which scheduled operations can begin

The following output shows an example of a report for a classic schedule that is displayed after you enter:

```
query schedule engpoldom
```

```
Domain       * Schedule Name    Action Start Date/Time      Duration Period Day
------------ - ---------------- ------ -------------------- -------- ------ ---
ENGPOLDOM      MONTHLY_BACKUP   Inc Bk 09/04/2002 12:45:14      2 H    2 Mo Sat
ENGPOLDOM      WEEKLY_BACKUP    Inc Bk 09/04/2002 12:46:21      4 H    1 W  Sat
```

For enhanced schedules, the standard schedule format displays a blank period column and an asterisk in the day of week column. Issue FORMAT=DETAILED to display complete information about an enhanced schedule. Refer to the *Administrator's Reference* for command details. The following output shows an example of a report for an enhanced schedule that is displayed after you enter:

```
query schedule engpoldom
```

```
Domain       * Schedule Name    Action Start Date/Time      Duration Period Day
------------ - ---------------- ------ -------------------- -------- ------ ---
ENGPOLDOM      MONTHLY_BACKUP   Inc Bk 09/04/2002 12:45:14      2 H    2 Mo Sat
ENGPOLDOM      WEEKLY_BACKUP    Inc Bk 09/04/2002 12:46:21      4 H         (*)
```

Managing Node Associations with Schedules

You can add and delete node associations from schedules. Nodes can be associated with more than one schedule.

You can perform the following activities to manage associations of client nodes with schedules.

Task	Required Privilege Class
Add new nodes to existing schedules	System or restricted policy over the domain to which the node is assigned
Move nodes to existing schedules	System or restricted policy over the domain to which the node is assigned
Delete nodes associated with a schedule	System or restricted policy over the domain to which the node is assigned
Display nodes associated with a specific schedule	Any administrator

Adding New Nodes to Existing Schedules

You can add new nodes to existing schedules by associating the node with the schedule. To associate client nodes with a schedule, you can use the Administration Center or you can issue the DEFINE ASSOCIATION command from the command line interface. For information, see "Associating Client Nodes with Schedules" on page 435.

Moving Nodes from One Schedule to Another

You can move a node from one schedule to another schedule by:

1. Associating the node to the new schedule. For information, see "Associating Client Nodes with Schedules" on page 435.
2. Deleting the association of that node from the original schedule. For information, see "Removing Nodes from Schedules" on page 444.

Displaying Nodes Associated with Schedules

You can display information about the nodes that are associated with a specific schedule. For example, you should query an association before deleting a client schedule.

Figure 59 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 59. Query Association Output

Removing Nodes from Schedules

When you remove the association of a node to a client schedule, the client no longer runs operations specified by the schedule. However, the remaining client nodes still use the schedule.

To delete the association of the ENGNOD client with the ENGWEEKLY schedule, in the policy domain named ENGPOLDOM, enter:

```
delete association engpoldom engweekly 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.

You can perform the following activities to manage event records:

Task	Required Privilege Class
Display information about scheduled events	Any administrator
Set the retention period for event records	System
Delete event records	System or unrestricted policy

Displaying Information about Scheduled Events

To help manage schedules for client operations, you can request information about scheduled and completed events by using the QUERY EVENT command.

- To get information about past and projected scheduled processes, use a simple query for events. If the time range you specify includes the future, the results show which events should occur in the future based on current schedules.
- To get information about scheduled processes that did not complete successfully, use the exceptions-only option with the query.

To minimize the processing time when querying events:

- Minimize the time range
- For client schedules, restrict the query to those policy domains, schedules, and client node names for which information is required

Displaying All Client Schedule Events

You can display information about all client events by issuing the QUERY EVENT command. The information includes events for both successful and failed schedules. If the administrator specifies a time range that includes the future, Tivoli Storage Manager displays future events with a status of *future*.

Figure 60 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 60. Events for a Node

Displaying Events that Ended Unsuccessfully

You can display information about scheduled events that ended unsuccessfully by using exception reporting. For example, you can issue the following command to find out which events were missed in the previous 24 hours, for the DAILY_BACKUP schedule in the STANDARD policy domain:

```
query event standard daily_backup begindate=-1 begintime=now
enddate=today endtime=now exceptionsonly=yes
```

Figure 61 on page 446 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 61. Exception Report of Events

Displaying Past Events

If you query the server for events, the server may display past events even if the event records have been deleted. Such events are displayed with a status of *Uncertain*, indicating that complete information is not available because the event records have been deleted. To determine if event records have been deleted, check the message that is issued after the DELETE EVENT command is processed.

Managing Event Records in the Server Database

By default, the server retains event records for 10 days before automatically removing them from the database. The server automatically deletes event records from the database after the event retention period has passed and after the startup window for the event has elapsed.

You can specify how long event records stay in the database before the server automatically deletes them by using the SET EVENTRETENTION command. You can also manually delete event records from the database, if database space is required.

Setting the Event Retention Period

You can modify the retention period for event records in the database. To change the retention period to 15 days, enter:

```
set eventretention 15
```

Manually Deleting Event Records

You may want to manually delete event records to increase available database space. For example, to delete all event records written prior to 11:59 p.m. on June 30, 2002, enter:

```
delete event 06/30/2002 23:59
```

Managing the Throughput of Scheduled Operations

In the Tivoli Storage Manager environment where many nodes attempt to initiate scheduled operations simultaneously, you may have to manage scheduling throughput. You can choose a scheduling mode, and you can control how often client nodes contact the server to perform a scheduled operation.

By default, clients contact the server. To limit the start of backup-archive client sessions to the server only, change the SESSIONINITIATION parameter to SERVERONLY either on the REGISTER NODE command or on the UPDATE NODE command, and 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. By doing so, you specify that the server will not accept client requests for sessions.

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. If you select the CLIENTORSERVER option, 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.

Administrators can perform the following activities to manage the throughput of scheduled operations.

Task	Required Privilege Class
Modify the default scheduling mode	System
Modify the scheduling period for incremental backup operations	System
Balance the scheduled workload for the server	System
Set the frequency at which client nodes contact the server	System

Modifying the Default Scheduling Mode

Tivoli Storage Manager provides two scheduling modes: *client-polling* and *server-prompted*. The mode indicates how client nodes interact with the server for scheduling operations. With client-polling mode, client nodes poll the server for the next scheduled event. With server-prompted mode, the server contacts the nodes at the scheduled start time.

By default, the server permits both scheduling modes. The default (ANY) allows nodes to specify either scheduling mode in their client options files. You can modify this scheduling mode.

If you modify the default server setting to permit only one scheduling mode, *all* client nodes must specify the same scheduling mode in their client options file. Clients that do not have a matching scheduling mode will not process the scheduled operations. The default mode for client nodes is client-polling.

The scheduler must be started on the client node's machine before a schedule can run in either scheduling mode.

For more information about modes, see "Overview of Scheduling Modes."

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 32 on page 448 and Table 31 on page 448.

Table 31. 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 32. 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.

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 set the scheduling mode on client nodes. They specify either the client-polling or the server-prompted scheduling mode on the command line or in the client user options file. (On UNIX and Linux systems, root users set the scheduling mode in the client system options file.)

For more information, refer to the appropriate *Backup-Archive Clients Installation and User's Guide*.

Specifying the Schedule Period for Incremental Backup Operations

When you define a backup copy group, you specify the copy frequency, which is the minimum interval between successive backups of a file. When you define a schedule, you specify the length of time between processing of the schedule. Consider how these interact to ensure that the clients get the backup coverage that you intend.

See "Defining and Updating a Backup Copy Group" on page 381.

Balancing the Scheduled Workload for the Server

You can control the server's workload and ensure that the server can perform all scheduled operations within the specified window. To enable the server to complete all schedules for clients, you may need to use trial and error to control the workload. To estimate how long client operations take, test schedules on several representative client nodes. Keep in mind, for example, that the first incremental backup for a client node takes longer than subsequent incremental backups.

You can balance the server's scheduled workload by:

- Adjusting the number of sessions that the server allocates to scheduled operations
- Randomizing scheduled start time for client operations (if clients use client-polling scheduling mode)
- Increasing the length of the startup window

Setting the Number of Sessions the Server Allocates to Scheduled Operations

The maximum number of concurrent client/server sessions is defined by the MAXSESSIONS server option. Of these sessions, you can set a maximum percentage to be available for processing scheduled operations. Limiting the number of sessions available for scheduled operations ensures that sessions are available when users initiate any unscheduled operations, such as restoring file or retrieving files.

If the number of sessions for scheduled operations is insufficient, you can increase either the total number of sessions or the maximum percentage of scheduled sessions. However, increasing the total number of sessions can adversely affect server performance. Increasing the maximum percentage of scheduled sessions can reduce the server availability to process unscheduled operations.

For example, assume that the maximum number of sessions between client nodes and the server is 80. If you want 25% of these sessions to be used by for scheduled operations, enter:

```
set maxschedsessions 25
```

The server then allows a maximum of 20 sessions to be used for scheduled operations.

The following table shows the tradeoffs 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. A startup window is defined by the start time and duration during which a schedule must be initiated. For example, if the start time is 1:00 a.m. and the duration is 4 hours, the startup window is 1:00 a.m. to 5:00 a.m.

For the client-polling scheduling mode, you can specify the percentage of the startup window that the server can use to randomize start times for different client nodes associated with a schedule.

If you set randomization to 0, no randomization occurs. This process can result in communication errors if many client nodes try to contact the server at the same instant.

The settings for randomization and the maximum percentage of scheduled sessions can affect whether schedules are successfully completed for client nodes. Users receive a message if all sessions are in use when they attempt to process a schedule. If this happens, you can increase randomization and the percentage of scheduled sessions allowed to make sure that the server can handle the workload. The maximum percentage of randomization allowed is 50%. This limit ensures that half of the startup window is available for retrying scheduled commands that have failed.

To set randomization to 50%, enter:

```
set randomize 50
```

It is possible, especially after a client node or the server has been restarted, that a client node may not poll the server until *after* the beginning of the startup window in which the next scheduled event is to start. In this case, the starting time is randomized over the specified percentage of the *remaining* duration of the startup window.

Consider the following situation:

- The schedule start time is 8:00 a.m. and its duration is 1 hour. Therefore the startup window for the event is from 8:00 to 9:00 a.m.
- Ten client nodes are associated with the schedule.
- Randomization is set to 50%.
- Nine client nodes poll the server before 8:00 a.m.
- One client node does not poll the server until 8:30 a.m.

The result is that the nine client nodes that polled the server *before* the beginning of the startup window are assigned randomly selected starting times between 8:00 and 8:30. The client node that polled at 8:30 receives a randomly selected starting time that is between 8:30 and 8:45.

Increasing the Length of the Schedule Startup Window

Increasing the size of the startup window (by increasing the schedule's duration) can also affect whether a schedule completes successfully. A larger startup window gives the client node more time to attempt initiation of a session with the server.

Controlling How Often Client Nodes Contact the Server

To control how often client nodes contact the server to perform a scheduled operation, an administrator can set:

- How often nodes query the server (see "Setting How Often Clients Query the Server")
- The number of command retry attempts (see "Setting the Number of Command Retry Attempts" on page 452)
- The amount of time between retry attempts (see "Setting the Amount of Time between Retry Attempts" on page 452)

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.

Setting How Often Clients Query the Server

When scheduling client nodes with client-polling scheduling, you can specify how often the nodes query the server for a schedule. If nodes poll frequently for schedules, changes to scheduling information (through administrator commands) are propagated more quickly to the nodes. However, increased polling by client nodes also increases network traffic.

For the client-polling scheduling mode, you can specify the maximum number of hours that the scheduler on a client node waits between attempts to contact the server to obtain a schedule. You can set this period to correspond to the frequency with which the schedule changes are being made. If client nodes poll more frequently for schedules, changes to scheduling information (through administrator commands) are propagated more quickly to client nodes.

If you want to have all clients using polling mode contact the server every 24 hours, enter:

```
set 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.

The maximum number of command retry attempts does not limit the number of times that the client node can contact the server to obtain a schedule. The client node never gives up when trying to query the server for the next schedule.

Be sure not to specify so many retry attempts that the total retry time is longer than the average startup window.

If you want to have all client schedulers retry a failed attempt to process a scheduled command up to two times, enter:

```
set maxcmdretries 2
```

Maximum command retries can also be set on each client with a client option, MAXCMDRETRIES. The server value overrides the client value once the client successfully contacts the server.

Setting the Amount of Time between Retry Attempts

You can specify the length of time that the scheduler waits between command retry attempts. Command retry attempts occur when a client node is unsuccessful in establishing a session with the server or when a scheduled command fails to process. Typically, this setting is effective when set to half of the estimated time it takes to process an average schedule.

If you want to have the client scheduler retry every 15 minutes any failed attempts to either contact the server or process scheduled commands, enter:

```
set retryperiod 15
```

You can use this setting in conjunction with the SET MAXCMDRETRIES command (number of command retry attempts) to control when a client node contacts the server to process a failed command. See “Setting the Number of Command Retry Attempts.”

The retry period can also be set on each client with a client option, RETRYPERIOD. The server value overrides the client value once the client successfully contacts the server.

Specifying One-Time Actions for Client Nodes

You can use the DEFINE CLIENTACTION command to specify that one or more client nodes perform a one-time action if the client schedulers are active. If the scheduling mode is set to prompted, the client performs the action within 3 to 10 minutes. If the scheduling mode is set to polling, the client processes the command at its prescribed time interval. The time interval is set by the QUERYSCHEDPERIOD client option.

The DEFINE CLIENTACTION command causes Tivoli Storage Manager to automatically define a schedule and associate client nodes with that schedule. The schedule name and association information is returned to the server console or the administrative client with messages ANR2500I and ANR2510I. 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.

For example, you can issue a DEFINE CLIENTACTION command that specifies an incremental backup command for client node HERMIONE in domain ENGPOLDOM:

```
define clientaction hermione domain=engpoldom action=incremental
```

Tivoli Storage Manager defines a schedule and associates client node HERMIONE with the schedule. The server assigns the schedule priority 1, sets the period units (PERUNITS) to ONETIME, and determines the number of days to keep the schedule active based on the value set with SET CLIENTACTDURATION command.

For a list of valid actions, see the DEFINE CLIENTACTION command in the *Administrator's Reference*. You can optionally include the OPTIONS and OBJECTS parameters.

Determining How Long the One-Time Schedule Remains Active

You can determine how long schedules that were defined via DEFINE CLIENTACTION commands remain active by using the SET CLIENTACTDURATION command. This 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

© Copyright IBM Corp. 1993, 2005

Chapter 17. Managing Server Operations

Administrators can perform such server operations as licensing purchased features, starting and halting the server, and monitoring server information. See the following sections:

Tasks:
“Licensing IBM Tivoli Storage Manager”
“Starting and Halting the Server” on page 459
“Moving the IBM Tivoli Storage Manager Server” on page 463
“Changing the Date and Time on the Server” on page 464
“Managing Server Processes” on page 465
“Preemption of Client or Server Operations” on page 467
“Setting the Server Name” on page 468
“Adding or Updating Server Options” on page 468
“Automatic Tuning of Server Options” on page 469
“Getting Help on Commands and Error Messages” on page 469

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see “Managing Servers with the New Administration Center” on page 36.

Licensing IBM Tivoli Storage Manager

This section describes the 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 address at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

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).

© Copyright IBM Corp. 1993, 2005

- Network Enabler (network connections for clients).
- AFS/DFS Support, (the S/390® platform includes the S/390 UNIX client as part of Managed System for SAN).

Registering Licensed Features

You must register a new license if you want to add support for any of the following features that are not already in your existing license agreement. Tivoli Storage Manager uses a license file and the REGISTER LICENSE command to complete this task. Licenses are stored in enrollment certificate files, which contain licensing information for the server product. The enrollment certificate files are on the installation CD-ROM. When registered, the licenses are stored in a **nodelock** file within the current directory.

See the IBM Tivoli Storage Manager *readme* file for the latest information regarding enrollment certificate files.

To register a license, you must issue the REGISTER LICENSE command as well as the license file associated with the license. For example, to use two Tivoli Storage Manager for Space Management (HSM clients), issue the following command:

```
register license file=spacemgr.lic number=2
register license file=mgsyslan.lic number=2
```

To register 20 managed systems that move data over a local area network, issue the following command:

```
register license file=mgsyslan.lic  number=20
```

To register 10 IBM Tivoli Storage Manager for Lotus Notes clients that move data over a LAN, using the TCP/IP communication protocol, issue the following commands:

```
register license file=lnotes.lic number=10
register license file=mgsyslan.lic number=10
```

Certain license files are incremental, while others are of the yes- or no-type. With the exception of the yes- or no-type licenses, you can specify any number of license files to register. Always specify the total number of licenses you want registered. The REGISTER LICENSE command updates the **nodelock** file based on the total number of licenses you want registered. If you have twenty licenses and require ten additional licenses, you must register thirty.

To unregister a license or reduce the quantity of your registered licenses, you must erase the **nodelock** file found in the server directory of your installation. This will also require you to reregister any previously registered licenses.

For example, to register IBM Tivoli Storage Manager for 30 managed systems that move data over a local area network, issue the following commands:

```
register license file=mgsyslan.lic number=30
```

You can also register a license by specifying the product password that is included in the license certificate file. For example:

```
register license 5s3qydpnwx7njdxnafksqas4
```

Attention: IBM Tivoli Storage Manager licenses are associated with the CPU chip of the machine on which IBM Tivoli Storage Manager is installed. If you change that CPU chip, you must first erase the existing **nodelock** files and then reregister all your licenses.

Saving Your Licenses

Save the CD-ROM containing your enrollment certificate files. You may you need to register your licenses again for any of the following reasons:

- The server is corrupted.
- The server is moved to a different machine.
- The **nodelock** file is destroyed or corrupted. IBM Tivoli Storage Manager stores license information in the **nodelock** file, which is located in the directory from which the server is started.

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 or authorized reseller to modify your agreement.

An administrator can monitor license compliance by:

Auditing licenses
Use the AUDIT LICENSES command to compare the current configuration with the current licenses.

Note: During a license audit, the server calculates, by node, the amount of backup, archive, and space management storage in use. This calculation can take a great deal of CPU time and can stall other server activity. Use the AUDITSTORAGE server option to specify that storage is not to be calculated as part of a license audit.

Displaying license information
Use the QUERY LICENSE command to display details of your current licenses and determine licensing compliance.

Scheduling automatic license audits
Use the SET LICENSEAUDITPERIOD command to specify the number of days between automatic audits.

Starting and Halting the Server

Task	Required Privilege Class
Start, halt, and restart the server	System or operator

Starting the Server

The following events occur when you start or restart the IBM Tivoli Storage Manager server:

- The server invokes the communication methods specified in the server options file.
- The server uses the volumes specified in the dsmserv.dsk file for the database and recovery log to record activity. It also identifies storage pool volumes to be used.
- The server starts an IBM Tivoli Storage Manager server console session that is used to operate and administer the server until administrative clients are registered to the server.

To start the server, complete the following steps:

1. Change to the /opt/tivoli/tsm/server/bin directory from a Linux session.
   Enter:
   ```
   >cd /opt/tivoli/tsm/server/bin
   ```
2. Start the server by entering:
   ```
   dsmserv
   ```

   Note: If the server does not start, set the ulimit parameter to unlimited. For example,
   ```
   ulimit -d unlimited
   ```

When the server is started, IBM Tivoli Storage Manager displays the following information:

- Product licensing and copyright information
- Processing information about the server options file
- Communication protocol information
- Database and recovery log information
- Storage pool volume information
- Server generation date
- Progress messages and any errors encountered during server initialization

If IBM Tivoli Storage Manager detects an invalid system date and time, the server is disabled, and expiration, migration, reclamation, and volume history deletion operations are not allowed. An error message (ANR0110E) is displayed. 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. After the system date is resolved, you must issue the ENABLE SESSIONS command to re-enable the server for client sessions.

The date and time check occur when the server is started and once each hour thereafter. An invalid date is one that is:

- Earlier than the server installation date and time
- More than one hour earlier than the last time the date was checked
- More than 30 days later than the last time the date was checked

Automating Server Startup

To configure the IBM Tivoli Storage Manager server to start automatically, perform the following steps:

1. Log in to the system as user root

2. Copy the file dsmserv.rc from the directory where the server is installed to /etc/init.d:, for example:

   ```
   cp /opt/tivoli/tsm/server/bin/dsmserv.rc /etc/init.d
   ```

3. Check to ensure that the file has execute permission, for example:

   ```
   chmod 755 /etc/init.d/dsmserv.rc
   ```

4. Create a link to /etc/init.d/dsmserv.rc called S90dsmserv in /etc/rc.d/rc**x**.d, where **x** is the run level. The run level is usually 3 or 5 depending on whether you're booting into X-windows or not.

Defining Environment Variables

If you want to run the IBM Tivoli Storage Manager server from a directory other than the default directory or to run multiple servers, you may have to define environment variables.

An *environment variable* describes the operating environment of a process, such as the home directory or the terminal in use. It provides the path that the server requires to find and create files.

For example, to define the DSMSERV_DIR environment variable to point to the opt/tivoli/tsm/server/bin directory so that the server can find various files, such as dsmreg.lic or the message file (*dsmameng.txt*) enter:

```
export DSMSERV_DIR=/opt/tivoli/tsm/server/bin
```

You can also define an environment variable to point to the server options file. For example, to define the DSMSERV_CONFIG environment variable to point to the server options file, enter:

```
export DSMSERV_CONFIG=/opt/tivoli/tsm/server/bin/ filename.opt
```

where *filename* is the name you assigned your server options file (dsmserv.opt).

Notes:

1. The -o parameter of the DSMSERV command can also be used to specify an options file name.
2. Use the *set environment* command if your shell is in the "bash" family:

   ```
   > setenv DSMSERV_DIR /opt/tivoli/tsm/server/bin
   ```

3. If you want to save this environment, save these entries in the *.bash* file of your $HOME directory.
4. The dsmserv.dsk is always read from the directory in which the server is started.

Running Multiple Servers on a Single Machine

To have multiple servers running on a single machine, issue the DSMSERV FORMAT command from different directories to create multiple pairs of recovery log and database files. You do not have to install the server executable files in more than one directory.

However, if non-root users will be running servers, you must modify the access permission by adding read permission to the following files:

- dsmlicense
- dsmtli.drv

Use these commands as a root user to modify the permission for these files:

```
> chmod 755 /opt/tivoli/tsm/server/bin/dsmlicense
> chmod 755 /opt/tivoli/tsm/server/bin/dsmtli.drv
```

The following procedure shows how to set up an additional IBM Tivoli Storage Manager server:

1. Determine the directory where you want the server files created, for example, /usr/tivoli/tsm/myserver, and make that directory:

   ```
   > mkdir /opt/tivoli/tsm/myserver
   ```

2. Copy the dsmserv.opt file to your directory:

   ```
   > cp /opt/tivoli/tsm/server/bin/dsmserv.opt
   /opt/tivoli/tsm/myserver/dsmserv.opt
   ```

   Note: Ensure that the communication parameters are unique among all other IBM Tivoli Storage Manager servers. The communication protocols are:

   - TCPPORT for TCP/IP
   - HTTPPORT for HTTP Access in the Web Administrative Client Browser

   For example, if your first server is using the default TCPPORT of 1500, ensure that the new server is using a TCPPORT other than 1500 by providing a real value in the server options file.

3. Set your path on the server console. Define your environment variables, for example:

   ```
   export DSMSERV_DIR=/opt/tivoli/tsm/server/bin
   ```

   Ensure that you are in the target directory before continuing.

4. Format the database and recovery log files, for example:

   ```
   > /opt/tivoli/tsm/server/bin/dsmfmt -m -db dbvol2 5
   > /opt/tivoli/tsm/server/bin/dsmfmt -m -log logvol2 9
   ```

   In this example, db indicates the database log, -m indicates megabytes and log indicates the recovery log. Refer to *Administrator's Reference* for more information on these commands.

5. Create the database and recovery log in the desired directory for the new server, for example:

   ```
   > /opt/tivoli/tsm/server/bin/dsmserv format 1 logvol2 1 dbvol2
   ```

   Note: You need additional license authorizations to run additional servers. You can use the register license file command to register these licenses. See "Registering Licensed Features" on page 458 for more information.

Halting the Server

You can halt the server without warning if an unplanned operating system problem requires the server to be stopped.

When you halt the server, all processes are abruptly stopped and client sessions are canceled, even if they are not complete. Any in-progress transactions are rolled back when the server is restarted. Administrator activity is not possible.

If possible, halt the server only after current administrative and client node sessions have completed or canceled. To shut down the server without severely impacting administrative and client node activity with the server, you must:

1. Disable the server to prevent new client node sessions from starting by issuing the DISABLE SESSIONS command. This command does not cancel sessions currently in progress or system processes like migration and reclamation.

2. Notify any existing administrative and client node sessions that you plan to shut down the server. The server does not provide a network notification facility; you must use external means to notify users.
3. Cancel any existing administrative or client node sessions by issuing the CANCEL SESSION command and the associated session number. To obtain session numbers and determine if any sessions are running, use the QUERY SESSION command. If a session if running, a table will appear showing the session number on the far left side of the screen.
4. Find out if any other processes are running, such as server migration or inventory expiration, by using the QUERY PROCESS command. If a database backup process is running, allow it to complete before halting the server. If other types of processes are running, cancel them by using the CANCEL PROCESS command.

   Note: If the process you want to cancel is currently waiting for a tape volume to be mounted (for example, a process initiated by EXPORT, IMPORT, or MOVE DATA commands), the mount request is automatically cancelled. If a volume associated with the process is currently being mounted by an *automated* library, the cancel may not take effect until the mount is complete.
5. Halt the server to shut down all server operations by using the HALT command.

Note: The QUIESCE option on the HALT command is recommended *only* if you plan to do a database dump by using the DSMSERV DUMPDB command immediately after halting. Because IBM Tivoli Storage Manager supports online database backup (BACKUP DB command), the DSMSERV DUMPDB command should be rarely, if ever, needed.

The HALT command can be replicated using the ALIASHALT server option. The server option allows you to define a term other than HALT that will perform the same function. The HALT command will still function, however the server option provides an additional method for issuing the HALT command. See the *Administrator's Reference* for more information.

Stopping the Server When Running as a Background Process

If you started the server as a background process and want to stop the server, connect to the server as an administrative client and issue the HALT command. If you cannot connect to the server with an administrative client and you want to stop the server, you must cancel the process by using the **kill** command with the process ID number (pid) that is displayed at initialization.

Note: Before you issue the **kill** command, ensure that you know the correct process ID for the IBM Tivoli Storage Manager server.

The dsmserv.lock file, in the directory from which the server is running, can be used to identify the process ID of the process to kill. To display the file enter:

```
cat /opt/tivoli/tsm/server/bin/dsmserv.lock
```

Moving the IBM Tivoli Storage Manager Server

There are two ways to move an IBM Tivoli Storage Manager server to a new machine that is running the same operating system:

- Back up the database, and restore it on another machine on which IBM Tivoli Storage Manager is installed. The backup and restore method is described in this section.
- Export the server directly to the new machine, or to sequential media so that it can be imported to the new machine See Chapter 22, "Exporting and Importing Data," on page 609 for details. The export and import method may take longer than the backup and restore method.

The prerequisites for the backup and restore method are:

- The same operating system must be running on both machines.
- The sequential storage pool must be accessible from both machines. You will have to move any libraries and devices from the original machine to the new machine or they must be accessible through a SAN.
- The restore operation must be done by a IBM Tivoli Storage Manager server at a code level that is the same as or later than that on the machine that was backed up.
- Only manual and SCSI library types are supported for the restore operation.

In the following example, the IBM Tivoli Storage Manager server on machine A is moved to machine B.

- **On machine A:**
  1. Migrate all disk storage pool data to sequential media. See "Migration for Disk Storage Pools" on page 239 for details.
  2. Perform a full database backup to sequential media.

     ```
     backup db devclass=8mm type=full
     ```
  3. Copy the volume history file and the device configuration file.
- **On machine B:**
  1. Install IBM Tivoli Storage Manager.
  2. Restore the database:

     ```
     dsmserv restore db devclass=8mm volumenames=vol001,vol002,vol002
     ```

Changing the Date and Time on the Server

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.

If the server detects an invalid date or time, server sessions become disabled. 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

Task	Required Privilege Class
Display information about a server background process	Any administrator
Cancel a server process	System

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.

Most processes occur quickly and are run in the foreground, but others that take longer to complete run as background processes.

The server runs the following operations as background processes:
- Auditing licenses
- Auditing a volume
- Backing up the database
- Backing up a storage pool
- Defining a database volume copy
- Defining a recovery log volume copy
- Deleting a database volume
- Deleting a file space
- Deleting a recovery log volume
- Deleting a storage volume
- Expiring the inventory
- Exporting or importing data
- Extending the database or recovery log
- 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
- Reducing the database or recovery log
- Restoring a storage pool
- Restoring a volume
- Varying a database or recovery log volume online

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. However, 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 in the following command:

```
cancel process 2
```

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.

  Notes:

  1. To list open mount requests, issue the QUERY REQUEST command. You can also query the activity log to determine if a given process has a pending mount request.
  2. A mount request indicates that a volume is needed for the current process. However, the volume might not be available in the library. If the volume is not available, the reason might be that you either issued the MOVE MEDIA command or CHECKOUT LIBVOLUME command, or that you manually removed the volume from the library.

Preemption of Client or Server Operations

The server can preempt server or client operations for a higher priority operation when a mount point is in use and no others are available, or access to a specific volume is required. You can use the QUERY MOUNT command to see the status of the volume for the mount point.

Mount Point Preemption

The following high priority operations can preempt operations for a mount point:
- Backup database
- Restore
- Retrieve
- HSM recall
- Export
- Import

The following operations cannot preempt other operations nor can they be preempted:
- Audit Volume
- Restore from a copy storage pool
- Prepare a recovery plan
- Store data using a remote data mover

The following operations can be preempted and are listed in order of priority. The server selects the lowest priority operation to preempt, for example reclamation.
1. Move data
2. Migration from disk to sequential media
3. Backup, archive, or HSM migration
4. Migration from sequential media to sequential media
5. Reclamation

You can disable preemption by specifying NOPREEMPT in the server options file. When this option is specified, the BACKUP DB command is the only operation that can preempt other operations.

Volume Access Preemption

A high priority operation that requires access to a specific volume currently in use by a low priority operation can automatically preempt the operation. For example, if a restore request requires access to a volume in use by a reclamation process and a drive is available, the reclamation process is canceled and message ANR0494I or ANR1441I is issued.

The following high priority operations can preempt operations for access to a specific volume:
- Restore
- Retrieve
- HSM recall

The following operations cannot preempt other operations nor can they be preempted:
- Audit Volume
- Restore from a copy storage pool
- Prepare a recovery plan
- Store data using a remote data mover

The following operations can be preempted, and are listed in order of priority. The server preempts the lowest priority operation, for example reclamation.

1. Move data
2. Migration from disk to sequential media
3. Backup, archive, or HSM migration
4. Migration from sequential media
5. Reclamation

You can disable preemption by specifying NOPREEMPT in the server options file. When this option is specified, no operation can preempt another operation for access to a volume.

Setting the Server Name

Task	Required Privilege Class
Specify the server name	System

At installation, the server name is set to SERVER1. After installation, you can use the SET SERVERNAME command to change the server name. You can use the QUERY STATUS command to see the name of the server.

To specify the server name as WELLS_DESIGN_DEPT., for example, enter the following:

```
set servername wells_design_dept.
```

You must set unique names on servers that communicate with each other. See "Setting Up Communications Among Servers" on page 568 for details.

Attention: Changing any values on a source server that is used for virtual volume operations can impact the ability of the source server to access and manage the data it has stored on the corresponding target server. On a network where clients connect to multiple servers, it is recommended that all of the servers have unique names. Changing the server name after Windows clients are connected forces the clients to re-enter the passwords.

Changing the server name using the SET SERVERNAME command may have additional implications varying by platform. Some examples to be aware of are:

- Passwords may be invalidated
- Device information may be affected
- Registry information on Windows platforms may change

Adding or Updating Server Options

Task	Required Privilege Class
Add or update a server option	System

You can add or update server options by editing the dsmserv.opt file, using the SETOPT command. For information about editing the server options file, refer to *Administrator's Reference*.

Adding or Updating a Server Option without Restarting the Server

A system administrator can add or update a limited number of server options without stopping and restarting the server. The added or updated server option is appended to the end of the server options file.

The following example shows how to use the SETOPT command to update the existing server option for MAXSESSIONS:

```
setopt maxsessions 20
```

The following lists server options that can be added or updated:

- BUFPOOLSIZE
- COMMTIMEOUT
- EXPINTERVAL
- EXPQUIET
- IDLETIMEOUT
- LOGWARNFULLPERCENT
- MAXSESSIONS
- RESTOREINTERVAL
- THROUGHPUTDATATHRESHOLD
- THROUGHPUTTIMETHRESHOLD

Note: SETOPT commands in a macro cannot be rolled back.

Automatic Tuning of Server Options

For optimal performance, the server can tune the BUFPOOLSIZE server option automatically. To do this, set the SELFTUNEBUFPOOLSIZE option to YES. Before expiration processing, the server resets the database buffer pool and examines the database buffer pool cache hit ratio. The server accounts for the amount of available storage and adjusts the buffer pool size as needed. For information about the SELFTUNEBUFPOOLSIZE and BUFPOOLSIZE server options, refer to *Administrator's Reference*.

Getting Help on Commands and Error Messages

Any administrator can issue the HELP command to display information about administrative commands and messages from the server and the administrative command-line client. You can issue the HELP command with no operands to display a menu of help selections. You also can issue the HELP command with operands that specify help menu numbers, commands, or message numbers.

To display the help menu, enter:

```
help
```

To display help information on the REMOVE commands, enter:

```
help remove
```

To display help information on a specific message, such as ANR0992I for example, enter:

```
help 0992
```

Additional information is also available in the online documentation.

Chapter 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.

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 Tivoli Storage Manager scripts that contain a sequence of commands with WAIT=YES. You can also use a scheduler external to Tivoli Storage Manager 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.

See the following sections:

Tasks:
"Automating a Basic Administrative Command Schedule"
"Tailoring Schedules" on page 473
"Copying Schedules" on page 476
"Deleting Schedules" on page 476
"Managing Scheduled Event Records" on page 476
"IBM Tivoli Storage Manager Server Scripts" on page 477
"Using Macros" on page 484

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Automating a Basic Administrative Command Schedule

This section describes how to set up a basic administrative command schedule using Tivoli Storage Manager defaults. To later update or tailor your schedules, see "Tailoring Schedules" on page 473.

© Copyright IBM Corp. 1993, 2005

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 IBM Tivoli Storage Manager Activity Log" on page 523.
2. You cannot schedule MACRO or QUERY ACTLOG commands.

Task	Required Privilege Class
Define, update, copy, or delete administrative schedules	System
Display information about scheduled operations	Any administrator

Defining the Schedule

Use the DEFINE SCHEDULE command to create a new schedule for processing an administrative command. Include the following parameters:

- Specify the administrative command to be issued (CMD=).
- Specify whether the schedule is activated (ACTIVE=).

For example:

```
define schedule backup_archivepool type=administrative
cmd='backup stgpool archivepool recoverypool' active=yes
```

This command results in the following:

- The schedule created is *BACKUP_ARCHIVEPOOL*.
- The schedule is to process the administrative command:

  ```
  backup stgpool archivepool recoverypool
  ```

  This command specifies that primary storage pool ARCHIVEPOOL is backed up to the copy storage pool RECOVERYPOOL.
- The schedule is currently active.
- Administrative command output is redirected to the activity log.
- The following defaults are in effect:
  - The start date and time defaults to the current date and time.
  - The length of the startup window is 1 hour.
  - The priority for the schedule is 5. If schedules conflict, the schedule with the highest priority (lowest number) is run first.
  - The schedule never expires.

To change the defaults, see "Tailoring Schedules" on page 473.

Verifying the Schedule

You can verify the details of what you have scheduled by using the QUERY SCHEDULE command. When you use the QUERY SCHEDULE command, you must specify the TYPE=ADMINISTRATIVE parameter to view an administrative command schedule. The following figure shows an example of a report that is displayed after you enter:

```
query schedule backup_archivepool type=administrative
```

```
*   Schedule Name        Start Date/Time        Duration   Period   Day
-   ----------------     --------------------   --------   ------   ---
     BACKUP_ARCHIVE-     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 476.

Tailoring Schedules

To control more precisely when and how your schedules run, specify values for schedule parameters instead of accepting the defaults when you define or update schedules.

Schedule name
All schedules must have a unique name, which can be up to 30 characters.

Schedule style
You can specify either classic or enhanced scheduling. With classic scheduling, you can define the interval between the startup windows for a schedule. With enhanced scheduling, you can choose the days of the week, days of the month, weeks of the month, and months the startup window can begin on.

Initial start date, initial start time, and start day
You can specify a past date, the current date, or a future date for the initial start date for a schedule with the STARTDATE parameter.

You can specify a start time, such as 6 p.m. with the STARTTIME parameter.

For classic scheduling, you can use the DAYOFWEEK parameter to specify that the startup window begins on a certain day of the week, over the weekend, during the week, or on any day. If the start date and time specified fall on a day that does not correspond to your value for the day of the week, the start date and time are shifted forward in 24-hour increments until the day of the week is satisfied. If you select a value for the day of the week other than ANY, schedules may not process when you expect. This depends on the values for PERIOD and PERUNITS. Use the QUERY EVENT command to project when schedules will process to ensure that you achieve the desired result.

For enhanced scheduling, you can use the DAYOFWEEK parameter to specify that the startup window begin on one or more days of the week, over the weekend, during the week, or on any day. MONTH, DAYOFMONTH, and WEEKOFMONTH can also determine a start date. When used with the DAYOFMONTH parameter, DAYOFWEEK must have a value of ANY. If DAYOFWEEK=WEEKDAY or DAYOFWEEK=WEEKEND, you must specify a value of either FIRST or LAST for the parameter WEEKOFMONTH.

Duration of a startup window
You can specify the duration of a startup window, such as 12 hours, with

the DURATION and DURUNITS parameters. The server must start the scheduled service within the specified duration, but does not necessarily complete it within that period of time. If the schedule needs to be retried for any reason, the retry attempt must begin before the startup window elapses or the operation does not restart.

If the schedule does not start during the startup window, the server records this as a *missed event* in the database. You can get an exception report from the server to identify schedules that did not run. For more information, see "Querying Events" on page 476.

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.

Example: Defining and Updating an Administrative Command Schedule

Classic Scheduling

To schedule the backup of the ARCHIVEPOOL primary storage pool periodically, use classic scheduling:

```
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:

```
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. When you copy a schedule, Tivoli Storage Manager copies the following information:

- A description of the schedule
- All parameter values from the original schedule

You can then update the new schedule to meet your needs.

To copy the BACKUP_ARCHIVEPOOL administrative schedule and name the new schedule BCKSCHED, enter:

```
copy schedule backup_archivepool bcksched type=administrative
```

Deleting Schedules

To delete the administrative schedule ENGBKUP, enter:

```
delete schedule engbkup type=administrative
```

Managing Scheduled Event Records

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

Each scheduled administrative command 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 processing of a scheduled command is created or missed.

Querying Events

To help manage schedules for administrative commands, you can request information about scheduled and completed events. You can request general or exception reporting queries.

- To get information about past and projected scheduled processes, use a general query. If the time range you specify includes the future, the query output shows which events should occur in the future based on current schedules.
- To get information about scheduled processes that did not complete successfully, use exception reporting.

To minimize the processing time when querying events, minimize the time range.

To query an event for an administrative command schedule, you must specify the TYPE=ADMINISTRATIVE parameter. Figure 62 on page 477 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 62. 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.

To set the retention period to 15 days, enter:

```
set eventretention 15
```

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

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.

To delete all event records written prior to 11:59 p.m. on June 30, 2002, enter:

```
delete event type=administrative 06/30/2002 23:59
```

IBM Tivoli Storage Manager Server Scripts

Tivoli Storage Manager provides for automation of common administrative tasks with server scripts that are stored in the database. The scripts can be processed directly on the server console, the Administration Center, or included in an administrative command schedule. Tivoli Storage Manager provides sample scripts in *scripts.smp.* The sample scripts have an example order of execution for scheduling administrative commands. For more information, see "Using SELECT Commands in IBM Tivoli Storage Manager Scripts" on page 521. The sample scripts can be loaded from the *scripts.smp* file by issuing the DSMSERV RUNFILE command. Refer to *Installation Guide* for details.

The administrator can run the script from the Administration Center, or scheduling the script for processing using the administrative command scheduler on the server. If one of the specified commands in the script does not process successfully, the remaining commands are not processed.

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

Task	Required Privilege Class
Define a server script	System, policy, storage, and operator

You can define a server script line by line, create a file that contains the command lines, or copy an existing script.

The following examples use commands to define and update scripts. However, you can easily define and update scripts using the Administration Center where you can also use local workstation cut and paste functions.

Note: The Administration Center only supports ASCII characters for input. If you need to enter characters that are not ASCII, do not use the Administration Center. Issue the DEFINE SCRIPT and UPDATE SCRIPT commands from the server console.

You can 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. Refer to *Administrator's Reference* for information.

You can use the ISSUE MESSAGE command to determine where a problem is within a command in a script. Refer to *Administrator's Reference* for information on how to use the ISSUE MESSAGE command.

For additional information about updating server scripts, or updating a command line, see "Updating a Script" on page 481.

Defining a Server Script Using Contents of Another File

You can define a script whose command lines are read in from another file that contains statements for the script to be defined. For example, to define a script whose command lines are read in from the file BKUP12.MAC, issue:

```
define script admin1 file=bkup12.mac
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.

Running Commands in Parallel or Serially

You have the options of running commands serially, in parallel, or serially and in parallel. You can run multiple commands in parallel and wait for them to complete before proceeding to the next command. Commands will run serially until the parallel command is encountered. Refer to the *Administrator's Reference* for more information on the PARALLEL and SERIAL script commands.

The following example illustrates how the parallel command is used to backup, migrate and reclaim storage pools.

```
/*run multiple commands in parallel and wait for
them to complete before proceeding*/
PARALLEL
/*back up four storage pools simultaneously*/
BACKUP STGPOOL PRIMPOOL1 COPYPOOL1 WAIT=YES
BACKUP STGPOOL PRIMPOOL2 COPYPOOL2 WAIT=YES
BACKUP STGPOOL PRIMPOOL3 COPYPOOL3 WAIT=YES
BACKUP STGPOOL PRIMPOOL4 COPYPOOL4 WAIT=YES
/*wait for all previous commands to finish*/
SERIAL
/*after the backups complete, migrate stgpools
simultaneously*/
PARALLEL
MIGRATE STGPOOL PRIMPOOL1 DURATION=90 WAIT=YES
MIGRATE STGPOOL PRIMPOOL2 DURATION=90 WAIT=YES
MIGRATE STGPOOL PRIMPOOL3 DURATION=90 WAIT=YES
MIGRATE STGPOOL PRIMPOOL4 DURATION=90 WAIT=YES
/*wait for all previous commands to finish*/
SERIAL
/*after migration completes, 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. The following example continues an SQL statement across multiple command lines:

```
/*-----------------------------*/
/* Sample continuation example */
SELECT-
* FROM-
NODE WHERE-
PLATFORM='win32'
```

When this command is processed, it runs the following:

```
select * from nodes where platform='win32'
```

Using Substitution Variables

You can include substitution variables in a script. Substitution variables are specified with a $ character followed by a number that represents the position of the parameter when the script is processed. The following example SQLSAMPLE script specifies substitution variables $1 and $2:

```
/*---------------------------------------------*/
/* Sample substitution example */
/* ---------------------------------------------*/
SELECT-
$1 FROM-
NODES WHERE-
PLATFORM='$2'
```

When you run the script you must specify two values, one for $1 and one for $2. For example:

```
run sqlsample node_name aix
```

The command that is processed when the SQLSAMPLE script is run is:

```
select node_name from nodes where platform='aix'
```

Using Logic Flow Statements in a Script

You can use conditional logic flow statements based on return codes issued from previous command processing. These logic statements allow you to process your scripts based on the outcome of certain commands. You can use IF, EXIT, or GOTO (label) statements.

As each command is processed in a script, the return code is saved for possible evaluation before the next command is processed. The return code can be one of three severities: OK, WARNING, or ERROR. Refer to *Administrator's Reference* for a list of valid return codes and severity levels.

Specifying the IF Clause: You can use the IF clause at the beginning of a command line to determine how processing of the script should proceed based on the current return code value. In the IF clause you specify a return code symbolic value or severity.

The server initially sets the return code at the beginning of the script to RC_OK. The return code is updated by each processed command. If the current return code from the processed command is equal to any of the return codes or severities in the IF clause, the remainder of the line is processed. If the current return code is not equal to one of the listed values, the line is skipped.

The following script example backs up the BACKUPPOOL storage pool only if there are no sessions currently accessing the server. The backup proceeds only if a return code of RC_NOTFOUND is received:

```
/* Backup storage pools if clients are not accessing the server */
select * from sessions
/* There are no sessions if rc_notfound is received */
if(rc_notfound) backup stg backuppool copypool
```

The following script example backs up the BACKUPPOOL storage pool if a return code with a severity of warning is encountered:

```
/* Backup storage pools if clients are not accessing the server */
select * from sessions
/* There are no sessions if rc_notfound is received */
if(warning) backup stg backuppool copypool
```

Specifying the EXIT Statement: The EXIT statement ends script processing. The following example uses the IF clause together with RC_OK to determine if clients are accessing the server. If a RC_OK return code is received, this indicates that client sessions are accessing the server. The script proceeds with the exit statement, and the backup does not start.

```
/* Back up storage pools if clients are not accessing the server */
select * from sessions
/* There are sessions if rc_ok is received */
if(rc_ok) exit
backup stg backuppool copypool
```

Specifying the GOTO Statement: The GOTO statement is used in conjunction with a label statement. The label statement is the target of the GOTO statement. The GOTO statement directs script processing to the line that contains the label statement to resume processing from that point. The label statement always has a colon (:) after it and may be blank after the colon.

The following example uses the GOTO statement to back up the storage pool only if there are no sessions currently accessing the server. In this example, the return code of RC_OK indicates that clients are accessing the server. The GOTO statement directs processing to the **done:** label which contains the EXIT statement that ends the script processing:

```
/* Back up storage pools if clients are not accessing the server */
select * from sessions
/* There are sessions if rc_ok is received */
if(rc_ok) goto done
backup stg backuppool copypool
done:exit
```

Managing Server Scripts

You can update, copy, rename, query, delete, and run server scripts.

Task	Required Privilege Class
Update, copy, rename, query, and delete a script	System, policy, storage, and operator
Run a script	System, policy, storage, and operator

Updating a Script

You can update a script to change an existing command line or to add a new command line to a script.

To change an existing command line, specify the LINE= parameter.

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.

Appending a New Command: 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: 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: To add the SET REGISTRATION OPEN command as the new line 007 in the QSTATUS script, issue the following:

```
update script qstatus "set registration open" line=7
```

The QSTATUS script is updated to the following:

```
001 /* This is the QSTATUS script */
005 QUERY STATUS
007 SET REGISTRATION OPEN
010 QUERY STGPOOL
015 QUERY SESSION
```

Copying a Server Script

You can copy an existing script to a new script with a different name. For example, to copy the QSTATUS script to QUERY1 script, issue:

```
copy script qstatus query1
```

The QUERY1 command script now contains the same command lines as the QSTATUS command script.

Querying a Server Script

You can query a script to display information about the script. You can specify wildcard characters to display all scripts with names that match a particular pattern. When you query a script, you can direct the output to a file in a file system that the server can access. The various formats you can use to query scripts are as follows:

Format	Description
Standard	Displays the script name and description. This is the default.

Format	Description
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.
Raw	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.

The following is an example for querying a script in the standard format.

```
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=RAW and OUTPUTFILE parameters. You can use the resulting output as input into another script without having to create a script line by line.

The following is an example of querying the SRTL2 script in the raw format, 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. For example, to rename the QUERY1 script to QUERY5, issue:

```
rename script query1 query5
```

The QUERY1 script is now named QUERY5.

Deleting a Command from a Server Script

You can delete an individual command line from a script. When you specify a line number, only the corresponding command line is deleted from the script.

For example, to delete the 007 command line from the QSTATUS script, issue:

```
delete script qstatus line=7
```

Deleting a Server Script

To delete an entire script, issue the DELETE SCRIPT command.

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.

You can preview the command lines of a script without actually executing the commands by using the PREVIEW=YES parameter with the RUN command. If the script contains substitution variables, the command lines are displayed with the substituted variables. This is useful for evaluating a script before you run it.

For example, to process the QAIXC script previously defined, issue:

```
run qaixc
```

To process the following script that contains substitution variables:

```
/*----------------------------------------------*/
/* Sample continuation and substitution example */
/* ---------------------------------------------*/
SELECT-
$1 FROM-
NODES WHERE-
PLATFORM='$2'
```

Enter:

```
run qaixc node_name aix
```

Where $1 is node_name and $2 is aix.

Using Macros

Tivoli Storage Manager supports macros on the administrative client. A macro is a file that contains one or more administrative client commands. You can only run a macro from the administrative client in batch or interactive modes. Macros are stored as a file on the administrative client. Macros are not distributed across servers and cannot be scheduled on the server.

Macros can include the following:

- Administrative commands

  For more information on administrative commands, see “Writing Commands in a Macro” on page 485.
- Comments

  For more information on comments, see “Writing Comments in a Macro” on page 485.
- Continuation characters

  For more information on continuation characters, see “Using Continuation Characters” on page 486.
- Variables

  For more information on variables, see “Using Substitution Variables in a Macro” on page 486.

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 "Controlling Command Processing in a Macro" on page 487.

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

Refer to the *Administrator's Reference* for more information on how commands are entered and the general rules for entering administrative commands. The administrative client ignores any blank lines included in your macro. However, a completely blank line terminates a command that is continued (with a continuation character).

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

```
register admin pease mypasswd -
  contact='david pease, x1234'
grant authority pease -
  classes=policy,storage -
  domains=domain1,domain2 -
  stgpools=stgpool1,stgpool2
```

This example uses continuation characters in the macro file. For more information on continuation characters, see "Using Continuation Characters" on page 486.

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.

Writing Comments in a Macro

You can add comments to your macro file. To write a comment:

- Write a slash and an asterisk (/*) to indicate the beginning of the comment.
- Write the comment.
- Write an asterisk and a slash (*/) to indicate the end of the comment.

You can put a comment on a line by itself, or you can put it on a line that contains a command or part of a command.

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

```
/* auth.mac-register new nodes  */
```

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

```
domain=domain1            /*assign node to domain1  */
```

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

Using Continuation Characters

You can use continuation characters in a macro file. Continuation characters are useful when you want to execute a command that is longer than your screen or window width.

Attention: Without continuation characters, you can enter up to 256 characters. With continuation characters, you can enter up to 1500 characters. In the MACRO command, these maximums are *after* any substitution variables have been applied (see "Using Substitution Variables in a Macro").

To use a continuation character, enter a dash or a back slash at the end of the line that you want to continue. With continuation characters, you can do the following:

- Continue a command. For example:

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

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

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

- Continue a string of values enclosed in quotation marks by entering the first part of the string enclosed in quotation marks, followed by a dash or a back slash at the end of the line. Then, enter the remainder of the string on the next line enclosed in the *same* type of quotation marks. For example:

  ```
  contact="david pease, bldg. 100, room 2b, san jose,"-
  "ext. 1234, alternate contact-norm pass,ext 2345"
  ```

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

Using Substitution Variables in a Macro

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

A substitution variable consists of a percent sign (%), followed by a number that indicates the number of the substitution variable. When you run the file with the MACRO command, you must specify values for the variables.

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

```
/* register new nodes */
register node %1 %2 -      /*  userid password                  */
  contact=%3 -             /* 'name, phone number'              */
  domain=%4                /*  policy domain                    */
```

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

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

```
macro auth.mac david davidpw "david pease, x1234" domain1
```

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

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

Running a Macro

Use the MACRO command when you want to run a macro. You can enter the MACRO command in batch or interactive mode.

If the macro does not contain substitution variables (such as the REG.MAC macro described in the "Writing Commands in a Macro" on page 485), 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 486), 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
```

Controlling Command Processing in a Macro

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

If you specify the ITEMCOMMIT option when you enter the DSMADMC command, the server commits each command in a 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 465.

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 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 database does not store client data. Instead, the database points to the locations of the client files in the storage pools.

The database includes information about:

- Client nodes and administrators
- Policies and schedules
- Server settings
- Locations of client files on server storage
- Server operations (for example, activity logs and event records)

Note: If the database is unusable, the entire Tivoli Storage Manager server is unavailable. If a database is lost and cannot be recovered, the backup, archive, and space-managed data for that server is lost. See Chapter 23, "Protecting and Recovering Your Server," on page 639 for steps that you can take to protect your database.

The recovery log contains information about database updates that have not yet been committed. Updates can include activities such as defining a management class, backing up a client file, and registering a client node. Changes to the database are recorded in the recovery log to maintain a consistent database image.

The following shows authority requirements for tasks in this chapter:

Task	Required Privilege Class
Manage disk volumes used by the database and recovery log	System or unrestricted storage
Display information about the database and recovery log	Any administrator

See the following sections:

Concepts:
"How IBM Tivoli Storage Manager Processes Transactions" on page 490
"How IBM Tivoli Storage Manager Manages Space" on page 492

Tasks:
"Estimating and Monitoring Database and Recovery Log Space Requirements" on page 494
"Increasing the Size of the Database or Recovery Log" on page 496
"Decreasing the Size of the Database or Recovery Log" on page 501
"Optimizing Database and Recovery Log Performance" on page 503

© Copyright IBM Corp. 1993, 2005

Note: Mirroring of the database and recovery log is described in the chapter on data protection. See "Mirroring the Database and Recovery Log" on page 644.

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

How IBM Tivoli Storage Manager Processes Transactions

To support multiple transactions from concurrent client sessions, the server holds transaction log records in the recovery log buffer pool until they can be written to the recovery log. These records remain in the buffer pool until the active buffer becomes full or Tivoli Storage Manager forces log records to the recovery log. The log records for a given transaction are forced into stable storage (the recovery log volume-disk) when the transaction is committed.

Changes resulting from transactions are held in the buffer pool temporarily and are not made to the database immediately. The database information stored on disk remains consistent because the server ensures that the recovery log records representing the updates to these database pages is written to disk. This allows the server to manage when it writes changed database pages from the buffer pool out to the actual database volumes. During restart-recovery, the server uses the recovery 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 insures that the necessary redo and undo information is available to replay these transaction changes against the database information. At some point after a transaction is committed, the server deletes the transaction record from the recovery log.

Performance Considerations: Transferring Files as a Group between Client and Server

A transaction is the unit of work exchanged between the client and server. The client program can transfer more than one file or directory between the client and server before it commits the data to server storage. Therefore, a transaction can contain more than one file or directory. This is called a *transaction group*. Tivoli Storage Manager provides a TXNGROUPMAX server option that allows you to specify the number of files or directories contained within a transaction group.

It is possible to affect the performance of client backup, archive, restore, and retrieve operations by using a larger value for the TXNGROUPMAX option. You can use the TXNGROUPMAX option to increase performance when Tivoli Storage Manager writes to tape. This performance can be considerable when a user transfers multiple small files.

If you increase the value of TXNGROUPMAX by a large amount, monitor the effects on the recovery log. A larger value can increase utilization of the recovery log, as well as an increased length of time for a transaction to commit. Another

thing to consider along with TXNGROUPMAX is the number of concurrent sessions that will be run. It may be possible to run with a higher TXNGROUPMAX value with a few clients running. However, if there are hundreds of clients running concurrently, they may need to reduce the TXNGROUPMAX to help manage the recovery log usage to support this high number of concurrent clients. If the effects are severe enough, they may impact server operations. See "Monitoring the Database and Recovery Log" on page 495 for more information.

The following are examples of how the TXNGROUPMAX option can affect performance throughput for operations to tape and the recovery log. The maximum number of concurrent client/server sessions is defined by the MAXSESSIONS server option.

- The TXNGROUPMAX option is set to 20. The MAXSESSIONS option 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=1,000
  ```

  This represents 1,000 log records in the recovery log. Each time a transaction ends (commits), the server can free 200 of those log records. Over time and as transactions end, the recovery log can release the space used by the oldest transactions. These transactions complete and the log progresses forward.
- The TXNGROUPMAX option is set to 2,000. The MAXSESSIONS option 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:

  ```
  2,000*10*5=20,000
  ```

  This represents 100,000 log records in the recovery log. Each time a transaction ends (commits), the server can free 20,000 of those log records. Over time and as transactions end, the recovery log can release the space used by the oldest transactions. These transactions complete and the log progresses forward.

Based on the previous two examples, five concurrent transactions with a TXNGROUPMAX setting of 2,000 consume significantly 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 is a comparison of the two example TXNGROUPMAX settings. This example becomes more significant if we include an example that a given log record takes 100 bytes.

Table 33. Example of log bytes consumed by five concurrent sessions

TXNGROUPMAX Setting	Number of Log Bytes Consumed
TXNGROUPMAX=20	100,000
TXNGROUPMAX=2,000	10,000,000

There are several related server options that can be used to tune server performance and reduce the risk of running out of recovery log space:

- The THROUGHPUTTIMETHRESHOLD and THROUGHPUTDATATHRESHOLD options should be used in conjunction with the TXNGROUPMAX option to prevent a slower performing node from holding a transaction open for extended periods.

- If the database is in roll-forward mode, lowering the database backup trigger can reduce the risk of running out of recovery log space when a larger TXNGROUPMAX value is specified. For more information, see "Automating Database Backups" on page 655.
- Increasing the size of the recovery log is another step to consider when increasing the TXNGROUPMAX setting. For more information, see "Increasing the Size of the Database or Recovery Log" on page 496.

Evaluate each node's performance and characteristics before increasing the TXNGROUPMAX setting. Nodes that only have a few larger objects to transfer will not benefit as much as nodes that have multiple, smaller objects to transfer. For example, a file server would benefit more from a higher TXNGROUPMAX setting than would a database server that had 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 frequently perform high log usage operations. The raw or physical performance of the disk drives holding the database and recovery log will possibly become an issue with an increased TXNGROUPMAX setting. The drives need to handle higher transfer rates in order to handle the increased load on the recovery log and database.

The TXNGROUPMAX option can be set as a global server option value, or it can be set individually for a node. Refer to the REGISTER NODE command and the server options in the *Administrator's Reference*. We recommended that you specify a conservative TXNGROUPMAX value (between 4 and 512). Select higher values for individual nodes that will benefit from the increased transaction size.

How IBM Tivoli Storage Manager Manages Space

Tivoli Storage Manager tracks all volumes defined to the database as one logical volume and all volumes defined to the recovery log as another logical volume. In Figure 63, the database consists of four volumes: VOL1 through VOL4, which the server tracks as a single logical volume.

Figure 63. A Server Database

To manage the database and recovery log effectively, you must understand the following concepts:

- Available space
- Assigned capacity
- Utilization

Available Space

Not all of the space that is allocated for the volumes of the database or of the recovery log can be used for database and recovery log information. The server subtracts 1MB from each physical volume for overhead. The remaining space is divided into 4MB partitions. For example, you allocate four 25MB volumes for the

database. For the four volumes, 4MB are needed for overhead leaving 96MB of available space as shown in figure Figure 64:

Figure 64. An Example of Available Space

Assigned Capacity

Assigned capacity is the available space that can be used for database or recovery log information. During installation, the assigned capacities of the database and recovery log match the available space. If you add volumes after installation, you increase your available space. However, to increase the assigned capacity, you must also extend the database or recovery log. See "Step 2: Extending the Capacity of the Database or Recovery Log" on page 500 for details.

Utilization

Utilization is the percent of the assigned capacity in use at a specific time. *Maximum percent utilized* is the highest utilization since the statistics were reset. For example, an installation performs most backups after midnight. Figure 65 shows that utilization statistics for the recovery log were reset at 9 p.m. the previous evening and that the maximum utilization occurred at 12 a.m.

Figure 65. An Example of Recovery Log Utilization

Unless many objects are deleted, the database maximum percent utilized is usually close to the utilization percentage.

Estimating and Monitoring Database and Recovery Log Space Requirements

The size of the database depends on the number of client files to be stored and the method by which the server manages them. If you can estimate the maximum number of files that might be in server storage at any time, you can estimate the database size from the following information:

- Each stored **version of a file** requires about 400 to 600 bytes of database space.
- Each **cached** or **copy storage pool** file requires about 100 to 200 bytes of database space.
- **Overhead** could require up to 25% in additional space.

In the example below, the computations are probable maximums. In addition, the numbers are not based on the use of file aggregation. In general, aggregation of small files reduces the required database space. For details about aggregation, see "How the Server Groups Files before Storing" on page 234. Assume the following numbers for a Tivoli Storage Manager system:

Versions of files

Backed up files
Up to 500,000 client files might be backed up. Storage policies call for keeping up to 3 copies of backed up files:

```
500,000 files x 3 copies = 1,500,000 files
```

Archived files
Up to 100,000 files might be archived copies of client files.

Space-managed files
Up to 200,000 files migrated from client workstations might be in server storage.

Note: File aggregation does not affect space-managed files.

At 600 bytes per file, the space required for these files is:

```
(1,500,000 + 100,000 + 200,000) x 600 = 1.0GB
```

Cached and copy storage pool files

Cached copies
Caching is enabled in a 5GB disk storage pool. The pool's high and low migration thresholds are 90% and 70% respectively. Thus, 20% of the disk pool, or 1GB, is occupied by cached files.

If the average file size is about 10KB, about 100,000 files are in cache at any one time.

```
100,000 files x 200 bytes = 19MB
```

Copy storage pool files
All primary storage pools are backed up to the copy storage pool:

```
(1,500,000 + 100,000 + 200,000) x 200 bytes = 343MB
```

Therefore, cached files and copy storage pool files require about 0.4GB of database space.

Overhead
About 1.4GB is required for file versions and cached and copy storage pool files. Up to 50% additional space (or 0.7GB) should be allowed for overhead.

The database should then be approximately 2.1GB.

If you cannot estimate the numbers of files, you can roughly estimate the database size as from 1% to 5% of the required server storage space. For example, if you need 100GB of server storage, your database should be between 1GB and 5GB. See "Estimating Space Needs for Storage Pools" on page 277 for details.

During SQL queries of the server, intermediate results are stored in temporary tables that require space in the free portion of the database. Therefore, the use of SQL queries requires additional database space. The more complicated the queries, the greater is the space required.

The size of the recovery log depends on the number of concurrent client sessions and the number of background processes executing on the server. The maximum number of concurrent client sessions is set in the server options.

Attention: Be aware that the results are estimates. The actual size of the database may differ from the estimate because of factors, such as, the number of directories and the length of the path and file names. You should periodically monitor your database and recovery log and adjust their sizes as necessary.

Begin with at least a 12MB recovery log. If you use the database backup and recovery functions in roll-forward mode, you should begin with at least 25MB. See "Database and Recovery Log Protection" on page 642 and "Estimating the Size of the Recovery Log" on page 653 for more information.

Monitoring the Database and Recovery Log

You should regularly monitor the database and recovery log to see if you should add or delete space. To monitor daily utilization, you might want to reset the maximum utilization counters each day. Utilization statistics are reset in two ways:

- Automatically when the server is restarted
- By issuing the RESET DBMAXUTILIZATION or RESET LOGMAXUTILIZATION commands

For example, to reset the database utilization statistic, enter:

```
reset dbmaxutilization
```

If the SELFTUNEBUFPOOLSIZE server option is in effect, the buffer pool cache hit ratio statistics are reset at the start of expiration. After expiration, the buffer pool size is increased if the cache hit ratio is less than 98%. The increase in the buffer pool size is in small increments and may change after each expiration. The change in the buffer pool size is not reflected in the server options file. You can check the current size at any time using the QUERY STATUS command. Use the SETOPT BUFPOOLSIZE command to change the buffer pool size.

To display information about the database or recovery log, issue the QUERY DB or QUERY LOG command. For example:

```
query db
```

The server displays a report, like this:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
       96       96         0        92   4,096    24,576        86   0.3   0.3
```

To display information about the database or recovery log volumes, issue the QUERY DBVOLUME or the QUERY LOGVOLUME command. For example:

```
query dbvolume db1.dsm format=detailed
```

The server displays a report, like this:

```
Volume Name (Copy 1): /home/bill/dsmserv/build/db1
         Copy Status: Sync'd
Volume Name (Copy 2):
         Copy Status: Undefined
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 12
Allocated Space (MB): 12
     Free Space (MB): 0
```

Note: Tivoli Storage Manager displays output from this command from the lowest to the highest number. If a volume is deleted, Tivoli Storage Manager reuses that volume number the next time that a volume is defined. A query can then display volumes that are not in numerical sequence. You can reset the order of your database or recovery log volumes by specifying the desired order with the DSMSERV LOADFORMAT command.

See the indicated sections for details about the following entries:

- Available space, “Available Space” on page 492
- Assigned capacity, “Assigned Capacity” on page 493
- Utilization and maximum utilization, “Utilization” on page 493

If utilization is high, you may want to add space (see “Increasing the Size of the Database or Recovery Log”). If utilization is low, you may want to delete space (see “Decreasing the Size of the Database or Recovery Log” on page 501).

Note: You can also use a DEFINE SPACETRIGGER command to automatically check whether the database or recovery log exceeds a utilization percentage that you specify. See “Automating the Increase of the Database or Recovery Log” on page 497 for details.

Increasing the Size of the Database or Recovery Log

As your requirements change, you can increase or decrease the sizes of the database and recovery log. You can automate the process of increasing the sizes, or you can perform all the steps manually. See “Automating the Increase of the Database or Recovery Log” on page 497 or “Manually Increasing the Database or Recovery Log” on page 499.

Attention: Do not change the size of an allocated database or recovery log volume after it has been defined. If you change the size of a volume through the use of utilities outside of Tivoli Storage Manager, Tivoli Storage Manager may not initialize correctly, and data may be lost.

Note: Significantly increasing the recovery log size could significantly increase the time required to restart the server, back up the database, and restore the database.

Automating the Increase of the Database or Recovery Log

You can automate the process of increasing the database and recovery log sizes. With a DEFINE SPACETRIGGER command, you can specify the following:

- Utilization percentages at which the database or recovery log size is to be increased
- The size of the increase as a percentage of the current database or recovery log size
- The prefix to be used for a new volume
- The prefix or prefixes to be used for new files that are created for database or recovery log copy volumes when the volumes are mirrored
- The maximum size allowed for the database or recovery log

For example, assume that you have a 100GB database and a 3GB recovery log. You want to increase the database size by 25 percent when 85 percent is in use, but not to more than 200GB. When this utilization percentage is reached, you want copy volumes created in a separate directory. You also want to increase the recovery log size by 30 percent when 75 percent is in use, but not to more than 5GB.

Note: There is one time when the database or recovery log might exceed the maximum size specified: If the database or recovery log is less than the maximum size when expansion begins, it continues to the full expansion value. However, no further expansion will occur unless the space trigger is updated.

To add the new volumes to the /opt/tivoli/tsm/server/bin/ directory and create copy volumes in the k:/mirvol1 and j:/mirvol2 directory, issue the following commands:

```
define spacetrigger db fullpct=85 spaceexpansion=25
expansionprefix=/opt/tivoli/tsm/server/bin/
mirrorprefixes="k:/mirvol1/,j:/mirvol2/" maximumsize=200000

define spacetrigger log fullpct=75 spaceexpansion=30
expansionprefix=/opt/tivoli/tsm/server/bin/ maximumsize=50000
```

Note: To specify a directory as your prefix and allow the server to create files in that directory, use a forward-slash at the end of the directory you are specifying.

The server then monitors the database or recovery log and, if the utilization level is reached, does the following:

- Displays a message (ANR4413I or ANR4414I) that states the amount of space required to meet the utilization parameter specified in the command.
- Allocates space for the new volume.
- Defines the new volume.
- Extends the database or recovery log.

- If a volume is mirrored and there is enough disk space, the preceding steps are also performed for the mirrored copies.

Notes:

1. The maximum size of the recovery log is 13GB. The server will not automatically extend the recovery log beyond 12GB.
2. An automatic expansion may exceed the specified database or recovery log maximum size but not the 13GB recovery log limit. However, after the maximum has been reached, no further automatic expansions will occur.
3. A space trigger percentage may be exceeded between the monitoring of the database or recovery log and the time that a new volume is brought online.
4. If the server creates a database or recovery log volume and the attempt to add it to the server fails, the volume is not deleted. After the problem is corrected, you can define it with the DEFINE DBVOLUME or DEFINE LOGVOLUME command.
5. Only one or two copy volumes can be created per database or recovery log space trigger. Therefore, only one or two prefixes can be specified. If only one prefix is specified and two copy volumes are created, that single prefix will be used for both copy volumes. If two prefixes are specified, each prefix will be used in creating a single copy volume.
6. Automatic expansion will not occur during a database backup.
7. The database and recovery log utilization percentage may exceed the space trigger value. The server checks utilization after a database or recovery log commit.

   Also, deleting database volumes and reducing the database does not activate the trigger. Therefore, the utilization percentage can exceed the set value before new volumes are online.
8. The database and the recovery log size may exceed the specified MAXIMUMSIZE value. This value is a threshold for expansion. Tivoli Storage Manager checks the size and allows expansion if the database or the recovery log is less than the maximum size. Tivoli Storage Manager will not automatically expand the database or the recovery log if either is greater than the maximum size. However, Tivoli Storage Manager only checks the size that results after expansion to ensure that the maximum recovery log size is not exceeded.

Recovering When the Recovery Log Runs Out of Space

If the log mode is set to ROLLFORWARD and either the recovery log is too small or the database backup trigger is set too high, the recovery log could run out of space before database operations complete. If this happens, you may need to stop the server without enough recovery log space to restart the server. In some cases, the server halts itself.

To restart the server, first format a new volume (see "Using the DSMFMT Command to Format Volumes" on page 500). Then use the DSMSERV EXTEND LOG command to extend the size of the recovery log. For example, after formatting a 21MB volume named *new.reclog*, extend the recovery log by issuing the following command:

```
dsmserv extend log new.reclog 20
```

After the server is running, you can do the following:

- Back up the database, which frees the recovery log space
- Adjust the size of the recovery log, the database backup trigger, or both

Manually Increasing the Database or Recovery Log

To add space to the database or recovery log, do the following:

"Step 1: Creating Database and Recovery Log Volumes"
"Step 2: Extending the Capacity of the Database or Recovery Log" on page 500

Step 1: Creating Database and Recovery Log Volumes

You can allocate space and define a database or recovery log volume in a single operation. For example, to allocate a 100MB database volume named VOL5 in the /opt/tivoli/tsm/server/bin directory and define the volume, enter:

```
define dbvolume /opt/tivoli/tsm/server/bin/ formatsize=100
```

The available space of the database increases to 196MB, but the assigned capacity remains at 96MB. For Tivoli Storage Manager to use the space, you must extend the capacity (see "Step 2: Extending the Capacity of the Database or Recovery Log" on page 500). To verify the change, query the database or recovery log. For example, to query the database, enter:

```
query db
```

The server displays a report, like this:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
      196       96       100        92   4,096    24,576        86   0.3   0.3
```

The value in the *Maximum Extension* field should equal the available space of the new volume. In this example, a 101MB volume was allocated. This report shows that the available space has increased by 100MB; the assigned capacity is unchanged at 96MB; and the maximum extension is 100MB. Figure 66 illustrates these changes.

Figure 66. Adding Volumes Increases Available Space

You can also query the database and recovery log volumes to display information about the physical volumes that make up the database and recovery log.

Notes:

1. The maximum size of the recovery log is 13GB, and the maximum size of the database is 530GB. If you allocate a volume that would cause the recovery log

or database to exceed these limits, the subsequent DEFINE DBVOLUME or DEFINE LOGVOLUME command for the volume will fail.

2. For performance reasons, define more than one volume for the database and recovery log, and put these volumes on separate disks. This allows simultaneous access to different parts of the database or recovery log.
3. To use disk space efficiently, allocate a few large disk volumes rather than many small disk volumes. In this way, you avoid losing space to overhead processing.

   If you already have a number of small volumes and want to consolidate the space into one large volume, see "Decreasing the Size of the Database or Recovery Log" on page 501.
4. To protect database and recovery log volumes from media failure, use mirroring. See "Mirroring the Database and Recovery Log" on page 644 for details.

Using the DSMFMT Command to Format Volumes: You can still use the DSMFMT utility to allocate a database or recovery log volume. You would then issue the DEFINE DBVOLUME or DEFINE LOGVOLUME command without the FORMATSIZE parameter, and extend the database or recovery log (see "Step 2: Extending the Capacity of the Database or Recovery Log").

To allocate an additional 101MB to the database as volume VOL5, enter:

```
> dsmfmt -db vol5 101
```

Step 2: Extending the Capacity of the Database or Recovery Log

The database and recovery log are extended in 4MB increments. If you do not specify the extension in 4MB increments, the server rounds up to the next 4MB partition. For example, if you specify 1MB, the server extends the capacity by 4MB.

To increase the capacity of the database by 100MB, enter:

```
extend db 100
```

After the database has been extended, the available space and assigned capacity are both equal to 196MB, as shown in Figure 67.

Figure 67. Extending the Capacity of the Database

You can query the database or recovery log (QUERY DB and QUERY LOG commands) to verify their assigned capacities. The server would display a report, like this:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
      196      196         0       192   4,096    50,176       111   0.2   0.2
```

Decreasing the Size of the Database or Recovery Log

You may want to delete database or recovery log volumes for a number of reasons. For example:

- You have a significant amount of space that is unused.
- You want to consolidate a number of small volumes, each of which may have unusable space, into one large volume. To create a volume, see "Increasing the Size of the Database or Recovery Log" on page 496.

When you delete a database or recovery log volume, Tivoli Storage Manager tries to move data from the volume being deleted to other physical volumes in the database or recovery log.

To delete space, perform the following steps:

1. Determine if you can delete one or more volumes ("Step 1: Determining If Volumes Can Be Deleted").
2. Reduce the capacity of the database or recovery log to free existing space ("Step 2: Reducing the Capacity of the Database or Recovery Log" on page 502).
3. Delete the volume ("Step 3: Deleting a Volume from the Database or Recovery Log" on page 502).

Step 1: Determining If Volumes Can Be Deleted

To determine if volumes can be deleted from the database or recovery log, check the volume sizes and the amount of unused space.

To check the sizes of the volumes in the database, enter:

```
query dbvolume format=detailed
```

The server displays the following type of information:

```
Volume Name (Copy 1): VOL1
         Copy Status: Sync'd
Volume Name (Copy 2):
         Copy Status: Undefined
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 24
Allocated Space (MB): 24
     Free Space (MB): 0
```

In this example, VOL1, VOL2, VOL3, and VOL4 each have 24MB of available space, and VOL5 has 100MB. To determine if there is enough unused space to delete one or more volumes, enter:

```
query db
```

The server displays the following type of report.

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
      196      196         0       176   4,096    50,176     4,755   9.5   9.5
```

The *Maximum Reduction* field shows the assigned capacity not in use. In this example, you could reduce the database by up to 176MB. This is enough space to allow the deletion of VOL1, VOL2, VOL3, and VOL4.

If there is not enough space on the remaining volumes, allocate more space and define an additional volume. See "Increasing the Size of the Database or Recovery Log" on page 496. Continue with "Step 2: Reducing the Capacity of the Database or Recovery Log."

Step 2: Reducing the Capacity of the Database or Recovery Log

The database or recovery log capacity is reduced in 4MB increments. For example, based on the utilization of the database assume that VOL5 alone could contain all the data. To reduce the database by the amount of available space in VOL1 through VOL4, 96MB, enter:

```
reduce db 96
```

Reducing capacity is run as a background process and can take a long time. Issue a QUERY PROCESS command to check on the status of the process.

After reducing the database by 96MB, the assigned capacity is 100MB, and the maximum extension is 96MB, as shown in the following example:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
      196      100        96        92   4,096    24,576        86   0.3   0.3
```

Step 3: Deleting a Volume from the Database or Recovery Log

After you reduce the database or recovery log, use the smaller size for a few days. If the maximum utilization does not exceed 70%, you can delete the extra volumes.

Note: You cannot delete volumes if there is not enough free space for the server to move data from the volume being deleted to other physical volumes in the database or recovery log.

In our example, you determined that you can delete the four 24MB volumes from the database. You have reduced the database by 96MB. To delete VOL1 through VOL4 from the database, enter:

```
delete dbvolume vol1
delete dbvolume vol2
delete dbvolume vol3
delete dbvolume vol4
```

The server moves data from the volumes being deleted to available space on other volumes, as shown in Figure 68 on page 503.

Figure 68. Deleting Database Volumes

After the data has been moved, these volumes are deleted from the server.

Optimizing Database and Recovery Log Performance

Over time, the database size and organization can change to the point that performance is not optimal. Unloading and reloading the database can have the following benefits:

- Reduce utilized space in the database
- Reorganization of fragmented page allocations
- Improved performance of long-running scans of the database

The database and recovery log buffer pool sizes can also affect performance. A larger database buffer pool can improve performance. A larger recovery log buffer pool reduces how often the server forces records to the recovery log.

See “Reorganizing the Database” on page 508 for more information about restoring database efficiency.

Optimal Database Volume Allocation

There are many ways to improve performance through strategic volume allocation. Although you can use a single, large Tivoli Storage Manager database, better performance can be achieved over time by using multiple volumes. This allows Tivoli Storage Manager to run more I/O requests to the database at any given time. If possible, these volumes should be placed either on different storage devices or on arrays for maximum performance. Some improvement can still be gained by placing multiple volumes on a single disk. However, performance improvements will not be as significant after four volumes are allocated per disk.

If possible, place database volumes on areas of storage separate from the disk storage pool volumes. This prevents the degradation of database performance due to the large amounts of sequential I/O that happens to a disk storage pool volume.

There is no benefit to allocating multiple recovery log volumes because Tivoli Storage Manager will only write to one at a time. However, there is a performance advantage to placing the log on different storage than the database or storage pool volumes.

Utilizing Hardware with Non-Volatile Write Cache

Performance can be increased significantly by using hardware that provides a fast, non-volatile write cache for the database and log volumes. When choosing disk storage hardware on which to store the Tivoli Storage Manager database and log, look for devices with low cache hit latencies. To maintain database integrity, ensure that the hardware chosen can withstand various types of failure, such as power outages and controller failure.

Adjusting the Database Buffer Pool Size

You can let Tivoli Storage Manager dynamically adjust the size of the database buffer pool or you can adjust it manually. If you specify YES for the SELFTUNEBUFPOOLSIZE server option, the database buffer pool is dynamically adjusted. The cache hit ratio statistics (cache hit percentages) for the buffer pool are reset at the beginning of expiration. After expiration processing completes, the buffer pool size is adjusted dynamically.

Server expiration processing resets the database buffer pool before the next processing starts and examines if the database buffer pool cache hit ratio is above 98%. If the cache hit ratio is lower than 98%, the database buffer pool will be increased; if it is higher, the buffer pool size will not change. Increasing the database buffer pool will not be more than 10% of available real storage.

Manually Adjusting the Database Buffer Pool Size

Perform the following steps to track the database buffer pool statistics and adjust the buffer pool size:

Step 1: Reset Database Buffer Pool Utilization Statistics

Reset the buffer pool statistics. Initially, you might want to reset the statistics twice a day. Later, you can reset them less often. To reset, enter:

```
reset bufpool
```

Step 2: Monitor the Database Buffer Pool

Once reset, let enough time to pass for a sizable amount of server activity to occur. Then, to see if the database buffer pool is adequate for database performance, enter:

```
query db format=detailed
```

The server displays a report, like this:

```
          Available Space (MB): 8
        Assigned Capacity (MB): 4
        Maximum Extension (MB): 4
        Maximum Reduction (MB): 0
             Page Size (bytes): 4,096
            Total Usable Pages: 1,024
                    Used Pages: 94
                      Pct Util: 9.2
                  Max. Pct Util: 9.2
              Physical Volumes: 3
             Buffer Pool Pages: 64
         Total Buffer Requests: 4,702
                Cache Hit Pct.: 97.68
               Cache Wait Pct.: 0.00
           Backup in Progress?: No
    Type of Backup In Progress:
  Incrementals Since Last Full: 0
Changed Since Last Backup (MB): 0.37
            Percentage Changed: 100.00
Last Complete Backup Date/Time: 09/15/04 15:51:21
Estimate of Recoverable Space (MB): 321
Last Estimate of Recoverable Space: 09/13/04 06:03:20
```

Use the following fields to evaluate your current use of the database buffer pool:

Buffer Pool Pages
The number of pages in the database buffer pool. This value is determined by the server option for the size of the database buffer pool. At installation, the database buffer pool is set to 2048KB, which equals 128 database pages.

Total Buffer Requests
The number of requests for database pages since the server was last started or the buffer pool was last reset. If you regularly reset the buffer pool, you can see trends over time.

Cache Hit Pct
The percentage of requests for cached database pages in the database buffer pool that were not read from disk. A high value indicates that the size of your database buffer pool is adequate. If the value falls below 98%, consider increasing the size of the database buffer pool. For larger installations, performance could improve significantly if your cache hit percentage is greater than 99%.

Cache Wait Pct
The percentage of requests for database pages that had to wait for a buffer to become available in the database buffer pool. When this value is greater than 0, increase the size of the database buffer pool.

Step 3: Adjust the Database Buffer Pool

If the cache hit ratio is not high enough, you can increase the BUFPOOLSIZE.

Note: If operating system paging is present, do no increase the BUFPOOLSIZE. Operating system paging can be observed by issuing the VMSTAT command and watching the **pi** (page in) and **po** (page out) columns.

Adjusting the Recovery Log Buffer Pool Size

Do the following to adjust the size of the recovery log buffer pool:

Step 1: Monitor the Recovery Log Buffer Pool

To see how the recovery log buffer pool size affects recovery log performance, enter:

```
query log format=detailed
```

The server displays a report, like this:

```
  Available Space (MB): 12
Assigned Capacity (MB): 12
Maximum Extension (MB): 0
Maximum Reduction (MB): 8
     Page Size (bytes): 4,096
           Total Pages: 3,072
            Used Pages: 227
                 %Util: 7.4
            Max. %Util: 69.6
      Physical Volumes: 1
        Log Pool Pages: 32
    Log Pool Pct. Util: 6.25
    Log Pool Pct. Wait: 0.00
```

Use the following fields to evaluate the log buffer pool size:

Log Pool Pages
: The number of pages in the recovery log buffer pool. This value is set by the server option for the size of the recovery log buffer pool. At installation, the default setting is 128KB, which equals 32 recovery log pages.

Log Pool Pct. Util
: The percentage of pages used to write changes to the recovery log after a transaction is committed. A value below 10% means that the recovery log buffer pool size is adequate. If the percentage increases, consider increasing the recovery log buffer pool size.

Log Pool Pct. Wait
: The percentage of requests for pages that are not available because all pages are waiting to write to the recovery log.

  If this value is greater than 0, increase the recovery log buffer pool size.

Step 2: Adjust the Recovery Log Buffer Pool

Use the LOGPOOLSIZE server option to set the size of the recovery log buffer pool.

Estimating Database Reorganization

Over time, the server database may experience fragmentation. Reorganizing the database can improve query performance. See “Reorganizing the Database” on page 508 for details.

You can use the ESTIMATE DBREORGSTATS command to estimate how much space can be recovered if a reorganization is performed. This allows you to determine if a reorganization will offer any space savings.

See the *Administrator’s Reference* for more information on the ESTIMATE DBREORGSTATS command.

To estimate the amount of recoverable space issue:

```
estimate dbreorgstats
```

The server displays the following information:

```
ANR1784I A database reorganization would reduce the database utilization by an
estimated 312 MB.
ANR097I Process 2 for ESTIMATE DBREORG running in BACKGROUND processed 4151 items
with a completion state of SUCCESS at 14:32:34.
ANR0381I Bufferpool statistics were successfully reset.
```

Requesting Information about the Estimate Database Reorganization Process

To monitor the progress of the ESTIMATE DBREORGSTATS command, enter:

```
query process
```

The following figure shows a server background process report after the ESTIMATE DBREORGSTATS command is issued. The report displays a process ID number, a description, and a completion status for each background process.

```
 Process Process Description      Status
  Number
-------- ------------------------ ---------------------------------------------
       2 ESTIMATE DBREORG         Processing table 26 of 133 total tables.
                                  Evaluated 2778 pages.
```

Canceling the Estimate Database Reorganization Process

You can use the CANCEL PROCESS command to cancel the server background process started by the ESTIMATE DBREORGSTATS command.

To cancel the process created by the ESTIMATE DBREORGSTATS command in the example above, enter:

```
cancel process 2
```

Displaying Results of the Estimate Database Reorganization Command

You can use the QUERY DB command to display information about the last completed estimate.

```
query db format=detailed
```

```
              Available Space (MB): 8
            Assigned Capacity (MB): 4
            Maximum Extension (MB): 4
            Maximum Reduction (MB): 0
                 Page Size (bytes): 4,096
                Total Usable Pages: 1,024
                        Used Pages: 94
                          Pct Util: 9.2
                      Max. Pct Util: 9.2
                  Physical Volumes: 3
                 Buffer Pool Pages: 64
             Total Buffer Requests: 4,702
                    Cache Hit Pct.: 97.68
                  Cache Wait Pct.: 0.00
               Backup in Progress?: No
        Type of Backup In Progress:
      Incrementals Since Last Full: 0
    Changed Since Last Backup (MB): 0.37
                Percentage Changed: 100.00
    Last Complete Backup Date/Time: 09/15/03 15:51:21
Estimate of Recoverable Space (MB): 321
Last Estimate of Recoverable Space: 09/13/03 06:03:20
```

Use the following fields to evaluate the estimation statistics:

Estimate of Recoverable Space (MB)
An estimate of the minimum amount of recoverable space, in megabytes.

Last Estimate of Recoverable Space
The date and time of the last recoverable space estimate.

Reorganizing the Database

Over time, database volumes become fragmented. You can restore the efficiency of the database and improve database performance by reorganizing the database using database unload and reload processing. By reloading the database, you compress and reorganize it.

The procedure includes unloading the database, formatting database volumes and recovery log volumes to prepare for loading, and then loading the database. The operations read device information from the device configuration file, not from the server's database.

You can use a device class of FILE for the DSMSERV UNLOADDB and DSMSERV LOADDB operations. If you use any other type of device class for the operations, you must use a drive that is assigned to a manual library (library type of MANUAL). If the drive that you want to use is not assigned to a manual library, you must edit the device configuration file to temporarily change the definition so that it appears to be in a manual library.

Note: The VOLUMEHISTORY file used by LOADDB to set the sequence numbers for a database backup and a database snapshot is the volume history file specified in the server options file. The sequence numbers for a database backup and a database snapshot will be set for future operations based on the information read from the volume history file.

Procedure: Reorganizing the Database

To reorganize the database, follow these steps:

1. Determine the amount of space that can be recovered if a database reorganization is performed. See "Estimating Database Reorganization" on page 506
2. **Attention:** Perform a backup of your database. If an outage occurs while you are unloading and reloading your database, you can use your backup copy for recovering the database. If recovery is necessary, using the RESTORE DB command is much faster than going through the process to unload, loadformat, and load.
3. Ensure that a current device configuration file exists. You must also specify the name of the device configuration file by using the DEVCONFIG option in the server options file. See "Saving the Device Configuration File" on page 658.

   Note: If you have specified more than one file with the DEVCONFIG option, remove all but one file name for this process. After you complete this procedure, you can add the other file names back to the option.

   The device configuration file includes a copy of the device class, library, and drive definitions for the server. The utility commands in this procedure need these definitions.

   See the *Administrator's Reference* for details on the DEVCONFIG option.
4. Ensure that the device configuration file contains the required definitions for the device that you want to use for the operations.
   - To use hard disk, you must use a device class of type FILE for the operations, and the device class definition must exist in the device configuration file.
   - To use other device types, check the definition of the library and the drive in the device configuration file. The library and the drive must be defined as a manual library and drive. If not, make a copy of your device configuration file and store the original file in a safe place. Edit the copy of the file to temporarily have the library and the drive be treated by the server as a manual library and drive. Follow these guidelines:
     - For the library definition, change the library type to MANUAL and remove any parameters not allowed for a MANUAL type of library. For example, you have the library defined in your device configuration file like this:

       ```
       define library 8mmlib libtype=scsi shared=yes
       ```

       You need to change the definition to this:

       ```
       define library 8mmlib libtype=manual
       ```
     - For the drive definition, remove any parameters that do not apply to a drive in a manual library. For example, you have the drive defined like this:

       ```
       define drive 8mmlib drive01 element=82
       ```

       You need to change the definition to this:

       ```
       define drive 8mmlib drive01
       ```

   See *Administrator's Reference* for details on the commands.
5. Before unloading the database, estimate how many tapes you will need:
   - If the server is *not* running, use the size of your existing physical database volumes as an estimate of how many tapes to use.

- If the server is running, you can use the following steps to estimate the number of tapes required:
  a. Request information about the database by using the following command:
     ```
     query db
     ```
  b. Using the output of the QUERY DB command, multiply the *Used Pages* by the *Page Size* to determine space occupied by the database.
  c. Use the result to estimate the number of tapes of a specific device class that you will need to unload the database. The space required will likely be less than your estimate.

6. Halt the server if it is still running.
7. With the server *not* running, issue the DSMSERV UNLOADDB utility to unload the database to tape. For example, issue this command:
   ```
   dsmserv unloaddb devclass=tapeclass scratch=yes
   ```

   Because the library is defined to be a manual library, you will need to manually mount the tapes.

   Note: Keep track of the order in which the tape volumes are written when the database is unloaded. You must specify the volume names in the same order when you reload the database using the DSMSERV LOADDB utility. For this task, you can either:
   - Review the output generated by the DSMSERV UNLOADDB utility and record the order of the volumes.
   - Manually view the volume history file to identify the tape volumes containing the unloaded database. The volumes have a volume type of DBDUMP. See "Saving the Volume History File" on page 656 for details. (Do *not* restart the server and issue QUERY VOLHISTORY at this step.)
8. Format the database and recovery log by using the DSMSERV LOADFORMAT utility. This utility prepares the existing server database for the DSMSERV LOADDB utility. For example, issue this command:
   ```
   dsmserv loadformat 2 logvol1 logvol2 1 dbvol1
   ```

   This command prepares two recovery log volumes (logvol1 and logvol2), and one database volume (dbvol1).
9. Reload the database using the volumes that contain the data from the unload operation. For example:
   ```
   dsmserv loaddb devclass=tapeclass volumenames=db001,db002,db003
   ```

   For the volume names, ensure that you do the following:
   - Enter the volume names in the same order in which they were used for the DSMSERV UNLOADDB utility.
   - Separate the volume names with a comma and no intervening spaces.

   Note: After issuing this command, a message will be displayed indicating if the DSMSERV AUDITDB command is recommended or not. It is intended to verify the consistency and integrity of the server database, report errors if any exist, and repair those that it is able to. The DSMSERV AUDITDB is not usually necessary after reorganizing the database. It is only recommended after salvaging a server database using the DSMSERV DUMPDB, DSMSERV LOADFORMAT, and DSMSERV LOADDB commands. If errors are encountered during

normal production use of the server that suggest that the database is damaged, the root cause of the errors must be determined with the assistance of IBM Support. Performing DSMSERV AUDITDB on a server database that has structural damage to the database tables may result in the loss of more data or additional damage to the database.

10. If you edited your device configuration file in step 4 on page 509, replace the edited version of the device configuration file with the original version.
11. Start the server.

Chapter 20. Monitoring the IBM Tivoli Storage Manager Server

Administrators can monitor the server:

- To find the status of operations
- To display information about objects
- To monitor the record of activity
- To select the types of events to save
- To select a location to save events

See the following sections:

Tasks:
"Using IBM Tivoli Storage Manager Queries to Display Information"
"Using SQL to Query the IBM Tivoli Storage Manager Database" on page 518
"Using the IBM Tivoli Storage Manager Activity Log" on page 523
"Logging IBM Tivoli Storage Manager Events to Receivers" on page 526
"Monitoring IBM Tivoli Storage Manager Accounting Records" on page 541
"Using IBM Tivoli Storage Manager Operational Reporting" on page 543

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Using IBM Tivoli Storage Manager Queries to Display Information

IBM Tivoli Storage Manager provides a variety of QUERY commands that display formatted information about definitions, settings, processes, and status. For some commands, you can display the information in either of two formats: standard or detailed. The standard format presents less information than the detailed format, and is useful in displaying an overview of many objects. To displaying more information about a particular object, use the detailed format when supported by a given command.

Note: For information about creating customized queries of the database, see "Using SQL to Query the IBM Tivoli Storage Manager Database" on page 518.

Requesting Information about IBM Tivoli Storage Manager Definitions

During Tivoli Storage Manager system setup, an administrator can define many objects. These objects include storage management policies, database and recovery log volumes, storage pools, and device classes. Tivoli Storage Manager provides queries that display information about these objects.

© Copyright IBM Corp. 1993, 2005

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:

```
                     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.

To request information about client sessions, enter:

```
query session
```

Figure 69 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 69. Information about Client Sessions

Check the *wait time* and *session state*. The *wait time* determines the length of time (seconds, minutes, hours) the server has been in the current state. The *session state* 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

Most commands run in the foreground, but others generate background 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. 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 entering:

```
query process
```

Figure 70 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 70. Information about Background Processes

Requesting Information about Server Settings

Any administrator can request general server information, most of which is defined by SET commands. To request this information, enter:

```
query status
```

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
- Server-free status and batch-size
- 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

Task	Required Privilege Class
Query server options	Any administrator

Use the QUERY OPTION command to display information about one or more server options.

You can issue the QUERY OPTION command with no operands to display general information about all defined server options. You also can issue the QUERY OPTION command with a specific option name or pattern-matching expression to display information on one or more server options.

To display general information about all defined server options, enter:

```
query option
```

You can set options by editing the server options file. See *Administrator's Reference* for more information.

Querying the System

The QUERY SYSTEM command combines multiple queries of your Tivoli Storage Manager system into a single command. This command can be used to collect statistics and to provide information for problem analysis by IBM service. When you issue 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 DBVOLUME
Displays information about all database volumes (detailed format)

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 LOGVOLUME
Displays information about all recovery log volumes (detailed format)

QUERY MGMTCLASS
Displays all management classes (standard format)

QUERY OPTION
Displays all server options

QUERY PROCESS
Displays information about all active background processes

QUERY SCHEDULE
Displays client schedules (standard format)

QUERY SESSION
Displays information about all administrative and client node sessions in standard format

QUERY STATUS
Displays general server parameters, such as those defined by SET commands

QUERY STGPOOL
Displays information about all storage pools (detailed format)

QUERY VOLUME
Displays information about all storage pool volumes (standard format)

SELECT
Displays the results of two SQL queries:

```
select platform_name,count(*) from nodes group by platform_name
```

```
select stgpool_name,devclass_name,count(*) from volumes
group by stgpool_name,devclass_name
```

The first command displays the number of client nodes by platform.

The second command displays the name and associated device class of all storage pools having one or more volumes assigned to them.

Using SQL to Query the IBM Tivoli Storage Manager Database

You can use a standard SQL SELECT statement to get information from the database. The SELECT command is a subset of the SQL92 and SQL93 standards.

IBM Tivoli Storage Manager also provides an open database connectivity (ODBC) driver. The driver allows you to use a relational database product such as Lotus Approach® to query the database and display the results.

Using the ODBC Driver

IBM Tivoli Storage Manager provides an ODBC driver for Windows. The driver supports the ODBC Version 2.5 application programming interface (API). Because Tivoli Storage Manager supports only the SQL SELECT statement (query), the driver does not conform to any ODBC API or SQL grammar conformance level. After you install this driver, you can use a spreadsheet or database application that complies with ODBC to access the database for information.

The ODBC driver set-up is included in the client installation package. The client installation program can install the ODBC driver and set the corresponding registry values for the driver and data sources. For more information on setting up the ODBC driver, see *Backup-Archive Clients Installation and User's Guide*.

To open the database through an ODBC application, you must log on to the server (the defined data source). Use the name and password of a registered administrator. After you log on to the server, you can perform query functions provided by the ODBC application to access database information.

Issuing SELECT Commands

You can issue the SELECT command from the command line of an administrative client. You cannot issue this command from the server console.

The SELECT command supports a subset of the syntax of the SELECT statement as documented in the SQL92 and SQL93 standards. For complete information about how to use the SELECT statement, refer to these standards or to other publications about SQL.

Issuing the SELECT command to the server can use a significant amount of server resources to run the query. Complicated queries or queries that run for a long time can interfere with normal server operations. If your query requires excessive server resource to generate the results, you will receive a message asking you to confirm that you wish to continue.

Note: To allow any use of the SELECT command, the database must have at least 4MB of free space. For complex queries that require significant processing, additional free space is required in the database. See "Problems with Exhausting Temporary Table Storage" on page 520 for details.

Learning What Information Is Available: System Catalog Tables

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 to determine the location of the information that you want. For example, to get a list of all tables available for querying in the database, enter the following command:

```
select * from syscat.tables
```

The following shows part of the results of this command:

```
     TABSCHEMA: ADSM
       TABNAME: ACTLOG
   CREATE_TIME:
      COLCOUNT: 11
INDEX_COLCOUNT: 1
  UNIQUE_INDEX: FALSE
       REMARKS: Server activity log

     TABSCHEMA: ADSM
       TABNAME: ADMINS
   CREATE_TIME:
      COLCOUNT: 17
INDEX_COLCOUNT: 1
  UNIQUE_INDEX: TRUE
       REMARKS: Server administrators

     TABSCHEMA: ADSM
       TABNAME: ADMIN_SCHEDULES
   CREATE_TIME:
      COLCOUNT: 15
INDEX_COLCOUNT: 1
  UNIQUE_INDEX: TRUE
       REMARKS: Administrative command schedules

     TABSCHEMA: ADSM
       TABNAME: ARCHIVES
   CREATE_TIME:
      COLCOUNT: 10
INDEX_COLCOUNT: 5
  UNIQUE_INDEX: FALSE
       REMARKS: Client archive files
```

Examples

The SELECT command lets you customize a wide variety of queries. This section shows two examples. For many more examples of the command, see the *Administrator's Reference*.

Example 1: Find the number of nodes by type of operating system by issuing the following command:

```
select platform_name,count(*) as "Number of Nodes" from nodes
group by platform_name
```

This command gives results like the following:

```
PLATFORM_NAME     Number of Nodes
-------------     ---------------
OS/2                           45
AIX                            90
Windows                        35
```

Example 2: For all active client sessions, determine how long they have been connected and their effective throughput in bytes per second:

```
select session_id as "Session", client_name as "Client", state as "State",
  current_timestamp-start_time as "Elapsed Time",
  (cast(bytes_sent as decimal(18,0)) /
  cast((current_timestamp-start_time)seconds as decimal(18,0)))
  as "Bytes sent/second",
  (cast(bytes_received as decimal(18,0)) /
  cast((current_timestamp-start_time)seconds as decimal(18,0)))
  as "Bytes received/second"
  from sessions
```

This command gives results like the following:

```
                 Session: 24
                  Client: ALBERT
                   State: Run
            Elapsed Time: 0 01:14:05.000000
       Bytes sent/second: 564321.9302768451
   Bytes received/second: 0.0026748857944

                 Session: 26
                  Client: MILTON
                   State: Run
            Elapsed Time: 0 00:06:13.000000
       Bytes sent/second: 1638.5284210992221
   Bytes received/second: 675821.6888561849
```

Problems with Exhausting Temporary Table Storage

SQL SELECT queries run from temporary table storage in the database. At least a 4MB partition must be available in the database for this purpose. Without this partition, temporary table storage space will become exhausted, and the SELECT query will no longer run.

To determine how much temporary table storage space is available in your database, issue the QUERY DB command. The server displays a report, like the following:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
        8        4         4         0   4,096     1,024        94   9.3   9.2
```

Check the value in the **Maximum Reduction** field. If this field shows a value of at least 4MB, you can perform SELECT queries.

If the **Maximum Reduction** value is below 4MB, you will not be able to perform SELECT queries. The database is either full or fragmented.

- If the database is full, increase the size of the database. See "Increasing the Size of the Database or Recovery Log" on page 496 for details.
- If the database is fragmented, either add a volume or unload and load your database. To estimate the amount of space that can be recovered from database reorganization (to reduce fragmentation), issue the DBREORGSTATS command. To see the results from the last time this command was issued, issue the QUERY DB FORMAT=DETAILED command. See "Reorganizing the Database" on page 508 for details.

Note: Complex SELECT queries (for example, those including the ORDER BY clause, the GROUP BY clause, or the DISTINCT operator) may require more than 4MB temporary table storage space.

Using SELECT Commands in IBM Tivoli Storage Manager Scripts

A Tivoli Storage Manager script is one or more commands that are stored as an object in the database. You can run a script from an administrative client, the Administration Center, or the server console. You can also include it in an administrative command schedule to run automatically. See "IBM Tivoli Storage Manager Server Scripts" on page 477 for details. You can define a script that contains one or more SELECT commands. 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 define and extend database volumes and 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 "
/* Define a DB volume and extend the database                          */

def script def_db_extend '/*  ----------------------------------------*/'
upd script def_db_extend '/*  Script Name:  DEF_DB_EXTEND             */'
upd script def_db_extend '/*  Description: Define a database volume,   */'
upd script def_db_extend '/*               and extend the database     */'
upd script def_db_extend '/*  Parameter 1: db volume name              */'
upd script def_db_extend '/*  Parameter 2: extension megabytes         */'
```

```
upd script def_db_extend '/*  Example:  run def_db_extend VOLNAME 12   */'
upd script def_db_extend '/*  ----------------------------------------*/'
upd script def_db_extend ' def dbv  $1 '
upd script def_db_extend ' if (rc_ok) extend db $2'
upd script def_db_extend ' if (warning, error) q db f=d'
```

Canceling a SELECT Command

If a SELECT command will require a significant amount of resources, the server asks if you want to continue. You can cancel the command at that time. Cancel the command from the console session or an administrative client session.

Controlling the Format of SELECT Results

IBM Tivoli Storage Manager provides commands to control the format of results of SELECT commands. You can control:

- How SQL data types such as VARCHAR are displayed, in wide or narrow format (SET SQLDISPLAYMODE)
- The format of date and time values in the results (SET SQLDATETIMEFORMAT)
- Whether SQL arithmetic results are truncated or rounded (SET SQLMATHMODE)

Note: Using the SET commands to change these settings keeps the settings in effect only for the current administrative client session. You can query these settings by using the QUERY SQLSESSION command.

Querying the SQL Activity Summary Table

You can query the SQL activity summary table to view statistics about some of the client operations and server processes. Some of the client operations recorded to the table are BACKUP, RESTORE, ARCHIVE and RETRIEVE. Server processes include MIGRATION, RECLAMATION and EXPIRATION.

For a listing of the 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: 2002-07-22 19:32:00.000000
    END_TIME: 2002-07-22 19:32:56.000000
    ACTIVITY: BACKUP
      NUMBER: 43
      ENTITY: DWE
    COMMMETH: Named Pi
     ADDRESS:
    EXAMINED: 7
    AFFECTED: 7
      FAILED: 0
       BYTES: 2882311
        IDLE: 51
      MEDIAW: 0
```

```
 PROCESSES: 1
SUCCESSFUL: YES

ANS8002I Highest return code was 0.
```

- To display all events starting at or after 00:00 a.m. on September 24, 2002 until the present time, enter:

```
select * from summary where start_time>= '2002-09-24 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.

Creating Output for Use by Another Application

You can redirect the output of SELECT commands to a file in the same way as you would redirect the output of any command. When redirecting this output for use in another program (for example, a spreadsheet or database program), write the output in a format easily processed by the program to be used.

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 during the administrative session will be formatted into either comma-separated or tab-separated values. For details about using command line options, see the *Administrator's Reference*.

The use of command output redirection and one of the delimited output format options lets you create queries whose output can be further processed in other applications. For example, based on the output of a SELECT command, a spreadsheet program could produce graphs of average file sizes and file counts summarized by type of client platform.

For details about redirecting command output, see the *Administrator's Reference*.

Using the IBM Tivoli Storage Manager Activity Log

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

The activity log contains all messages normally sent to the server console during server operation. The only exceptions are responses to commands entered at the console, such as responses to QUERY commands.

Examples of messages sent to the activity log include:

- When client sessions start or end
- When migration starts and ends

- When backup versions expire
- What data is exported to tape
- When expiration processing is performed
- What export or import processing is performed

Any error messages sent to the server console are also stored in the activity log.

Use the following sections to adjust the size of the activity log, set an activity log retention period, and request information about the activity log.

Requesting Information from the Activity Log

You can request information stored in the activity log. To minimize processing time when querying the activity log, you can:

- Specify a time period in which messages have been generated. The default for the QUERY ACTLOG command shows all activities that have occurred in the previous hour.
- Specify the message number of a specific message or set of messages.
- Specify a string expression to search for specific text in messages.
- Specify the QUERY ACTLOG command from the command line for large queries instead of using the graphical user interface.
- Specify whether the originator is the server or client. If it is the client, you can specify the node, owner, schedule, domain, or session number. If you are doing client event logging to the activity log and are only interested in server events, then specifying the server as the originator will greatly reduce the size of the results.

For example, to review messages generated on May 30 between 8 a.m. and 5 p.m., enter:

```
query actlog begindate=05/30/2002 enddate=05/30/2002
begintime=08:00 endtime=17:00
```

To request information about messages related to the expiration of files from the server storage inventory, enter:

```
query actlog msgno=0813
```

Refer to *Messages* for message numbers.

You can also request information only about messages logged by one or all clients. For example, to search the activity log for messages from the client for node JEE:

```
query actlog originator=client node=jee
```

Requesting Session and Process Information

Often, the messages issued on the behalf of a session or process do not indicate (in the message itself) the session or process to which they are related. To help you correlate messages with the session or process for which they are issued, the server records additional information about messages that can be generated by querying the activity log. For example, an error message issued because of a volume mount failure for a client session operation does not have any explicit information to tie it back to the session that was performing the action. Scenario 1: Client FRED has been having problems getting it's scheduled backup to complete. During the nightly scheduled backup's, there are 1000 different sessions running for scheduled clients and TDP's and data is being stored to various destinations of DISK and TAPE. For the scheduled backup's when FRED has failed, it has not been clear

from the activity log which error messages were issued that related to FRED. Using this improvement to the activity log information, an administrator would only need to query the activity log to determine what session or sessions fred was using for the nightly backup. And than for each session, they could issue a ″QUERY ACTLOG search=″(SESSION: sessNum)″ or they could issue ″SELECT * from ACTLOG where SESSION=sessNum″. Either of these would report ALL the messages that were issued in relation to that session.

You can also request n information about a client session. For example, to search the activity log for a message that were issued in relation to a session, enter:

```
query actlog search="(SESSION:4)"
```

This command gives results like the following:

```
    EXAMPLE 2 (Client SESSION that performs a backup)
09/23/2003 10:26:38  ANR0406I Session 4 started for node FRED (WinNT) (Tcp/Ip
                      colind(2463)).  (SESSION: 4)
09/23/2003 10:26:40  ANR8493I FILE volume C:\CODE\522\00000000.BFS mounted in
                      drive OUTFILE4 (FILE) in library OUTFILE.  (SESSION: 4)
09/23/2003 10:26:40  ANR8340I FILE volume C:\CODE\522\00000000.BFS mounted.
                      (SESSION: 4)
09/23/2003 10:26:40  ANR8468I FILE volume C:\CODE\522\00000000.BFS 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)
```

Setting a Retention Period for the Activity Log

You can use the SET ACTLOGRETENTION command to specify how long activity log information is kept in the database. Activity log management is retention-based when the optional parameter MGMTSTYLE is set to its default value, DATE. The server automatically deletes messages from the activity log once the number of days that are specified pass. At installation, activity log management is retention-based, and the retention period is set to one day. To change the retention period to 10 days, for example, enter:

```
set actlogretention 10
```

To disable activity log retention, set the SET ACTLOGRETENTION command to zero. To display the current retention period and size of the activity log, query the server status.

Note: With retention-based management, you lose some control over the amount of space that the activity log occupies. For more information on size-based activity log management, see “Setting a Size Limit for the Activity Log.”

Setting a Size Limit for the Activity Log

You can use size-based activity log management as an alternative to retention-based management. This allows greater control over the amount of space that the activity log occupies. The server will periodically remove the oldest activity log records until the activity log size no longer exceeds the configured maximum size allowed. To manage the activity log by size, the parameter MGMTSTYLE must be set to the value SIZE. To change the maximum size of the activity log to 12 MB, for example, enter:

```
set actlogretention 12 mgmtstyle=size
```

To disable activity log retention, set the SET ACTLOGRETENTION command to zero. To display the current and maximum size of the activity log, query the server status.

Note: With size-based management, you lose some control over the length of time that activity log messages are kept. For more information on retention-based activity log management, see "Setting a Retention Period for the Activity Log" on page 525.

Logging IBM Tivoli Storage Manager Events to Receivers

The server and client messages provide a record of Tivoli Storage Manager activity that you can use to monitor the server. You can log server messages and most client messages as *events* to one or more repositories called *receivers*. You can log the events to any combination of the following receivers:

Tivoli Storage Manager server console and activity log
See "Logging Events to the IBM Tivoli Storage Manager Server Console and Activity Log" on page 528.

File and user exits
See "Logging Events to a File Exit and a User Exit" on page 528.

Tivoli event console
See "Logging Events to the Tivoli Enterprise Console" on page 529.

Simple Network Management Protocol (SNMP)
See "Logging Events to an SNMP Manager" on page 533.

Event server receiver (Enterprise Event Logging)
Routes the events to an event server. See "Enterprise Event Logging: Logging Events to Another Server" on page 538.

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 71 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 71. Event Logging Overview

Task	Required Privilege Class
Enable or disable events	System
Begin or end event logging	

Controlling Event Logging

To control event logging do the following:

1. Enable or disable logging for one or more event types and for one or more receivers.
2. Begin or end logging to one or more receivers.

Enabling and Disabling Events

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 start-up 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

A receiver for which event logging has begun is an *active receiver*. To begin and end logging for one or more receivers, issue the BEGIN EVENTLOGGING and END EVENTLOGGING commands.

At server start-up 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. See the appropriate receiver sections for details. 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. To enable all error and severe client events to the console and activity log, issue the following command:

```
enable events console,actlog error,severe
```

Note: 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 525 and "Setting a Size Limit for the Activity Log" on page 525). At server installation, activity log management is retention-based, and this value is set to one day. If you increase the retention period or the size limit, utilization is further increased.For more information about the activity log, see "Using the IBM Tivoli Storage Manager Activity Log" on page 523.

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.

Logging Events to a File Exit and a User Exit

You can log 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. 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 756.
- A user exit is an external interface in the form of an executable, user-written program. Tivoli Storage Manager supports user exits.

Note: Both types of event receivers must be specified in the server options file (dsmserv.opt) file.

Both file and user exits receive event data in the same data block structure. Setting up logging for these receivers is also similar:

1. Add an option for the exit to the server options file:

- **For a file exit:** Add either the FILEEXIT option (for a binary file exit) or FILETEXTEXIT (for a text file exit) option.
  - 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.
  - Specify the file where each logged event is to be stored.
  - 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 527 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 527 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. 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

Event Name	Source
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

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.

Controlling the Format of Events

Two server options are available to control the format of events sent from the Tivoli Storage Manager server to the Tivoli Enterprise Console. Enabling either of these options not only changes the event class format, but also generates a unique event class for individual Tivoli Storage Manager messages for the client, the server, application clients, Data Protection for IBM ESS for DB2, Data Protection for IBM ESS for Oracle, and Data Protection for IBM ESS for R/3.

Option Name	Function
UNIQUETECEVENTS	Changes the event class format and generates a unique event class for client, server, and some Data Protection messages
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.

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:

Data Protection for IBM ESS for DB2	EEP0000 to EEP9999
Data Protection for IBM ESS for Oracle	EEO0000 to EEO9999
Data Protection for R/3 and Data Protection for IBM ESS for R/3	BKI0000 to BKI9999

As noted, enabling UNIQUETDPTECEVENTS also enables UNIQUETECEVENTS. This means that all recognized Data Protection messages will be sent as unique events from the Tivoli Storage Manager server.

Based upon the setting of the option or options on the Tivoli Storage Manager server, the Tivoli Enterprise Console administrator must create a rule base using one of the following baroc files:

UNIQUETDPTECEVENTS Setting	**UNIQUETECEVENTS Setting**	**Baroc File**
NO	NO	ibmtsm.baroc
NO	YES	itsmuniq.baroc
YES	Defaults to YES because UNIQUETDPTECEVENTS is set to YES.	itsmdpex.baroc

Each successive baroc file accepts the events of the previous baroc file. For example, itsmuniq.baroc accepts all events in ibmtsm.baroc, and itsmdpex.baroc accepts all events contained in itsmuniq.baroc.

Encoding Events to UTF-8

Tivoli Storage Manager supports the following option to encode a Tivoli Enterprise Console event into UTF-8 before sending it to the Tivoli Enterprise Console server. Some Tivoli Enterprise Console patches (for example, Patch 0004 for Tivoli Enterprise Console Version 3.6 Modification 2) require UTF-8 encoded events for some locales.

Option Name	Function
TECUTF8EVENT	Encodes a Tivoli Enterprise Console event into UTF-8

To determine whether this option is enabled, issue the QUERY OPTION command.

To Set up a Tivoli Enterprise Console as a Receiver

To set up Tivoli as a receiver for event logging:

1. Define the Tivoli Storage Manager event classes to the Tivoli Enterprise Console with the ibmtsm.baroc file, which is distributed with the server.

   Note: Please refer to Tivoli Enterprise Console documentation for instruction on removing an existing baroc file, if needed, and installing a new baroc file. If you have migrated from ADSM Version 3 and have an existing ibmadsm.baroc file, do one of the following:
   - Remove the file.
   - Create a new rule base.
   - Copy the file.

   Before the events are displayed on a Tivoli Enterprise Console, you must import ibmtsm.baroc into an existing rule base or create a new rule base and activate it. To do this:
   - From the Tivoli desktop, click on the **Rule Base** icon to display the pop-up menu.
   - Select **Import**, then specify the location of the ibmtsm.baroc file.
   - Select the **Compile** pop-up menu.
   - Select the **Load** pop-up menu and **Load, but activate only when server restarts** from the resulting dialog.
   - Shut down the event server and restart it.

   To create a new rule base, do the following:
   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 ibmtsm.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 527 for more information.
4. In the server options file (dsmserv.opt), 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 527 for more information.

Logging Events to an SNMP Manager

Tivoli Storage Manager supports the simple network management protocol (SNMP) together with event logging to do the following:

- Set up an SNMP heartbeat monitor to regularly check that the Tivoli Storage Manager server is running.
- Send messages known as *traps* to an SNMP manager, such as NetView® or Tivoli Enterprise Console.
- Run Tivoli Storage Manager scripts and retrieve output and return codes. See "IBM Tivoli Storage Manager Server Scripts" on page 477 for details.

Tivoli Storage Manager also implements an SNMP subagent that can be configured to report exception conditions and provide support for a management information base (MIB). The management information base (MIB), which is shipped with Tivoli Storage Manager, defines the variables that will run server scripts and return the server scripts' results. You must register SNMPADMIN, the administrative client the server runs these scripts under. Although a password is not required for the subagent to communicate with the server and run scripts, a password should be defined for SNMPADMIN to prevent access to the server from unauthorized users. An SNMP password (community name) is required, however, to access the SNMP agent, which forwards the request to the subagent.

Note: Because the SNMP environment has weak security, you should consider not granting SNMPADMIN any administrative authority. This restricts SNMPADMIN to issuing only Tivoli Storage Manager queries.

SNMP SET requests are accepted for the name and input variables associated with the script names stored in the MIB by the SNMP subagent. This allows a script to be processed by running a GET request for the ibmAdsm1ReturnValue and ibmAdsm2ReturnValue variables. A GETNEXT request will not cause the script to run. Instead, the results of the previous script processed will be retrieved. When an entire table row is retrieved, the GETNEXT request is used. When an individual variable is retrieved, the GET request is used.

Here is a typical Tivoli Storage Manager configuration with SNMP:

1. Systems A, B, C: A Tivoli Storage Manager server communicates with a local subagent.
2. System D: A DPI-enabled SNMP agent is installed. This is required for communication between the Tivoli Storage Manager SNMP subagent, dsmsnmp, and the SNMP Manager you are using. A DPI-enabled SNMP agent is available as part of the AIX operating system.
3. System E: An SNMP manager, such as NetView, is installed.
4. The subagents on systems A, B, and C communicate with the agent on system D.
5. The agent on system D forwards SNMP traps to NetView on system E.

To run an arbitrary command from an SNMP management application, for example, NetView, follow these steps:

1. Choose the name and parameters for a Tivoli Storage Manager script.
2. Use the application to communicate with the SNMP agent. This agent changes the Tivoli Storage Manager MIB variable for one of the two script names that the Tivoli Storage Manager subagent maintains. The SNMP agent also sets the parameter variables for one of the two scripts.
3. Use the application to retrieve the variable *ibmAdsmReturnValue1.x* or *ibmAdsmReturnValue2.x*, where *x* is the index of the server that is registered with the subagent.

To set the variables associated with the script (for example, *ibmAdsmServerScript1/2* or *ibmAdsmM1Parm1/2/3*), the nodes on which the subagent and the agent are run must have read-write authority to the MIB variables. This is done through the SNMP configuration process on the system that the SNMP agent runs on.

Here is an AIX example:

```
community public 9.115.20.174 255.255.255.254 readWrite
community public 9.115.46.25  255.255.255.254 readWrite
community public 127.0.0.1    255.255.255.254 readWrite
community  public 9.115.20.176 255.255.255.254 readWrite
smux       1.3.6.1.4.1.2.3.1.2.2.1.1.2 public
```

The statements grant read-write authority to the MIB for the local node through the loopback mechanism (127.0.0.1), and to nodes with the three 9.115.xx.xx addresses. The smux statement allows the dpid2 daemon to communicate with snmpd.

Here is an example of this command used to set and retrieve MIB variables:

```
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmServerScript1.1=QuerySessions
```

This command issues the set operation (-ms), passing in community name **public**, sending the command to host **tpcnov73**, and setting up variable *ibmAdsmServerScript1* to have the value *QuerySessions*. *QuerySessions* is the name of a server script that has been defined on a server that will register with the Tivoli Storage Manager subagent. In this case, the first server that registers with the subagent is the *.1* suffix in *ibmAdsmServerScript1.1*. The following commands set the parameters for use with this script:

```
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm1.1=xyz
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm2.1=uvw
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm3.1=xxx
```

You can set zero to three parameters. Only the script name is needed. To make the *QuerySessions* script run, retrieve the *ibmAdsmM1ReturnValue* variable (in this case, *ibmAdsmM1ReturnValue.1*). For example:

```
snmpinfo -v -mg -c public -h tpcnov73 ibmAdsmM1ReturnValue.1
```

The results of the command are returned as a single string with embedded carriage return/newline characters.

Note: Not all MIB browsers properly handle embedded carriage return/newline characters.

In this case, *ibmAdsmM1ReturnCode.1* will contain the return code associated with the running of the script. If *ibmAdsmM2ReturnValue* is retrieved, the results of running the script named in *ibmAdsmServerScript2* are returned as a single numeric return code. Notice the *-mg* instead of *-ms* to signify the GET operation in the command to retrieve *ibmAdsmM1ReturnValue.1*. If the entire row is retrieved, the command is not run. Instead, the results from the last time the script was run are retrieved. This would be the case if the following command were issued:

```
snmpinfo -v -md -c public -h tpcnov73 ibmAdsm
```

in which all Tivoli Storage Manager MIB variables are displayed.

An SNMP agent is needed for communication between an SNMP manager and its managed systems. The SNMP agent is realized through the **snmpd daemon**. The Distributed Protocol Interface (DPI®) Version 2 is an extension of this SNMP agent.

SNMP managers can use the MIB that is shipped with Tivoli Storage Manager to manage the server. Therefore, an SNMP agent supporting DPI Version 2 must be used to communicate with the Tivoli Storage Manager subagent. This SNMP agent is not included with Tivoli Storage Manager. A supported DPI agent ships with AIX. The Tivoli Storage Manager subagent is included with Tivoli Storage Manager and, before server startup, must be started as a separate process communicating with the DPI-enabled SNMP agent.

Note: The snmpd daemon on Linux does not support DPI. Therefore, the subagent must be configured to communicate with a DPI-enabled SNMP agent that is running on a Windows or AIX system. Alternatively, you can configure the server to communicate with an existing Tivoli Storage Manager subagent. The Tivoli Storage Manager subagent, in turn, communicates with the DPI-enabled SNMP agent.

The SNMP manager system can reside on the same system as the Tivoli Storage Manager server, but typically would be on another system connected through SNMP. The SNMP management tool can be any application, such as NetView or Tivoli Enterprise Console, which can manage information through SNMP MIB

monitoring and traps. The Tivoli Storage Manager server system runs the processes needed to send Tivoli Storage Manager event information to an SNMP management system. The processes are:

- SNMP agent (snmpd)
- Tivoli Storage Manager SNMP subagent (dsmsnmp)
- Tivoli Storage Manager server (dsmserv)

Figure 72 illustrates a typical Tivoli Storage Manager implementation:

Figure 72. Tivoli Storage Manager SNMP Implementation

Figure 73 on page 537 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 73. Manager-Agent-Subagent Communication

Notes:

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.

Configuring IBM Tivoli Storage Manager SNMP

The IBM Tivoli Storage Manager SNMP set up procedure is illustrated by Figure 74:

Figure 74. IBM Tivoli Storage Manager SNMP Set Up

To set up Tivoli Storage Manager monitoring through SNMP, do the following:

1. Modify the server options file to specify the SNMP communication method. Figure 75 on page 538 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. For details about server options, see the server options section in *Administrator's Reference*.

```
commmethod                  snmp
   snmpsubagent             hostname jimbo communityname public timeout 600
   snmpheartbeatinterval    5
   snmpmessagecategory      severity
```

Figure 75. Example of SNMP Communication Method Options

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.

   Before starting the agent, ensure that the **dpid2** and **snmpd** subsystems have been started.
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. 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 76 on page 539 shows the relationship of a sending Tivoli Storage Manager server and a Tivoli Storage Manager event server.

Figure 76. 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 568.

   ```
   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

The QUERY ENABLED command displays a list of server or client events that are enabled or disabled by a specified receiver. Because the lists of enabled and disabled events could be very long, Tivoli Storage Manager displays the shorter of the two lists. For example, assume that 1000 events for client node HSTANFORD were enabled for logging to the user exit and that later two events were disabled. To query the enabled events for HSTANFORD, 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
```

Using Tivoli Decision Support

Beginning with IBM Tivoli Storage Manager Version 5.2, Tivoli Decision Support for Storage Management Analysis will no longer be shipped. If you are already using Tivoli Decision Support for Storage Management Analysis, you may continue to use it with this version of Tivoli Storage Manager.

To use Tivoli Decision Support for Storage Management Analysis on your Tivoli Storage Manager server or servers, you must first enable event logging of client events to the activity log. See "Logging Events to the IBM Tivoli Storage Manager Server Console and Activity Log" on page 528 for details.

For documentation about Tivoli Decision Support for Storage Management Analysis visit the Web site at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html.

Scheduling the Decision Support Loader with IBM Tivoli Storage Manager

You can schedule the Decision Support Loader (DSL) to run automatically using the Tivoli Storage Manager Scheduler. Before defining a schedule, ensure that the backup-archive client is installed on a dedicated Windows workstation where the DSL is installed.

Use the following procedure to schedule the DSL:

1. **On the Tivoli Storage Manager server:**
   a. Register the client node. Assume that the client node you registered is called ASTROdsl.
   b. Define a client schedule on the server from which the DSL will extract data. For example, if the schedule is called TSMDSL and client node ASTROdsl is registered in the STANDARD domain, enter:

      ```
      define schedule standard tsm_dsl action=c
      object='"c:\program files\tivoli\tsm\decision\tsmdsl"'
      ```

      Notes:
      1) The installation directory path for the DSL is:

         ```
         "c:\program files\tivoli\tsm\decision\tsmdsl.exe"
         ```

      2) Enclose the full directory path in quotation marks as shown in the previous example.

   c. Associate the client node to the *tsm_dsl* schedule. For example:

      ```
      define association standard tsm_dsl ASTROdsl
      ```

2. **On the client's workstation:**
   a. Ensure that the scheduler is installed.
   b. Start the scheduler for the client. Leave the scheduler running until scheduled rollups are no longer needed. To start the scheduler, you can open a command prompt window and navigate to where the backup-archive client is installed and enter:

      ```
      > dsmc schedule
      ```

Note: If the DSL is not processed according to the schedule you have defined, check the directory path where the DSL is installed.

Monitoring IBM Tivoli Storage Manager Accounting Records

Task	Required Privilege Class
Set accounting records on or off	System

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. At installation, accounting defaults to OFF. You can set accounting to ON by entering:

```
set accounting on
```

When accounting is 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
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.
26	Number of space-managed store transactions requested during the session
27	Amount of space-managed data, in kilobytes, sent by the client to the server
28	Number of space-managed retrieve transactions requested during the session
29	Amount of space, in kilobytes, retrieved by space-managed objects
30	Product release
31	Product level

The following shows a sample record:

```
3,8,ADSM,08/03/2000,16:26:37,node1,,AIX,1,Tcp/Ip,0,254,1713,0,0,47,1476,0,0,3316,960,27,5,1,4,0,0,0,0,7,2
```

Daily Monitoring Scenario

This section contains an example of daily monitoring of a Tivoli Storage Manager system. Depending on the configuration of your system, you may want to perform additional monitoring tasks. If a function does not complete properly, you can review the activity log for errors that occurred around the time of failure (see "Requesting Information from the Activity Log" on page 524 for details).

Note: To automate monitoring tasks, you can use operational reporting. See "Using IBM Tivoli Storage Manager Operational Reporting."

You can include the commands shown in a command script that you can run daily. Review the output of the script for any errors or problems.

1. Verify that drives are online. If there is a drive in the unavailable state, there may be errors with schedules.

   ```
   query drive
   ```

2. Verify that database and recovery log volumes are online and synchronized.

   ```
   query dbvolume
   query logvolume
   ```

3. Check the status of disk volumes. If any are offline, check for hardware problems.

   ```
   query volume devclass=disk
   ```

4. Check that scratch volumes are available.

   ```
   query libvolume
   ```

5. Check the access state of the tape volumes. For example, a volume that is not in the read-write state may indicate a problem. You may need to move data and check the volumes out of the library.

   ```
   query volume
   ```

6. Check database and recovery log statistics.

   ```
   query db
   query log
   ```

7. Verify that scheduled database backups completed successfully.

   ```
   query volhistory type=dbbackup
   ```

8. Check the activity log for error messages.

   ```
   query actlog search=ANR????E
   ```

Using IBM Tivoli Storage Manager Operational Reporting

This section contains the following tasks for Operational Reporting:

- "Generating a Comprehensive Report at Regular Intervals" on page 547
- "Generating a Comprehensive Monitor at Regular Intervals" on page 551
- "Completing the Set Up" on page 553
- "Installing the IBM Tivoli Storage Manager Operational Reporting Stand-alone Package" on page 546
- "Generating Reports and Monitors from the Command Line" on page 558

Operational Reporting Concepts

The IBM Tivoli Storage Manager operational reporting feature automates some of the monitoring tasks you typically perform manually. By generating reports and monitors, operational reporting notifies you if a server requires attention. Operational reports can be scheduled to run daily and are generated even if there

are no problems. Operational monitors are special types of reports, and can be scheduled to run hourly. The monitors will send you a notification only if there are issues. Operational reporting does not maintain a separate database of information and is not a trending tool.

Operational reporting is included as part of the Tivoli Storage Manager for Windows server and is also available as a stand-alone package on a Windows server. For information on installing the stand-alone package, see "Installing the IBM Tivoli Storage Manager Operational Reporting Stand-alone Package" on page 546.

Operational reporting is administered through the Microsoft Management Console on a Windows machine. All platforms of IBM Tivoli Storage Manager servers, Version 5.1.8 or Version 5.2.2 and later, are supported. Operational reporting runs as a service and supports multiple Tivoli Storage Manager servers running on a single machine.

An operational report consists of the following: a standard report, a customized summary, and optional extensions that you can create. You can select which sections to include in the report. The operational reporting installation package contains two default custom summary templates: one for a report and one for a monitor.

Default e-mail messages will notify you if the server is running smoothly, or if there are issues such as failed or missed schedules. You can also link to a Web summary page to check operational reports about your server. An operational monitor will notify you either through e-mail or by sending an instant message to your Windows desktop.

Operational reporting can write data to a file which can be read by a Tivoli Enterprise Console log file adapter. The log file adapter reads the information and forwards it to the Tivoli Enterprise Console.

Using Operational Reports and Operational Monitors

A Tivoli Storage Manager server must be running in order to generate reports and monitors. These can be sent to multiple users such as server administrators or managers. If there are problems, they appear at the beginning of the report along with recommendations on how to fix them. Any issues in the report are followed by the standard section, custom summary, a timing section, and any optional extensions. Monitors do not contain standard sections and typically run more quickly and frequently. Reports and monitors have default values.

Reports and monitors can run simultaneously, and can either be scheduled or run as needed. They allow you to flag issues by specifying notification rules in the custom summary section. If a threshold in a notification rule is met, you will be notified.

Reports and monitors include a timing section, which can help you determine performance problems. For example, if a report or monitor seems to be running slowly, you can use the timing section to identify where the time is being spent, and you can eliminate or optimize a section accordingly.

Figure 77 on page 545 displays an example of a report and a monitor on a Web summary. In the report and monitor, the server is running smoothly. Click on the **Expand All** tab to see all report and monitor sections per server.

Figure 77. Operational Reporting Report Overview

Figure 78 on page 546 illustrates a daily report with the issues and recommendations expanded. The issue is that too many client schedules have been missed. The recommendations are to confirm that a scheduler is running on each client, and that the option file of the scheduler is pointing to the correct server. The custom summary of the report is also expanded to show results for that server.

Figure 78. Operational Reporting Report Details

Installing the IBM Tivoli Storage Manager Operational Reporting Stand-alone Package

The IBM Tivoli Storage Manager Operational Reporting feature automates your daily monitoring tasks, and generates reports that can tell you if a server requires attention. You can run the Operational Reporting interface on Windows only, but it supports servers running on any supported operating system. See the Readme file, included with the installation package, for a list of supported IBM Tivoli Storage Manager servers. Operational reporting views a Tivoli Storage Manager server as being in one of two states: Running Smoothly or Needs Attention. The state of a Tivoli Storage Manager server is automatically determined and a message can be sent to you in an e-mail, posted to the Web, or displayed on the management console. The message provides access to a status report that can be customized. If a server needs attention, the message describes any issues and provides recommendations for getting Tivoli Storage Manager running smoothly again.

IBM Tivoli Storage Manager operational reporting is available as:

- Part of the Tivoli Storage Manager management console included when the Tivoli Storage Manager Windows server and storage agent packages are installed.
- A stand-alone package as an independent component. Use this package if you do not run Windows Tivoli Storage Manager servers or storage agents, and do not anticipate doing so in the future.

The stand-alone package will not be allowed to install on a machine where the Tivoli Storage Manager server or storage agent is installed, and vice-versa. If you

install the stand-alone version and later decide to install the Tivoli Storage Manager server on Windows, you must first uninstall the stand-alone console.

Installing Tivoli Storage Manager Operational Reporting as Stand-alone Package

1. Insert the Operational Reporting CD.
2. The Welcome InstallShield wizard panel displays. Click **Next**.
3. **Location to Save Files**: If needed, change the path where the files are saved, or accept the default. Click **Next**.
4. **Extracting Files**: Files will now be extracted, Setup Wizard will launch when the panel completes extracting files.
5. **Welcome to the Setup Install Shield Wizard**: Click **Next**.
6. **License Agreement**: Select **I accept** and click **Next**.
7. **Customer Information**: Enter your User Name and Organization Name. Click **Next**.
8. **Set Up Type**: Select **Complete**. Click **Next**.
9. **Ready to Install the Program**: Click **Install**.
10. **Install Shield Wizard Complete**: Installation of Operational Reporting is complete. Click **Finish**. The Operational Reporting icon should now appear on your desktop.

Configuring Operational Reporting

To quickly start operational reporting, follow the steps below.

1. Click on the operational reporting icon.
2. Right click on the **Tivoli Storage Manager** icon and choose **Add TSM Computer**.
3. Click **TSM Operational Reporting** and click **Add** to add a report, and click **OK** to accept the defaults.
4. Expand the tree view in the MMC snap-in to find your report.
5. Right-click on your report and choose **refresh** to see it.

Generating a Comprehensive Report at Regular Intervals

Assuming you have installed a Tivoli Storage Manager Windows server and want to initiate operational reporting, you can add one or more servers to the Tivoli Storage Management Console. You must add at least one server and it must be running in order to create an operational report. You can add additional UNIX, Linux, or Windows servers per machine as needed. See "Generating a Comprehensive Monitor at Regular Intervals" on page 551 for information about generating operational monitors.

Add the Server to the Tivoli Storage Management Console

1. From **Start** on the task bar, select **Programs→Tivoli Storage Manager→Management Console**.
2. On the Management Console, right click **Tivoli Storage Manager** and select **Add TSM Computer**.
3. In the TSM Computer Resource Properties window (see Figure 79 on page 548), under the **Computer Information** tab, enter the machine name.
4. Select an **Operating System**.
5. Check **Include Report Features** to enable operational reporting features. The **TSM Web Client** is optional.

6. Check **TSM Web Administrator** to enable the Management Console to communicate with the server.
7. Specify Tivoli Storage Manager server information:
   Server name
   TCP/IP Address
   TCP/IP Port
   HTTP Port
8. Click **Add**.
9. The Logon Information window pops up. Enter the Administrator name and Password. Click **OK**
10. You should now see the information you entered such as the server name and TCP/IP address. Click **OK**.

Figure 79. Adding a Tivoli Storage Manager Server to the Management Console

Add a Report for a Server

If you want to add reports to a server, do the following:

1. On the Management Console, right click **Tivoli Storage Manager** and select **TSM Operational Reporting**.
2. In the Properties window under the **Operational Reports** tab, click **Add** to add a server. Type in a name for the report. The default name displayed is Daily Report.

   You can also highlight which server you want to use from the directory which appears after you select **TSM Operational Reporting**, click **Add**, and type in a name for the report or use the default.
3. Click **OK**.

Set Up the Report Details

After you have added a report, the Properties window opens, and a list called Report Section appears. See Figure 80 on page 549. This list shows the information

which will be in the report you generate. By default, almost every box is checked in the Report Section under the **Report Details** tab. The two unchecked boxes are the Failed Schedule Notification and the Missed Schedule Notification. You can select whatever you would like to see in the report.

If you want to view smaller sections of the report instead of all of it at once, you can check the box for **Use collapsible sections**. This may also eliminate some difficulties with displaying the reports in your Web browser.

1. Select **Activate Report** if you want the report to run on a schedule.

   Note: When a report is not automatically activated, it can still be manually run as needed.
2. Specify timing information (time to start the report, the range of hours to use, and how often to repeat running the report).
3. Enter a number of **Web versions**. A Web version refers to the number of copies of a report you want to appear in a summary Web page.

Figure 80. Tivoli Storage Management Report Details

Options for Report Details

Add an extension to your report: You can add an extension to your report by clicking **Add**. The Custom Extension Editor window opens.

1. Enter an extension name and an SQL select statement. See the *Administrator's Reference* for more information on the SELECT command.
2. To use the SQL select statement builder, click **Advanced**.
3. Click **OK**. Any extensions you add will appear at the end of the Report Section; you can select them from there.

Edit an extension in your report: You can edit an extension in your report by clicking **Settings**. The **Settings** button is enabled only for the Custom Summary, Client Schedules, and Activity Log Details Report Sections.

1. In the Advanced Customization window, you can customize your summary of daily reports by moving, inserting, editing, deleting, copying, and cutting variables. See Figure 81.

   Note: When you edit a variable, the Custom Summary Record Editor window appears and you are able to edit the SQL variable and SQL select statement, report label, and notification rule. A notification rule sets up how you are notified about server reports based on the selection criteria you establish. You should test your select statements before saving them.

2. Click **Save** or **Save As** to save your modifications.

Figure 81. Tivoli Storage Management Customizing the Custom Summary Section

Set Up E-Mail Delivery of the Report

To set up e-mail delivery of your report, select the **E-Mail Recipients** tab in the Properties window (see Figure 80 on page 549) and follow the procedure below.

1. Enter the **Recipient name** and **E-mail address**.
2. Select **Text, HTML, or URL** from the drop down list for the format of your report.

   Note: If you select **Text**, your report arrives strictly as text. Selecting **HTML** means the report includes links and is formatted as a Web page. Selecting the **URL** format will place your daily reports on a Web summary page.

3. Click **Add**. You can **Edit, Delete, Import,** or **Export** a recipient from the list. You can use the same list across multiple reports.

   Note: Exporting your recipient list will make it easier to use the same list in other reports.

Set Up Automatic Report Notification

If you would like to automatically receive notification that reports are available, click the **Automatic Notification** tab in the Properties window (see Figure 80 on page 549). Select the desired level: Failed Schedule Notification or Missed Schedule Notification.

1. To specify that the query is to return only nodes that are associated with a schedule, select **Scheduled nodes only**. Leave this box unchecked if you would like to return all nodes, including those without a schedule.
2. Click **Query**.
3. Select **Node** in the list.
4. Provide a **Contact name and E-mail address** for the recipient.
5. Select either **Missed** or **Failed** for the level of notification you want.
6. Provide any additional contact information.
7. After you update individual nodes, click **Nodes Update**. This will send updates to the Tivoli Storage Manager server. Information is then stored in the node contact field.
8. Click **OK** if you are done.

Generating a Comprehensive Monitor at Regular Intervals

Operational monitors are special types of reports, and can be scheduled to run hourly. Unlike operational reports, monitors will send you a notification only if there are issues. Monitors do not contain standard sections and typically run more quickly and frequently. Assuming you have installed a Tivoli Storage Manager Windows server and want to initiate operational monitoring, you must select at least one server and it must be running in order to create a monitor. You can add additional UNIX, Linux, or Windows servers per machine as needed.

Add a Monitor for the Server

1. On the Management Console, right click **Tivoli Storage Manager** and select **TSM Operational Reporting**. A list of servers you defined appears. Select the server you would like to monitor.
2. In the Properties window under the **Operational Monitors** tab, click **Add**.
3. Type in a name for the monitor. The default name provided is Hourly Monitor.
4. Click **OK**.

Set Up the Monitor Details

After you have added a monitor, the Properties window opens (see Figure 82 on page 552). By default, the Custom Summary and Report Timing Information are selected in the Report Section under the **Monitor Details** tab.

Note: **Settings** is enabled only for the Custom Summary.

1. Select or deselect report section check boxes you would like to include in your report.
2. Select **Activate Monitor** if you want the monitor to be run on schedule.

   Note: When a monitor is not activated on a schedule, it can still be run manually as needed.
3. Specify timing information.
4. Enter a number of **Web versions**. A Web version refers to the number of copies of the monitors you want to appear in a summary Web page.

If you want to view smaller sections of the report instead of all of it at once, you can check the box **Use collapsible sections**. This may also eliminate some difficulties with displaying the reports in your Web browser.

Figure 82. Tivoli Storage Management Monitor Details

Options for Monitor Details

Add an extension to your monitor: You can add an extension to your monitor (see Figure 82) by clicking **Add**. The Custom Extension Editor window pops up.

1. Enter an **extension name**.
2. Enter the SQL select statement. For more options, use the SQL select statement builder by clicking **Advanced**.
3. Click **OK**. Any extensions you added will appear at the end of the Report Section, allowing you to select them.

Edit an extension in your monitor: You can edit an extension in your monitor (see Figure 82) by clicking **Settings**. In the Advanced Customization window, you can customize your summary of daily reports by moving, inserting, editing, deleting, copying, and cutting variables.

Set Up E-Mail Delivery of the Monitor Results

Open the **E-Mail Recipients** tab in the Properties window (see Figure 82) and follow the procedure below to set up the e-mail delivery of the monitor.

1. Enter a **Recipient name** and **E-mail address**.
2. Select **Text, HTML, or URL** format from the drop down list.
3. Click **Add**.
4. You can **Edit, Delete, Import** or **Export** a recipient from your list. You can use the same list across multiple monitors.

Set Up Delivery of Monitor Results via Net Send

Open the **Net Send Recipients** tab in the Properties window (see Figure 82 on page 552) and follow the procedure below to set up the Net Send instant message delivery of the monitor.

Note: To enable Net Send in Windows, open a command prompt and enter:

```
net start messenger
```

1. Enter **User or Computer to notify**. If you are not sure which computer you want to notify, click **Detect** to generate a list of computers on your network.
2. Click **Add**.
3. You can optionally **Edit, Delete** or **Test** notifications.
4. Click **OK**.

Completing the Set Up

Set up a Web Summary for the Reports and Monitors

If you would like to use the Web to look up reports and monitors rather than be notified through e-mail, complete the following steps. See Figure 83 on page 554.

Note: When enabled, **Web site virtual directory, local path to store pages in,** and **Name of default web page** contain default information. Configuration for Web summary is set up from the sender to the recipient. The recipient must have valid server identification in order to receive Web report notifications.

1. On the Management Console, right click **Tivoli Storage Manager** and select **TSM Operational Reporting**.
2. Open the Properties window and under the **Summary Information** tab, select **Create a web page with summary information**. Defaults are provided. You can either accept the default settings or choose your own. If you want to customize, you can select where to store the pages and the name of the default Web page.
3. If you want to create a log file (optional), you can select **Create a Tivoli Enterprise Console log file**.
4. Click **OK** when you are finished.

Figure 83. Tivoli Storage Management Console Summary Information

Modify the Operational Reporting Service

If you would like to modify your service (optional), right click **Tivoli Storage Manager** on the Management Console, and select **TSM Operational Reporting**. See Figure 84 on page 555.

Note: The operational reporting service is set to start automatically, so you only need to complete these steps if you want to change the service to manual, stop it, or start it.

In the Properties window under the **Report Service** tab, the following options are available:

- **Startup Type**. The choices are:
  - Manual
  - Automatic

  If you select Manual, you must start the service. If you select Automatic, the service is started for you.
- **Start**. Select **Start** to activate the reporting service.
- **Stop**. Select **Stop** to stop the reporting service.
- **Review**. Select **Review** to view the reports for logs and tracing for both the Report Service and the report command line. You can also view tracing for the Management Console if you chose to enable tracing changes (see "Set up a Web Summary for the Reports and Monitors" on page 553).

If you want to save the changes you made, click **OK**. If you want to exit the window without saving any changes, click **Cancel**.

Note: By default, the Report service and the Report command line options are selected.

Figure 84. Tivoli Storage Management Console Report Service

Viewing Reports and Other Features from the Management Console

On the Tivoli Storage Manager Management Console, you can view and work with reports and monitors, the activity log, scheduled events, sessions, and server options for operational reporting.

Viewing Operational Reports

In an open Management Console, expand the console tree of the Tivoli Storage Manager server for which you want to view reports and monitors. This will allow you to access all of the features available for your defined server.

To view operational reports, click on the report name you want to see. The report is displayed in the main panel of the Management Console. If you want a current report, you can generate an instant report using either the current time or the scheduled time. To use this feature, right click on the report you would like to generate. A menu of the following options is displayed:

- **Clear**. This clears the current report displayed, but does not delete the report.
- **Refresh Using Current® Time**. This report captures server data from the exact time you refreshed the report back to the number of hours you chose to cover specified in the report details. For example, if you refreshed the report at 5:00 p.m. on 11/23/03 and you chose to have the report cover a 24-hour period, you would generate a report that spanned from 5:00 p.m. on 11/22/03 to 5:00 p.m. on 11/23/03.
- **Refresh Using Scheduled Time**. This report captures server data starting from the scheduled time to start, back to the number of hours you specified in the report details. For example, if you scheduled your report to start at 1:00 p.m.

and cover an eight-hour period, then you would generate a report that spanned from 5:00 a.m. to 1:00 p.m., no matter what time you issued the refresh.

- **Properties**. This opens the Properties window for the report. From this window, you can modify Report Details, Automatic Notification for nodes and add or remove E-mail Recipients.
- **Remove**. This deletes the report. You will no longer be able to access the report and must start over if you want to generate a report.
- **Send E-mail**. This feature opens an E-mail composer with the appropriate report already attached, allowing you to quickly add comments and send the report to addresses in the E-mail Recipients list.
- **View**. This allows you to customize the view of the Management Console.
- **Help**. This displays the help for the Management Console.

Viewing Operational Monitors

In an open Management Console, expand the console tree of the Tivoli Storage Manager server for which you want to view reports and monitors. This allows you to access all of the features available for your defined server. To view operational monitors, click on the monitor name you want to see. The monitor information is displayed in the main panel of the Management Console.

If the time you chose to monitor has not passed or you want an updated monitor, you can generate an instant monitor using either the current time or scheduled time. This allows you to generate monitors at any given time period. To use this feature, right click on the monitor you would like to generate. A menu of the following options will be displayed:

- **Clear**. This clears the current report displayed, but does not delete the report.
- **Refresh Using Current Time**. This monitor captures server data from the exact time you refreshed the monitor backwards to the number of hours you specified in Monitor Details. For example, if you refreshed the monitor at 5:00 p.m. on 11/23/03 and you chose to have the monitor cover a 24-hour period, you would generate a monitor that spanned from 5:00 p.m. on 11/22/03 to 5:00 p.m. on 11/23/03.
- **Refresh Using Scheduled Time**. This monitor captures server data starting from the scheduled minute to start, back to the number of hours you specified in Monitor Details. For example, if you scheduled your monitor to start at Minute 30, covering an eight-hour period that repeats every 2 hours, and you refreshed the monitor at 6:45 p.m. then you would generate a monitor that spanned from 10:30 am to 6:30 p.m., no matter what time you issued the refresh.
- **Properties**. This opens the Properties window for the monitor. From this window, you can modify Monitor Details and add or remove E-mail and Net Send Recipients.
- **Remove**. This deletes the monitor. You are no longer able to access this monitor and must start over if you want to generate monitors.
- **Send E-mail**. This feature opens an E-mail composer with the appropriate monitor already attached, allowing you to quickly add comments and send the monitor to addresses in the E-mail Recipients list.
- **View**. This allows you to customize the view of the Management Console.
- **Help**. This displays the help for the Management Console.

Viewing the Activity Log

The server activity log contains messages normally sent to the server console during server operation. For operational reporting, you may only want to know

what errors or warnings have occurred during server operation. The following features are available by right clicking on **Activity Log** in the main panel of the Management Console:

- **Clear**. This clears the activity log displayed in the main panel of the Management Console.
- **Refresh**. This refreshes the activity log and displays the output in the main panel of the Management Console.
- **Properties**. This opens the properties window for the activity log.
  - **Activity Log** tab. From this tab, you can enable automatic refresh and change the refresh rate to an integer between 30 and 1440 minutes. You can also use either a relative or absolute time to query the activity log. Relative time will allow you to display messages that have occurred within the last 0 to 365 days, 0 to 23 hours and 0 to 59 minutes. If you want to be more specific in the activity log query, use absolute time. This will allow you to capture an exact time period by choosing a beginning date and time and an ending date and time for the act log messages.
  - **Search** tab. You can enable pattern matching to search for a specific Message Number, Search string, Originator (All, Client, or Server), Session, Node name, Client owner or Schedule name.
  - **Filter** tab. This allows you to enable message filtering. You can add messages you do not want displayed in your activity log query by either typing in the actual message or the message name. You can also narrow messages down by category, by choosing not to display information, warning or error messages. By default only the Information box is checked.
- **View**. This allows you to customize the view of the Management Console.
- **Help**. This displays the help for the Management Console.

Viewing Scheduled Events

In the main panel of the Management Console, you can use the scheduled events feature to see what events are scheduled.

Note: Future schedule events will not be displayed. This query only provides information on Started, Missed, Pending, Completed or Failed schedules.

You can use this feature by right clicking **Scheduled Events**. A menu of the following options is displayed:

- **Clear**. This clears the scheduled events displayed in the main panel of the Management Console.
- **Refresh**. This refreshes the scheduled events query and displays the output in the main panel of the Management Console.
- **Properties**. From here, you can enable automatic refresh and change the refresh rate to an integer between 5 and 1440 minutes. You can choose what event types you want in the query. Possible event types are Started, Missed, Pending, Completed and Failed. You can also use either a relative or absolute time to query scheduled events. Relative time will allow you to display events that have occurred within the last 0 to 365 days, 0 to 23 hours and 0 to 59 minutes. If you want to be more specific in the query, use absolute time. This allows you to capture an exact time period by choosing a beginning date and time and an ending date and time for a query.
- **View**. This allows you to customize the view of the Management Console.
- **Export List**. This exports the scheduled events displayed on the Management Console to a file. This option is displayed only after you use the refresh function.

- **Help**. This displays the help for the Management Console.

Viewing Sessions

This displays the current sessions in the main panel of the Management Console. You can use this feature by right clicking on **Sessions**. A menu of the following options will be displayed:

- **Clear**. This clears the sessions displayed in the main panel of the Management Console.
- **Refresh**. This refreshes the current sessions and displays the output in the main panel of the Management Console.
- **Properties**. From here, you can enable automatic refresh and change the refresh rate to an integer between 1 and 1440 minutes.
- **View**. This allows you to customize the view of the Management Console.
- **Export List**. This command exports the session information displayed on the Management Console to a file. This option will only be displayed after you have used the refresh function.
- **Help**. This displays the help for the Management Console.

Viewing Server Options

This displays all of the server options in the main panel of the Management Console. You can use this feature by right clicking on **Server Options**. A menu of the following options will be displayed:

- **Clear**. This clears the options displayed in the main panel of the Management Console.
- **Refresh**. This refreshes the option's query and displays the output in the main panel of the Management Console.
- **View**. This allows you to customize the view of the Management Console.
- **Export List**. This command exports the server options information displayed on the Management Console to a file. This option will only be displayed after you have used the refresh function.
- **Help**. This displays the help for the Management Console.

Generating Reports and Monitors from the Command Line

The operational reporting command line allows you to generate any report or monitor which exists in the Tivoli Storage Manager Management Console. Some advantages to using the command line are:

- You can include the command in your own scripts.
- Using the command line with the Windows Scheduler (available in the Windows Control Panel) can help refine control of scheduling options. If you create a Windows command file for operational reporting, you can use the Scheduler to schedule the command file and you have multiple options for when to run it.
- It can help with debugging. In case a report does not run from the Management Console, the command line may provide more detailed information about the problem.

To obtain operational reports from the command line, open a Windows Command Prompt and navigate to the server console directory.

Note: Before you can generate a report or monitor from the command line it must first be defined by using the Tivoli Storage Manager Management Console. After a report or monitor has been defined in the Management Console, it can be run from the command line using the commands described below.

The syntax for the command is:

```
tsmrept [options]
```

where the options are described in Table 34.

Notes:

1. Option names cannot be abbreviated.
2. Option values with embedded spaces must be entered within quotation marks.

Table 34. Options for the tsmrept command

Option	Description	Default
-type={report | monitor | list}	Produce a report, monitor, or list	report
-computer=*computer_name*	Server computer name	localhost
-instance=*instance_name*	Server instance name	server1
-name=*report_name*	Report or monitor name	"Daily Report"
-starthour={now | 0..23}	Starting hour	now
-startmin={now | 0..59}	Starting minute	now
-reporttemplate=*name*.xml	Customer report or monitor template name	default_rep_eng.xml
-remove={no | yes}	Delete the specified report	no
-notifyrecipients={no | yes}	E-mail the report to the recipients	no
-tcpserveraddress=*ip_address*	Tivoli Storage Manager server TCP/IP name or address	localhost
-tcpport=*port_number*	Tivoli Storage Manager server TCP/IP port	1500
-id=*administrator_ID*	Tivoli Storage Manager administrator ID	none
-pa=*administrator_password*	Tivoli Storage Manager administrator password	none
-outputformat= [filetext, filehtml, emailtext, emailhtml, screentext, screenhtml]	Type or types of output to produce	screentext,filehtml

Table 34. Options for the tsmrept command (continued)

Option	Description	Default
-sections= [summary, schedadmin, schedclient, loadsession, loadmount, loadmigr, loadrecl, loadexpiration, loadbackdb, loadbackstg, summarynodes, detailsactlog, summarymissedfiles, detailsmissedfiles, detailssessions, notifyfailed, notifymissed]	Section or sections to include	all
-?	Display the command help	none
-help	Display the command help	none

Examples Default values exist for each command line option, however, it is better to explicitly provide at least the -computer, -instance, -type, and -name options. To display this information for each defined report and monitor, issue the following command:

```
tsmrept -type=list
```

The output will be similar to this:

```
-COMPUTER=tsmserv -INSTANCE="server1" -TYPE=report -NAME="daily report"
-COMPUTER=tsmserv -INSTANCE="server2" -TYPE=report -NAME="daily report"
-COMPUTER=tsmserv -INSTANCE="server2" -TYPE=report -NAME="drm report"
-COMPUTER=tsmserv -INSTANCE="server2" -TYPE=monitor -NAME="hourly monitor"
-COMPUTER=ss2 -INSTANCE="server1" -TYPE=report -NAME="daily report"
-COMPUTER=ss2 -INSTANCE="server1" -TYPE=report -NAME="kathy's report"
-COMPUTER=ss2 -INSTANCE="server1" -TYPE=report -NAME="weekly report"
-COMPUTER=sunny -INSTANCE="server1" -TYPE=report -NAME="daily report"
```

You can then type in `tsmrept` on the command line and copy and paste any one of the lines above. For example, to produce the drm report, you would issue the following command:

```
tsmrept -COMPUTER=tsmserv -INSTANCE="server2" -TYPE=report -NAME="drm report"
```

The report is displayed on the screen as it is being produced. By default, the command creates an HTML and a text version of the report in the console directory. The file names consist of the computer, instance, type, and report name. In this example, in addition to displaying the text on the screen, the following files would be produced:

- tsmservserver2reptdrmreport.htm
- tsmservserver2reptdrmreport.txt

More Examples

- To get a list of defined reports and monitors:

```
cd /d "c:\program files\tivoli\tsm\console"
 tsmrept -type=list
```

- To generate a default report:

  ```
  tsmrept
  ```

- To generate a report only for computer ″tsmserver2″:

  ```
  tsmrept -computer=tsmserver2
  ```

- To generate a report:

  ```
  tsmrept -id=administrator -pa=password -tcpport=1501
  ```

- To display help for the `tsmrept` command:

  ```
  tsmrept /?
  ```

Chapter 21. Working with a Network of IBM Tivoli Storage Manager Servers

You may have a number of Tivoli Storage Manager servers in your network, at the same or different locations. Tivoli Storage Manager provides functions to help you configure, manage, and monitor the servers connected to a network. An administrator working at one Tivoli Storage Manager server can work with Tivoli Storage Manager servers at other locations around the world.

See the following sections:

Concepts:
"Concepts for Working with a Network of Servers"

Tasks:
"Planning for Enterprise Administration" on page 567
"Setting Up Communications Among Servers" on page 568
"Setting Up an Enterprise Configuration" on page 575
"Performing Tasks on Multiple Servers" on page 596
"Using Virtual Volumes to Store Data on Another Server" on page 601

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Concepts for Working with a Network of Servers

To manage a network of servers, you can use the following capabilities of IBM Tivoli Storage Manager:

- Configure and manage multiple servers with enterprise configuration.

  Distribute a consistent configuration for Tivoli Storage Manager servers through a configuration manager to managed servers. By having consistent configurations, you can simplify the management of a large number of servers and clients.
- Perform tasks on multiple servers by using command routing, enterprise logon, and enterprise console.
- Send server and client events to another server for logging.
- Monitor many servers and clients from a single server.
- Store data on another server by using virtual volumes.

In a network of Tivoli Storage Manager servers, a server can play a number of different roles. For example, a server may send volumes to be archived on another server and also receive routed commands from another server. In the following

© Copyright IBM Corp. 1993, 2005

descriptions, when a server sends data it 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.

Configuring and Managing Servers: Enterprise Configuration

The enterprise configuration functions of the IBM Tivoli Storage Manager 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 the other servers. You can make changes to configurations and have the changes automatically distributed.

Figure 85 on page 565 illustrates a simple configuration. To use enterprise configuration, you first select the Tivoli Storage Manager server that is to act as the *configuration manager*. You may want to dedicate a new server for this purpose. At the configuration manager, you define the details of the server configurations that you want to distribute. For example:

- You set up backup and archive policies and client option sets
- You designate one or more administrators to have access to the servers, and control their authority levels
- You 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*, you 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, you 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.

See "Setting Up an Enterprise Configuration" on page 575 for details.

Figure 85. Enterprise Configuration

Performing Tasks on Multiple Servers

You can use the Administration Center to access any Version 5.3 servers in your enterprise. This allows you to work with configuration managers and managed servers from a single interface. See "Managing Servers with the New Administration Center" on page 36 for more information.

From the command line of the Administration Center or from the command-line administrative client, you can also 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. See "Routing Commands" on page 597.

To make routing commands easier, you can define a server group that has servers as members. See "Setting Up Server Groups" on page 599. Commands that you route to a server group are sent to all servers in the group.

Central Monitoring

Tivoli Storage Manager provides you with several ways to centrally monitor the activities of a server network:

- Enterprise event logging, in which events are sent from one or more of servers to be logged at an event server. See "Enterprise Event Logging: Logging Events to Another Server" on page 538 for a description of the function and "Setting Up Communications for Enterprise Configuration and Enterprise Event Logging" on page 568 for communications set up.

- Allowing designated administrators to log in to any of the servers in the network with a single user ID and password. See "Working with Multiple Servers Using the Administration Center" on page 596.
- Routing query commands to one or more of the servers in the network. See "Routing Commands to One or More Servers" on page 597 for a description of the function and "Setting Up Communications for Enterprise Configuration and Enterprise Event Logging" on page 568 for communications set up.

Storing Data 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 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. See "Licensing IBM Tivoli Storage Manager" on page 457.

Example Scenarios

The functions for managing multiple servers can be applied in many ways. Here are just two scenarios to give you some ideas about how you can put the functions to work for you:

- 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.

Managing IBM Tivoli Storage Manager Servers from One Location

Enterprise management allows you to set up and manage the servers in your network from one location, the enterprise console. For example, suppose that you are an administrator 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 do 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.

Managing Servers from Any Server

Enterprise management allows you to 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.

Planning for Enterprise Administration

To take full advantage of the functions of Enterprise Administration, you should decide on the following:

- 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

This section describes how to set up communications for enterprise configuration, enterprise event logging, and command routing. Communication setup for server-to-server virtual volumes is described in "Setting Up Source and Target Servers for Virtual Volumes" on page 602.

When you set up communications among servers for any purpose, ensure that servers have unique names. See "Setting the Server Name" on page 468 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/IP, is identical. The examples shown here apply to both functions. If you are set up for one, you are set up for the other. However, be aware that the configuration manager and event server are not defined simply by setting up communications. You must identify a server as a configuration manager (SET CONFIGMANAGER command) or an event server (DEFINE EVENTSERVER command). Furthermore, a configuration manager and an event server can be the same server or different servers.

Enterprise configuration
: Each managed server must be defined to the configuration manager, and the configuration manager must be defined to each managed server.

Enterprise event logging
: Each server sending events to an event server must be defined to the event server, and the event server must be defined to each source server.

The following examples of setting up communications could be used to create these configurations:

- A server named HEADQUARTERS as a configuration manager and two servers, MUNICH and STRASBOURG, as managed servers.
- HEADQUARTERS as an event server and MUNICH and STRASBOURG as source servers.

For a pair of servers to communicate with each other, each server must be defined to the other. For example, if a configuration manager manages three managed servers, there are three server pairs. You can issue separate definitions from each server in each pair, or you can "cross define" a pair in a single operation. Cross

definition can be useful in large or complex networks. The following scenarios and accompanying figures illustrate the two methods.

Using separate definitions — Follow this sequence:

1. **On MUNICH**: Specify the server name and password of MUNICH.

   On STRASBOURG: Specify the server name and password of STRASBOURG.

   On HEADQUARTERS: Specify the server name and password of HEADQUARTERS.
2. **On HEADQUARTERS**: Define MUNICH (whose password is BERYL and whose address is 9.115.2.223:1919) and STRASBOURG (whose password is FLUORITE and whose address is 9.115.2.178:1715).

   On MUNICH and STRASBOURG: Define HEADQUARTERS (whose password is AMETHYST and whose address is 9.115.4.177:1823).

Figure 86 shows the servers and the commands issued on each:

Figure 86. 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 87 shows the servers and the commands issued on each:

Figure 87. Communication Configuration with Cross Definition

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 will fail. 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

This section describes how to set up communications for command routing. You must define the target servers to the source servers, and the same administrator must be registered on all servers. Using enterprise configuration, you can easily distribute the administrator information to all the servers.

Note: You must be registered as an administrator with the same name and password on the source server and all target servers. The privilege classes do not need to be the same on all servers. However, to successfully route a command to another server, an administrator must have the minimum required privilege class for that command on the server from which the command is being issued.

For command routing in which one server will always be the sender, you would only define the target servers to the source server. If commands can be routed from any server to any other server, each server must be defined to all the others.

Only One Source Server

The example in this section shows how to set up communications for administrator HQ on the server HEADQUARTERS who will route commands to the servers MUNICH and STRASBOURG. Administrator HQ has the password SECRET and has system privilege class. Here is the procedure:

- **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 serverpassword set in the server definition.

- **On MUNICH and STRASBOURG** Register administrator HQ with the required privilege class on each server:

  ```
  register admin hq secret
  grant authority hq classes=system
  ```

Note: If your server network is using enterprise configuration, you can automate the preceding operations. You can distribute the administrator and server lists to MUNICH and STRASBOURG. In addition, all server definitions and server groups are distributed by default to a managed server when it first subscribes to any profile on a configuration manager. Therefore, it receives all the server definitions that exist on the configuration manager, thus enabling command routing among the servers.

Multiple Source Servers

The examples in this section show 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 must define all the servers to each other. 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.

Separate Definitions: Follow this sequence:

1. **On MUNICH:** Specify the server name and password of MUNICH. Register administrator HQ and grant HQ system authority.

   On STRASBOURG: Specify the server name and password of STRASBOURG. Register administrator HQ and grant HQ system authority.

   On HEADQUARTERS: Specify the server name and password of HEADQUARTERS. Register administrator HQ and grant HQ system authority.
2. **On HEADQUARTERS:** Define MUNICH (whose password is BERYL and whose address is 9.115.2.223:1919) and STRASBOURG (whose password is FLUORITE and whose address is 9.115.2.178:1715).

   On MUNICH: Define HEADQUARTERS (whose password is AMETHYST and whose address is 9.115.4.177:1823) and STRASBOURG.

   On STRASBOURG: Define HEADQUARTERS and MUNICH.

Figure 88 on page 573 shows the servers and the commands issued on each:

Figure 88. Communication Configuration with Separate Server Definitions

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. Register administrator HQ and grant HQ system authority.

   On STRASBOURG: Specify the server name, password, and high and low level addresses of STRASBOURG. Specify that cross define is permitted. Register administrator HQ and grant HQ system authority.

   On HEADQUARTERS: Specify the server name, password, and high and low level addresses of HEADQUARTERS. Register administrator HQ and grant HQ system authority.
2. **On HEADQUARTERS:** Define MUNICH and STRASBOURG, specifying that cross define should be done.
3. **On MUNICH:** Define STRASBOURG, specifying that cross define should be done.

Note: If your server network is using enterprise configuration, you can automate the preceding operations. You can distribute the administrator lists and server lists to MUNICH and STRASBOURG. In addition, all server definitions and server groups are distributed by default to a managed server when it first subscribes to any profile on a configuration manager. Therefore, it receives all the server definitions that exist on the configuration manager, thus enabling command routing among the servers.

Figure 89 shows the servers and the commands issued on each:

Figure 89. Communication Configuration with Cross Definitions

Updating and Deleting Servers

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. See "Setting Up Server Groups" on page 599 for information about server groups.

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.

Setting Up an Enterprise Configuration

After you set up server communication as described in "Setting Up Communications for Enterprise Configuration and Enterprise Event Logging" on page 568, you set up the configuration manager and its profiles. With the profiles, you designate the configuration information that can be distributed to managed servers. Then you can set up other servers as managed servers. The managed servers receive configuration information through subscriptions to profiles on the configuration manager. Each managed server stores the distributed information as managed objects in its database. Managed servers receive periodic updates of the configuration information from the configuration manager, or an administrator can trigger an update by command.

You can distribute the following configuration information from a configuration manager to managed servers:

- Administrators, including authorities for them

- Policy objects, including policy domains, and the policy sets, management classes, copy groups and client schedules associated with them.
- Administrative command schedules
- Tivoli Storage Manager server scripts
- Client option sets
- Server definitions
- Server groups

For details on the attributes that are distributed with these objects, see "Associating Configuration Information with a Profile" on page 580.

"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.

Enterprise Configuration Scenario

To illustrate how you might use these functions, suppose that your enterprise has offices around the world, with one or more Tivoli Storage Manager servers at each location. To make managing these servers easier, you want to control the configuration of all Tivoli Storage Manager servers from one Tivoli Storage Manager server in the headquarters office. Figure 90 shows the hierarchy that you want to set up.

Figure 90. 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

Figure 91 shows the specific commands needed 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 91. Setting Up a Configuration Manager

1. Decide whether to use the existing Tivoli Storage Manager server in the headquarters office as the configuration manager or to install a new Tivoli Storage Manager server on a system.
2. Set up the communications among the servers. See "Setting Up Communications Among Servers" on page 568 for details.
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. For more information, see "Creating the Default Profile on a Configuration Manager" on page 579.
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. See "Defining a Server Group and Members of a Server Group" on page 599 for details.

   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.

   For details, see "Creating and Changing Configuration Profiles" on page 580.

Setting Up a Managed Server

Figure 92 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 92. 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.

A server becomes a managed server when that server first subscribes to a profile on a configuration manager.

1. Query the server to look for potential conflicts.

See “Getting Information about Profiles” on page 587. 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. See “Associating Configuration Information with a Profile” on page 580 for details on the exceptions.

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. See “Subscribing to a Profile” on page 589.

   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. See “Defining or Updating Primary Storage Pools” on page 220 or “Renaming a Storage Pool” on page 300.

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 473.

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. See “Refreshing Configuration Information” on page 593.

Creating the Default Profile on a Configuration Manager

Task	Required Privilege Class
Set up a server as a configuration manager	System

To set up one Tivoli Storage Manager server as the source for configuration information for other servers, you 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.

Enter the following command:

```
set configmanager on
```

When a server becomes a configuration manager, the server automatically creates a default profile named DEFAULT_PROFILE. The default profile contains any

definitions of servers and server groups that exist on the configuration manager. You can change or delete the profile named DEFAULT_PROFILE.

When a managed server first subscribes to a profile on a configuration manager, the configuration manager automatically also subscribes the managed server to the profile named DEFAULT_PROFILE, if it exists. The information distributed via this profile gets refreshed in the same way as other profiles. This helps ensure that all servers have a consistent set of server and server group definitions for all servers in the network.

If you do not change the DEFAULT_PROFILE, whenever a managed server subscribed to the DEFAULT_PROFILE profile refreshes configuration information, the managed server receives definitions for all servers and server groups that exist on the configuration manager at the time of the refresh. As servers and server groups are added, deleted, or changed on the configuration manager, the changed definitions are distributed to subscribing managed servers.

Creating and Changing Configuration Profiles

You create configuration profiles on a configuration manager, which distributes the information associated with the profiles to any managed server that subscribes to those profiles. Creating a configuration profile includes these steps:

1. Defining the profile
2. Associating the configuration information with the profile

Once you define the profile and its associations, a managed server can subscribe to the profile and obtain the configuration information.

After you define a profile and associate information with the profile, you can change the information later. While you make changes, you can lock the profiles to prevent managed servers from refreshing their configuration information. To distribute the changed information associated with a profile, you can unlock the profile, and either wait for each managed server to refresh its configuration to get the changed information or notify each managed server to refresh its configuration. The following sections provide information on each of these tasks.

Defining the Profile

Task	Required Privilege Class
Define profiles	System

When you define the profile, you select the name and can include a description. For example, to define a profile named ALLOFFICES, enter the following command:

```
define profile alloffices
 description='Configuration to be used by all offices'
```

Associating Configuration Information with a Profile

Task	Required Privilege Class
Define profile associations	System

After you define a profile, you associate the configuration information that you want to distribute via that profile. You can associate the following configuration information with a profile:

- Tivoli Storage Manager administrators, including their authorities. See "Configuration Information for Tivoli Storage Manager Administrators" for tips.
- Policy domains. See "Configuration Information for Policy Domains" on page 582 for tips.
- Servers definitions. See "Configuration Information for Servers and Server Groups" on page 582 for tips.
- Server groups. See "Configuration Information for Servers and Server Groups" on page 582 for tips.
- Administrative command schedules. See "Configuration Information for Administrative Command Schedules" on page 583 for tips.
- Tivoli Storage Manager server scripts. See "IBM Tivoli Storage Manager Server Scripts" on page 477 for tips.
- Client option sets. See "Managing Client Option Sets" on page 342 for tips.

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.

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 could 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.

Configuration Information for Policy Domains

When you point to a policy domain in a profile, the configuration information that will be sent to the managed servers includes the policy domain itself, and all policy sets with their associated management classes, copy groups, and client schedules in the domain. A configuration manager does *not* distribute the following:

- An active policy set and any of its associated management classes, copy groups, and client schedules. On each managed server, you must activate a policy set in each managed policy domain.
- Associations between clients and schedules. To have clients in a managed policy domain run client schedules, you must associate the clients with the schedules on the managed server.
- Client actions, which are schedules created by using the DEFINE CLIENTACTION command. On each managed server, you can define and delete client actions, even if the corresponding domain is a managed object.
- Definitions for any storage pools identified as destinations in the policy. Definitions of storage pools and device classes are not distributed by a configuration manager.

  Policy domains can refer to storage pool names in the management classes, backup copy groups, and archive copy groups. As you set up the configuration information, consider whether managed servers already have or can set up or rename storage pools with these names.

A subscribing managed server may already have a policy domain with the same name as the domain associated with the profile. The configuration refresh overwrites the domain defined on the managed server unless client nodes are already assigned to the domain. Once the domain becomes a managed object on the managed server, you can associate clients with the managed domain. Future configuration refreshes can then update the managed domain.

If nodes are assigned to a domain with the same name as a domain being distributed, the domain is not replaced. This safeguard prevents inadvertent replacement of policy that could lead to loss of data. To replace an existing policy domain with a managed domain of the same name, you can do the following steps on the managed server:

1. Copy the domain.
2. Move all clients assigned to the original domain to the copied domain.
3. Trigger a configuration refresh.
4. Activate the appropriate policy set in the new, managed policy domain.
5. Move all clients back to the original domain, which is now managed.

Configuration Information for Servers and Server Groups

The DEFAULT_PROFILE that is automatically created on a configuration manager already points to all servers and server groups defined to that server. If you leave the DEFAULT_PROFILE intact, you do not need to include servers or server

groups in any other profile. Any servers and server groups that you define later are associated automatically with the default profile and the configuration manager distributes the definitions at the next refresh.

For a server definition, the following attributes are distributed:

- Communication method
- TCP/IP address (high-level address)
- 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 35 summarizes what happens when servers or server groups being distributed have the same names as servers or server groups on the managed server.

Table 35. 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.

Configuration Information for Administrative Command Schedules

When the configuration manager distributes administrative schedules, the schedules are not active on the managed server. An administrator on the managed server must activate any managed schedules to have them run on the managed server.

A configuration refresh does not replace or remove any local schedules that are active on a managed server. However, a refresh can update an active schedule that is already managed by a configuration manager.

Changing a Profile

Task	Required Privilege Class
Define profile associations	System
Update profiles	System

You can change a profile and its associated configuration information. 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. See "Removing Configuration Information from Managed Servers" on page 585 for details.

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'
```

Preventing Access to Profiles While You Make Changes

If you are making changes to a profile, you may want to prevent any subscribing managed server from refreshing its configuration information until you are done. You can lock the profile to prevent access to the profile by a managed server. Locking prevents a managed server from getting information that is incomplete because you are still making changes.

Task	Required Privilege Class
Lock and unlock profiles	System

For example, to lock the ALLOFFICES profile for two hours (120 minutes), enter the following command:

```
lock profile alloffices 120
```

You can let the lock expire after two hours, or unlock the profile with the following command:

```
unlock profile alloffices
```

Distributing Changed Configuration Information

To distribute the changed profile, you can wait for each managed server to refresh its configuration to get the changed information, or you can notify each managed server from the configuration manager. Managed servers refresh profile

information on a configuration refresh period. See "Refreshing Configuration Information" on page 593 for how to set this period.

Task	Required Privilege Class
Notify servers that subscribe to profiles to refresh configuration information	System

From the configuration manager, to notify all servers that are subscribers to the ALLOFFICES profile, enter the following command:

```
notify subscribers profile=alloffices
```

The managed servers then refresh their configuration information, even if the time period for refreshing the configuration has not passed.

Removing Configuration Information from Managed Servers

Task	Required Privilege Class
Delete profile associations	System

To remove configuration information from managed servers, you can do one of two things. You candelete the association of the object with the profile, or delete the object itself from the configuration manager.

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. See "Deleting Subscriptions" on page 592.

On the configuration manager, you can delete the association of objects with a profile. For example, you may want to remove some of the administrators that are associated with the ADMINISTRATORS profile. With an earlier command, you had included all administrators defined on the configuration manager (by specifying ADMINS=*). To change the administrators included in the profile you must first delete the association of all administrators, then associate just the administrators that you want to include. Do the following:

1. Before you make these changes, you may want to prevent any servers from refreshing their configuration until you are done. Enter the following command:

   ```
   lock profile administrators
   ```

2. Now make the change by entering the following commands:

   ```
   delete profassociation administrators admins=*

   define profassociation administrators
   admins=admin1,admin2,admin3,admin4
   ```

3. Unlock the profile:

   ```
   unlock profile administrators
   ```

4. You may want to notify any managed server that subscribes to the profile so that servers refresh their configuration information:

   ```
   notify subscribers profile=administrators
   ```

When you delete the association of an object with a profile, the configuration manager no longer distributes that object via the profile. Any managed server subscribing to the profile deletes the object from its database when it next contacts

the configuration manager to refresh configuration information. However, a managed server does not delete the following objects:

- An object that is associated with another profile to which the server subscribes.
- A policy domain that has client nodes still assigned to it. To delete the domain, you must assign the affected client nodes to another policy domain on the managed server.
- An administrator that currently has a session open with the server.
- An administrator that is the last administrator with system authority on the managed server.

  Also the managed server does not change the authority of an administrator if doing so would leave the managed server without any administrators having the system privilege class.

  You can avoid both problems by ensuring that you have locally defined at least one administrator with system privilege on each managed server.
- An administrative schedule that is active. To remove an active schedule, you must first make the schedule inactive on the managed server.
- A server definition for a server that currently has an open connection from the managed server.
- A server definition that is specified in the definition of a device class that is a SERVER device type.
- A server definition that is the definition for the event server for the managed server.

If you no longer need an object defined on the configuration manager itself or on any managed server, you can delete the object itself. Deleting the object itself from the configuration manager has an effect similar to deleting the association of that object with the profile. Theconfiguration manager no longer distributes that object, and a managed server attempts to delete the object from its database when it refreshes configuration information.

Deleting Profiles

Task	Required Privilege Class
Delete profiles	System

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, you should first delete the subscriptions on each managed server. 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
```

See “Deleting Subscriptions” on page 592 for more details about deleting subscriptions on a managed server.

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.

Getting Information about Profiles

Task	Required Privilege Class
Request information about profiles	Any administrator

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. 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

Task	Required Privilege Class
Define subscriptions to profiles	System
Set the period for configuration refreshes	System

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.

Notes:

1. Unless otherwise noted, the commands in this section would be run on a managed server:
2. An administrator at the managed server could issue the commands.
3. You could log in from the enterprise console and issue them.
4. 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 may 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.

A Subscription Scenario

This section describes a typical scenario in which a server subscribes to a profile on a configuration manager, HEADQUARTERS. In this scenario an administrator for the HEADQUARTERS server has defined three profiles, ADMINISTRATORS, ENGINEERING, and MARKETING, each with its own set of associations. In addition, DEFAULT_PROFILE was automatically defined and contains only the server and server group definitions defined on the HEADQUARTERS server. An administrator for HEADQUARTERS has given you the names of the profiles that you should be using. To subscribe to the ADMINISTRATORS and ENGINEERING profiles and keep them current, perform the following steps:

1. Display the names of the objects in the profiles on HEADQUARTERS.

   You might want to perform this step to see if the object names on the profiles are used on your server for any objects of the same type. Issue this command:

   ```
   query profile * server=headquarters format=detailed
   ```

   You might want to get detailed information on some of the objects by issuing specific query commands on either your server or the configuration manager.

   Note: If any object name matches and you subscribe to a profile containing an object with the matching name, the object on your server will be replaced, with the following exceptions:

   - A policy domain is not replaced if the domain has client nodes assigned to it.
   - An administrator with system authority is not replaced by an administrator with a lower authority level if the replacement would leave the server without a system administrator.
   - The definition of a server is not replaced unless the server definition on the managed server allows replacement.
   - A server with the same name as a server group is not replaced.
   - A locally defined, active administrative schedule is not replaced

2. Subscribe to the ADMINISTRATORS and ENGINEERING profiles.

   After the initial subscription, you do not have to specify the server name on the DEFINE SUBSCRIPTION commands. If at least one profile subscription already exists, any additional subscriptions are automatically directed to the same configuration manager. Issue these commands:

   ```
   define subscription administrators server=headquarters

   define subscription engineering
   ```

   The object definitions in these profiles are now stored on your database. In addition to ADMINISTRATORS and ENGINEERING, the server is also subscribed by default to DEFAULT_PROFILE. This means that all the server and server group definitions on HEADQUARTERS are now also stored in your database.

3. Set the time interval for obtaining refreshed configuration information from the configuration manager.

If you do not perform this step, your server checks for updates to the profiles at start up and every 60 minutes after that. Set up your server to check HEADQUARTERS for updates once a day (every 1440 minutes). If there is an update, HEADQUARTERS sends it to the managed server automatically when the server checks for updates.

```
set configrefresh 1440
```

Note: You can initiate a configuration refresh from a managed server at any time. To initiate a refresh, simply reissue the SET CONFIGREFRESH with any value greater than 0. The simplest approach is to use the current setting:

```
set configrefresh 1440
```

Querying Subscriptions

Task	Required Privilege Class
Request information about subscriptions	Any administrator
Request information about profiles	Any administrator

From time to time, you may want to see what profiles a server is subscribed to. You may also want to see the last time that the configuration associated with that profile was successfully refreshed on your server. 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

Task	Required Privilege Class
Delete subscriptions to profiles	System

If you decide that a server no longer needs to subscribe to a profile, you can delete the subscription. 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

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)

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.

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.

Handling Problems with Configuration Refresh

To monitor for any problems during a configuration refresh, you can watch the server console or activity log of the managed server. One problem that may occur is that the refresh process may 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. See "Associating Configuration Information with a Profile" on page 580 for details on when objects cannot be distributed.

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.

Returning Managed Objects to Local Control

You may 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 do this from the configuration manager or from the managed servers.

To do this from the configuration manager, you do not simply delete the association of the object from the profile, because that would cause the object to be deleted from subscribing managed servers. To ensure the object remains in the databases of the managed servers as a locally managed object, you can copy the current profile, make the deletion, and change the subscriptions of the managed servers to the new profile.

For example, servers are currently subscribed to the ENGINEERING profile. The ENGDOMAIN policy domain is associated with this profile. You want to return control of the ENGDOMAIN policy domain to the managed servers. You can do the following:

1. Copy the ENGINEERING profile to a new profile, ENGINEERING_B:
   ```
   copy profile engineering engineering_b
   ```
2. Delete the association of the ENGDOMAIN policy domain from ENGINEERING_B:
   ```
   delete profassociation engineering_b domains=engdomain
   ```
3. Use command routing to delete subscriptions to the ENGINEERING profile:
   ```
   americas,europe,asia: delete subscription engineering
   discardobjects=no
   ```
4. Delete the ENGINEERING profile:
   ```
   delete profile engineering
   ```
5. Use command routing to define subscriptions to the new ENGINEERING_B profile:
   ```
   americas,europe,asia: define subscription engineering_b
   ```

To return objects to local control when working on a managed server, you can delete the subscription to one or more profiles. When you delete a subscription, you can choose whether to delete the objects associated with the profile. To return objects to local control, you do not delete the objects. For example, use the following command on a managed server:

```
delete subscription engineering discardobjects=no
```

Setting Up Administrators for the Servers

Include in your profiles any administrators that you want to give access to all servers in the network. These administrators must then maintain their passwords on the configuration manager. To ensure passwords stay valid for as long as expected on all servers, set the password expiration period to the same time on all servers. One way to do this is to route a SET PASSEXP command from one server to all of the others.

Ensure that you have at least one administrator that is defined locally on each managed server with system authority. This avoids an error on configuration refresh when all administrators for a server would be removed as a result of a change to a profile on the configuration manager.

Handling Problems with Synchronization of Profiles

In rare situations, when a managed server contacts a configuration manager to refresh configuration information, the configuration manager may determine that the profile information on the two servers is not synchronized. It may appear that the configuration information is more recent on the managed server than on the configuration manager. This could occur in the following situations:

- The database on the configuration manager has been restored to an earlier time and now has configuration information from profiles that appear to be older than what the managed server has obtained.
- On the configuration manager, an administrator deleted a profile, forcing the deletion even though one or more managed servers still subscribed to the profile. The administrator redefined the profile (using the same name) before the managed server refreshed its configuration information.

If the configuration manager still has a record of the managed server's subscription to the profile, the configuration manager does not send its profile information at the next request for refreshed configuration information. The configuration manager informs the managed server that the profiles are not synchronized. The managed server then issues a message indicating this condition so that an administrator can take appropriate action. The administrator can perform the following steps:

1. If the configuration manager's database has been restored to an earlier point in time, the administrator may want to query the profile and associated objects on the managed server and then manually update the configuration manager with that information.
2. Use the DELETE SUBSCRIPTION command on the managed server to delete subscriptions to the profile that is not synchronized. If desired, you can also delete definitions of the associated objects, then define the subscription again.

It is possible that the configuration manager may not have a record of the managed server's subscription. In this case, no action is necessary. When the managed server requests a refresh of configuration information, the configuration manager sends current profile information and the managed server updates its database with that information.

Switching a Managed Server to a Different Configuration Manager

To switch a managed server from one configuration manager to another, perform the following steps:

1. Query profiles on the server that will be the new configuration manager to compare with current profiles to which the managed server subscribes.
2. On the managed server, delete all subscriptions to profiles on the current configuration manager. Remember to delete the subscription to the profile named DEFAULT_PROFILE. Consider whether to discard the managed objects in the database when you delete the subscriptions.

   Verify that all subscriptions have been deleted by querying subscriptions.
3. Change server communications as needed. Define the server that will be the new configuration manager. You can delete the server that was formerly the configuration manager.
4. On the managed server, define subscriptions to profiles on the new configuration manager.

Deleting Subscribers from a Configuration Manager

Under normal circumstances, you do not need to delete subscribers from a configuration manager. You only need to delete a subscription to a profile on the managed server (by using the DELETE SUBSCRIPTION command). When you issue the DELETE SUBSCRIPTION command, the managed server automatically notifies the configuration manager of the deletion by refreshing its configuration information. As part of the refresh process, the configuration manager is informed of the profiles to which the managed server subscribes and to which it does not subscribe. If the configuration manager cannot be contacted immediately for a refresh, the configuration manager will find out that the subscription was deleted the next time the managed server refreshes configuration information.

Deleting subscribers from a configuration manager is only necessary as a way to clean up in certain unusual situations. For example, you may need to delete subscribers if a managed server goes away completely or deletes its last subscription without being able to notify the configuration manager. You then use the DELETE SUBSCRIBER command to delete all subscriptions for that subscriber (the managed server) from the configuration manager's database.

Renaming a Managed Server

To rename a managed server, perform the following steps:

1. By using command routing or by logging on to the managed server, change the name of the managed server. Use the enterprise console or use the SET SERVERNAME command. See "Setting the Server Name" on page 468 for more information before using 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.

Performing Tasks on Multiple Servers

To make performing tasks with multiple servers easier, Tivoli Storage Manager provides the following functions:

- The Administration Center interface
- Command routing
- Server group definitions that can be used to simplify command routing

Working with Multiple Servers Using the Administration Center

The Administration Center is a Web-based interface that can be used to centrally configure and manage multiple IBM Tivoli Storage Manager Version 5.3 servers. See "Managing Servers with the New Administration Center" on page 36 for more information. You can log in to the Administration Center and access all of the Tivoli Storage Manager servers and Web clients for which you have administrative authority.

Routing Commands

If you have set up your servers as described in "Setting Up Communications for Command Routing" on page 571, you can route Tivoli Storage Manager administrative commands to one or more servers. 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. 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 (see "Setting Up Server Groups" on page 599), 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*.

Routing Commands to One or More Servers

The following sections describe how you can route commands to one or more servers, and to server groups. To successfully route commands to other servers, you must have the proper administrative authority on all servers that receive the command for processing.

The return codes for command routing can be one of three severities: 0, ERROR, or WARNING. See *Administrator's Reference* for a list of valid return codes and severity levels.

Routing Commands to Single Servers: To route a command to a single server, enter the defined server's name, a colon, and then the command to be processed. For example, to route a QUERY STGPOOL command to the server that is named ADMIN1, enter:

```
admin1: query stgpool
```

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

```
(admin1) query stgpool
```

Note: When writing scripts, you must use the parentheses for server routing information.

To route a command to more than one server, separate the server names with a comma. For example, to route a QUERY OCCUPANCY command to three servers named ADMIN1, GEO2, and TRADE5 enter:

```
admin1,geo2,trade5: query occupancy
```

or

```
(admin1,geo2,trade5) query occupancy
```

The command QUERY OCCUPANCY is routed to servers ADMIN1, GEO2, and TRADE5. If a server has not been defined with the DEFINE SERVER command, that server is skipped and the command routing proceeds to the next server in the route list.

The routed command output of each server is displayed in its entirety at the server that initiated command routing. In the previous example, output for ADMIN1 would be displayed, followed by the output of GEO2, and then the output of TRADE5.

Processing of a command on one server does not depend upon completion of the command processing on any other servers in the route list. For example, if GEO2 server does not successfully complete the command, the TRADE5 server continues processing the command independently.

Routing Commands to Server Groups: A server group is a named group of servers. Once you set up the groups, you can route commands to the groups. See "Setting Up Server Groups" on page 599 for how to set up a server group.

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.

Routing Commands to Single Servers and Server Groups: You can route commands to multiple single servers and to server groups at the same time. For example, to route the QUERY DB command to servers HQSRV, REGSRV, and groups WEST_COMPLEX and NORTH_COMPLEX, enter:

```
hqsrv,regsrv,west_complex,north_complex: query db
```

or

```
(hqsrv,regsrv,west_complex,north_complex) query db
```

The QUERY DB command is sent for processing to servers HQSRV, REGSRV, to BLD12 and BLD13 (both members of WEST_COMPLEX), and to NE12 and NW12 (both members of NORTH_COMPLEX).

Duplicate references to servers are removed in processing. For example, if you route a command to server BLD12 and to server group WEST_COMPLEX (which includes BLD12), the command is sent only once to server BLD12.

Setting Up Server Groups

You can make command routing more efficient by creating one or more server groups and adding servers to them. You can then route commands to server groups in addition to or in place of routing commands to single servers. This section describes how to set up server groups. To use server groups, you must do the following tasks:

1. Define the server groups.
2. Add the servers as members of the appropriate group.

After you have the server groups set up, you can manage the groups and group members.

Defining a Server Group and Members of a Server Group

Task	Required Privilege Class
Define a server group	System
Define a server group member	System

You can define groups of servers to which you can then route commands. The commands are routed to all servers in the group. To route commands to a server group you must do the following:

1. Define the server with the DEFINE SERVER command if it is not already defined (see “Setting Up Communications for Command Routing” on page 571).
2. Define a new server group with the DEFINE SERVERGROUP command. Server group names must be unique because both groups and server names are allowed for the routing information.
3. Define servers as members of a server group with the DEFINE GRPMEMBER command.

The following example shows how to create a server group named WEST_COMPLEX, and define servers BLD12 and BLD13 as members of the WEST_COMPLEX group:

```
define servergroup west_complex
define grpmember west_complex bld12,bld13
```

Managing Server Groups

You can query, copy, rename, update, and delete server groups as necessary.

Task	Required Privilege Class
Query a server group	System
Copy a server group	System
Rename a server group	System
Update a server group description	System
Delete a server group	System

Querying a Server Group: To query server group WEST_COMPLEX, enter:

```
query servergroup west_complex
```

The following is sample output from a QUERY SERVERGROUP command:

```
Server Group      Members        Description          Managing profile
--------------------------------------------------------------------------
WEST_COMPLEX      BLD12, BLD13
```

Copying a Server Group: To copy the entire server group contents of WEST_COMPLEX to a different server group named NEWWEST, enter:

```
copy servergroup west_complex newwest
```

This command creates the new group. If the new group already exists, the command fails.

Renaming a Server Group: To rename an existing server group NORTH_COMPLEX to NORTH, enter:

```
rename servergroup north_complex north
```

Updating a Server Group Description: To update the NORTH server group to modify its description, enter:

```
update servergroup north description="Northern marketing region"
```

Deleting a Server Group: To delete WEST_COMPLEX server group from the Tivoli Storage Manager server, enter:

```
delete servergroup west_complex
```

This command removes all members from the server group. The server definition for each group member is not affected. If the deleted server group is a member of other server groups, the deleted group is removed from the other groups.

Managing Group Members

You can move and delete group members from a previously defined group.

Task	Required Privilege Class
Move a group member to another group	System
Delete a group member	

Moving a Group Member to Another Group: To move group member TRADE5 from the NEWWEST group to the NORTH_COMPLEX group, enter:

```
move grpmember trade5 newwest north_complex
```

Deleting a Group Member from a Group: To delete group member BLD12 from the NEWWEST server group, enter:

```
delete grpmember newwest bld12
```

When you delete a server, the deleted server is removed from any server groups of which it was a member.

Querying Server Availability

You can test a connection from your local server to a specified server with the PING SERVER command. To ping the server GEO2, enter:

```
ping server geo2
```

The PING SERVER command uses the user ID and password of the administrative ID that issued the command. If the administrator is not defined on the server being pinged, the ping fails even if the server may be running.

Using Virtual Volumes to Store Data on Another Server

Tivoli Storage Manager lets a server (a *source server*) store the results of database backups, export operations, and storage pool operations on another server (a *target server*). The data is stored as *virtual volumes*, which appear to be sequential media volumes on the source server but which are actually stored as archive files on a target server. Virtual volumes can be any of the following:

- Database backups
- Storage pool backups
- Data that is backed up, archived, or space managed from client nodes
- Client data migrated from storage pools on the source server
- Any data that can be moved by EXPORT and IMPORT commands

The source server is a client of the target server, and the data for the source server is managed only by the source server. In other words, the source server controls the expiration and deletion of the files that comprise the virtual volumes on the target server. The use of virtual volumes is not supported when the source server and the target server reside on the same Tivoli Storage Manager server.

At the target server, the virtual volumes from the source server are seen as archive data. The source server is registered as a client node (of TYPE=SERVER) at the target server and is assigned to a policy domain. The archive copy group of the default management class of that domain specifies the storage pool for the data from the source server.

Note: If the default management class does not include an archive copy group, data cannot be stored on the target server.

Using virtual volumes can benefit you in the following ways:

- The source server can use the target server as an electronic vault for rapid recovery from a disaster.
- Smaller Tivoli Storage Manager source servers can use the storage pools and tape devices of larger Tivoli Storage Manager servers.

- For incremental database backups, it can decrease wasted space on volumes and under use of high-end tape drives.

Be aware of the following when you use virtual volumes:

- If you use virtual volumes for database backups, you might have the following situation: SERVER_A backs up its database to SERVER_B, and SERVER_B backs up its database to SERVER_A. If this is the only way databases are backed up, if both servers are at the same location, and if a disaster strikes that location, you may have no backups with which to restore your databases.
- Moving large amounts of data between the servers may slow down your communications significantly, depending on the network bandwidth and availability.
- You can specify in the device class definition (DEVTYPE=SERVER) how often and for how long a time the source server will try to contact the target server. Keep in mind that frequent attempts to contact the target server over an extended period can affect your communications.
- Under certain circumstances, inconsistencies may 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 (see "Reconciling Virtual Volumes and Archive Files" on page 606 for details).
- If you want to enable data validation between a source and target server, enable the settings using both the DEFINE SERVER and REGISTER NODE commands. For more information see "Validating a Node's Data" on page 411 and *Administrator's Reference*.
- Storage space limitations on the target server affect the amount of data that you can store on that server.
- To minimize mount wait times, the total mount limit for all server definitions that specify the target server should 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 could exceed the target server's tape drives.

Note: When you issue a DEFINE SERVER command, the source server sends a verification code to the target server. When the source server begins a session with the target server, it also sends the verification code. If the code matches what was previously stored on the target, the session is opened in read/write mode. If the verification code is lost at the source server (for example, after a database restore), the code can be reset by issuing an UPDATE SERVER command with the FORCESYNC=YES parameter.

Setting Up Source and Target Servers for Virtual Volumes

In the source/target relationship, the source server is defined as a client node of the target server. To set up this relationship, a number of steps must be performed at the two servers. In the following example (illustrated in Figure 93 on page 604), the source server is named DELHI and the target server is named TOKYO.

- **At DELHI**:
  1. Define the target server:
     - TOKYO has a TCP/IP address of 9.115.3.221:1845
     - Assigns to TOKYO the password CALCITE.

– Assigns DELHI as the node name by which the source server DELHI will be known at 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 TOKYO:**

  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
     ```

     After issuing these commands, ensure that you assign the source server to the new policy domain (UPDATE NODE) and activate the policy. See "Changing Policy" on page 360 for details.

Figure 93. Communication configuration for virtual volumes

Performing Operations at the Source Server

You can perform certain operations at the source server that cause data to be stored in a storage pool at the target server. These operations are:

- Database backups
- Storage pool backups
- Client data backup, archive, or migration
- Data migration from one storage pool to another
- Export of server information

The following sections describe how to perform these operations. In the examples, the following is assumed:

- The definitions shown in the previous section have been done.
- An operational TCP/IP connection exists between both servers.
- Both servers are running.

Back Up the Database

You can back up the database of a source server to a target server. For example, to perform an incremental backup of the source server and send the volumes to the target server, issue the following command:

```
backup db type=incremental devclass=targetclass
```

You can also do an automatic database backup to a target server. For example, if you have issued the following command, a database backup occurs automatically when more than 60 percent of recovery log space is used:

```
define dbbackuptrigger devclass=targetclass logfullpct=60
```

Back Up a Storage Pool

You can back up a storage pool of a source server to a target server. For example, a primary storage pool named TAPEPOOL is on the source server. You can define a

copy storage pool named TARGETCOPYPOOL, also on the source server. TARGETCOPYPOOL must have an associated device class whose device type is SERVER. When you back up TAPEPOOL to TARGETCOPYPOOL, the backup is sent to the target server. To do so, issue the following commands:

```
define stgpool targetcopypool targetclass pooltype=copy
  maxscratch=20
backup stgpool tapepool targetcopypool
```

Store Client Data on a Target Server

You can configure your Tivoli Storage Manager system so that when client nodes registered to the source server back up, archive, or migrate their data, that data is sent to the target server. When clients restore, retrieve, or recall their data, the source server gets the data from the target server.

To configure your system, ensure that the management policy for those nodes specifies a storage pool that has a device class whose device type is SERVER. For example, the following command defines the storage pool named TARGETPOOL.

```
define stgpool targetpool targetclass maxscratch=20
  reclaim=100
```

Note: Reclamation of a storage pool automatically begins when the percentage of reclaimable space, which is specified by the RECLAIM parameter, is reached. Reclamation of a target storage pool can involve the movement of a great deal of data from the target server to the source server and back to the target. If this operation occurs automatically during peak operating periods, it could slow network performance significantly. If you set the value to 100, reclamation will not occur automatically. For details about storage pool reclamation and how to begin it manually, see "Reclaiming Space in Sequential Access Storage Pools" on page 267.

Migrate Data from a Source Server Storage Pool to a Target Server Storage Pool

You can set up your storage pool hierarchy so that client data is migrated from a storage pool on the source server to the target server. For example, storage pool TAPEPOOL is on the source server. The TAPEPOOL definition specifies NEXTSTGPOOL=TARGETPOOL. TARGETPOOL has been defined on the source server as a storage pool of device type SERVER. When data is migrated from TAPEPOOL, it is sent to the target server.

```
define stgpool tapepool tapeclass nextstgpool=targetpool
  maxscratch=20
```

Export Server Information to a Target Server

You can use any of the Tivoli Storage Manager EXPORT commands to export data from one Tivoli Storage Manager source server to sequential media on a target Tivoli Storage Manager server. You must specify a device class with a device type specified as SERVER. For example, to copy server information directly to a target server, issue the following command:

```
export server devclass=targetclass
```

Import Server Information from a Target Server: If data has been exported from a source server to a target server, you can import that data from the target server to a third server. The server that will import the data uses the node ID and password of the source server to open a session with the target server. That session is in read-only mode because the third server does not have the proper verification code.

For example, to import server information from a target server, issue the following command:

```
import server devclass=targetclass
```

Reconciling Virtual Volumes and Archive Files

If you have restored the database on the source or target server, you should perform reconciliation between the virtual volumes on the source server and the archive files on the target server. You should also perform reconciliation if you have any other reason to suspect inconsistencies. For example, frequent communication errors between target and source servers could introduce a problem.

To perform reconciliation, issue the RECONCILE VOLUMES command specifying a device class of 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 36.

Table 36. 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

Table 36. FIX Parameter Reconciliation (continued)

FIX=	At the Source Server	At the Target Server	Action
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 22. 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
Perform 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:
"Data That Can Be Exported and Imported"
Tasks for Exporting Directly to Another Server:
"Exporting Data Directly to Another Server" on page 612
"Preparing to Export to Another Server for Immediate Import" on page 614
"Monitoring the Server-to-Server Export Process" on page 615
Tasks for Exporting to Sequential Media:
"Exporting and Importing Data Using Sequential Media Volumes" on page 617
"Exporting Tasks" on page 619
"Importing Data from Sequential Media Volumes" on page 621

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

Data That Can Be Exported and Imported

Administrators can export or import the following types of data:

- Server control information, which includes:
  - Administrator definitions
  - Client node definitions
  - Policy and scheduling definitions

© Copyright IBM Corp. 1993, 2005

- File data from server storage, which includes file space definitions and authorization rules. You can request that file data be exported in any of the following groupings of files:
  - Active and inactive versions of backed up files, archive copies of files, and space-managed files
  - Active versions of backed up files, archive copies of files, and space-managed files
  - Active and inactive versions of backed up files
  - Active versions of backed up files
  - Archive copies of files
  - Space-managed files

Exporting Restrictions

Restrictions for exporting data are as follows:

- 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.
- Data exported from a server at version 4.1.2 or later with Unicode support cannot be imported to a server at an earlier version.
- Data can be exported from a server with retention protection enabled and will not be retention protected when imported on another server.
- Data that is in deletion hold status can be exported, but the hold status will not be preserved on the export media. It will not be in deletion hold when it is imported to another server.
- You cannot export nodes of type NAS. Export processing will exclude these nodes.

Deciding What Information to Export

Your decision on what information to export depends on why you are exporting that information:

- To balance the work load 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.

Note: Because results could be unpredictable, ensure that expiration, migration, backup, or archive processes are not running when the EXPORT NODE command is issued.

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. See "Preventing Administrative Clients from Accessing the Server" and "Preventing 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. See:

- "Options to Consider Before Exporting" on page 612
- "Preventing Client Nodes from Accessing the Server"
- "Preventing Administrative Clients from Accessing the Server"

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.

For more information on canceling sessions, see "Canceling an IBM Tivoli Storage Manager Session" on page 344. For more information on locking or unlocking administrators from the server, see "Locking and Unlocking Administrators from the Server" on page 352.

Preventing Client Nodes from Accessing the Server

If client node information is exported while that client is backing up, archiving, or migrating files, the latest file copies for the client may not be exported to tape. To prevent users from accessing the server during export operations, cancel existing client sessions as described in "Canceling an IBM Tivoli Storage Manager Session" on page 344. Then you can do one of the following:

- Disable server access to prevent client nodes from accessing the server, as described in "Disabling or Enabling Access to the Server" on page 346.

  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.
- Lock out particular client nodes from server access, as described in "Locking and Unlocking Client Nodes" on page 320.

  This option is useful when you export a subset of client node information from the source server and want to prevent particular client nodes from accessing the server until the export operation is complete.

After the export operation is complete, allow client nodes to access the server again by:

- Enabling the server, as described in "Disabling or Enabling Access to the Server" on page 346
- Unlocking client nodes, as described in "Locking and Unlocking Client Nodes" on page 320

Exporting Data Directly to Another Server

You can export all server control information or a subset of server control information by specifying 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

The following sections describe options to consider before you export data for immediate import to another server.

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, see "Understanding How Duplicate File Spaces Are Handled" on page 628.

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 describes 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 does the following:

- 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. See "Querying the Activity Log for Export or Import Information" on page 633 for more information.

Incremental Export

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. For Tivoli Storage Manager servers at version 5.1 or higher, 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 being moved 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.

Replace Definitions

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. For more information, see "Determining Whether to Replace Existing Definitions" on page 623. This option is available when you issue any of the EXPORT commands.

Preparing to Export to Another Server for Immediate Import

When you export data to another server on the network, the export results in an immediate import on the target server. You can export data to a Tivoli Storage Manager server of the same or different operating system as the originating server. A server-to-server export operation does the following:

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, do the following:

- 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*.
- Consider setting up enterprise configuration for the target server so you can distribute consistent backup and archive policies to the target server. For details, see Chapter 21, "Working with a Network of IBM Tivoli Storage Manager Servers," on page 563.
- Use the DEFINE SERVER command to define the name of the target server or the originating server. For more information, see "Setting Up Communications Among Servers" on page 568.
- 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.

Previewing Results of an Export Operation for Immediate Import

When you export data to another server, you can use the PREVIEWIMPORT option to determine how much data will be transferred without actually moving any data. When PREVIEWIMPORT=NO, the export operation is performed, and the data is immediately imported to the target server. This option is available when you issue any EXPORT command.

Issue each EXPORT command with PREVIEWIMPORT=YES to determine which objects and how much data will be copied to the target server. Use this information to determine how much storage pool space is required on the target server. The server sends the messages to the server console and to the activity log for each operation:

To determine how much space is required to export all server data, enter:

```
export server filedata=all previewimport=yes
```

After you issue this command, the server starts a background process and issues a message similar to the following:

```
EXPORT SERVER started as Process 4
```

You can view the preview results on the server console or by querying the activity log.

You can request information about the background process, as described in "Requesting Information about an Export or Import Process" on page 632. If necessary, you can cancel an export or import process, as described in "Canceling Server Processes" on page 466.

Directing Import Messages to an Output File

The information generated by the validation process can help you define a storage hierarchy that supports the storage destinations currently defined in the import data.

You can direct import messages to an output file to capture any error messages that are detected during the import process. Do this by starting an administrative client session in console mode before you invoke the import command.

For example, to direct messages to an output file named IMPSERV.OUT, enter:

```
> dsmadmc -consolemode -outfile=impserv.out
```

Monitoring the Server-to-Server Export Process

You can view the export and import process in two ways:

- 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 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.
- Watch for mount messages, because the server might request mounts of volumes that are not in the library. Check-in of volumes may be required.
- The connection between the servers might time-out. You may need to adjust the COMMTIMEOUT and IDLETIMEOUT server options on one or both servers.

Exporting Administrator Information to Another Server

When you issue the EXPORT ADMIN command, the server exports administrator definitions. Each administrator definition includes:

- Administrator name, password, and contact information
- Any administrative privilege classes the administrator has been granted
- Whether the administrator ID is locked from server access

You can specify a list of administrator names, or you can export all administrator names.

The following example exports all the administrator definitions to the target server defined as OTHERSERVER. It allows you to preview the export without actually exporting the data for immediate import.

```
export admin * toserver=otherserver previewimport=yes
```

You can preview the result on the server console or by querying the activity log.

Exporting Client Node Information to Another Server

When you issue the EXPORT NODE command, the server exports client node definitions. Each client node definition includes:

- User ID, password, and contact information

- Name of the policy domain to which the client is assigned
- File compression status
- Whether the user has the authority to delete backed up or archived files from server storage
- Whether the client node ID is locked from server access

You can also specify whether to export file data. File data includes file space definitions and authorization rules. You can request that file data be exported in any of the following groupings of files:

- Active and inactive versions of backed up files, archive copies of files, and space-managed files
- Active versions of backed up files, archive copies of files, and space-managed files
- Active and inactive versions of backed up files
- Active versions of backed up files
- Archive copies of files
- Space-managed files

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
```

Note: 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 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, to export policy information directly to SERVERB, issue the following 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.

For example, you want 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. You want to replace definitions on the target server, and you want the data that is exported to begin with any data inserted in the originating server beginning 10/25/2002. To issue this command, enter:

```
export server toserver=serv23 fromdate=10/25/2002 filedata=all
 mergefilespaces=yes dates=relative
```

Exporting and Importing Data Using Sequential Media Volumes

This section describes how to export and import data using sequential media volumes.

Note: When exporting data from a Tivoli Storage Manager server other than an OS/400® PASE server (for example, from a Tivoli Storage Manager Windows server) to a Tivoli Storage Manager OS/400 PASE server, use a server-to-server export rather than an export to sequential media. The CHECKIN LIBVOLUME command fails on a Tivoli Storage Manager OS/400 PASE server when the server attempts to check in a sequential-media volume containing export data from servers other than OS/400 PASE servers. If you must use an LTO, Ultrium or 3592 device to create an export tape, follow these steps:

1. Define a Tivoli Storage Manager library of type MANUAL.
2. Define an LTO, Ultrium or 3592 tape drive in the MANUAL library.
3. Label a tape with six characters or less.
4. Perform the export.

Preparing to Export or Import Data (Sequential Media)

Before you export or import data, do the following:

- Use the EXPORT or IMPORT command with the PREVIEW parameter to verify what data will be moved
- 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.

Issue each EXPORT or IMPORT command with PREVIEW=YES to determine which objects and how much data will be moved. The server sends the following types of messages to the server console and to the activity log for each operation:

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.

To determine how much space is required to export all server data, enter:

```
export server filedata=all preview=yes
```

After you issue this command, the server starts a background process and issues a message similar to the following:

```
EXPORT SERVER started as Process 4
```

You can view the preview results on the server console or by querying the activity log.

You can request information about the background process, as described in "Requesting Information about an Export or Import Process" on page 632. If necessary, you can cancel an export or import process, as described in "Canceling Server Processes" on page 466.

Planning for Sequential Media Used to Export Data

To export data, you must specify a device class that supports sequential media and identify the volumes that will be used to store the exported data. Use this section to help you select the device classes and prepare sequential media volumes.

Selecting a Device Class: You can query the source and target servers to select a device class on each server that supports the same device type. If you cannot find a device class on each server that supports a matching device type, define a new device class for a device type that is available to both servers. See Chapter 9, "Defining Device Classes," on page 197.

Notes:

1. If the mount limit for the device class selected is reached when you request an export (that is, if all the drives are busy), the server automatically cancels lower priority operations, such as reclamation, to make a mount point available for the export.
2. You can export data to a storage pool on another server by specifying a device class whose device type is SERVER. For details, see "Using Virtual Volumes to Store Data on Another Server" on page 601.

Estimating the Number of Removable Media Volumes to Label: To estimate the number of tapes needed to store export data, divide the number of bytes to be moved by the estimated capacity of a volume.

For example, cartridge system tape volumes used with 3490 tape devices have an estimated capacity of 360MB. If the preview shows that you need to transfer 720MB of data, label at least two tape volumes before you export the data.

Using Scratch Media: The server allows you to use scratch media to ensure that you have sufficient space to store all export data. If you use scratch media, record the label names and the order in which they were mounted. Or, use the USEDVOLUMELIST parameter on the export command to create a file containing the list of volumes used.

Labeling Removable Media Volumes: During an import process, you must specify the order in which volumes will be mounted. This order must match the order in which tapes were mounted during the export process. To ensure that tapes are mounted in the correct order, label tapes with information that identifies the order in which they are mounted during the import process. For example, label tapes as DSM001, DSM002, DSM003, and so on.

When you export data, record the date and time for each labeled volume. Store this information in a safe location, because you will need the information when

you import the data. Or, if you used the USEDVOLUMELIST parameter on the export command, save the resulting file. This file can be used on the import command volumes parameter.

Exporting Tasks

You can export all server control information or a subset of server control information by specifying 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. See "Labeling Removable Media Volumes" on page 618 for information on labeling tape volumes.

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.

Exporting Administrator Information

When you issue the EXPORT ADMIN command, the server exports administrator definitions. Each administrator definition includes:

- Administrator name, password, and contact information
- Any administrative privilege classes the administrator has been granted
- Whether the administrator ID is locked from server access

You can specify a list of administrator names, or you can export all administrator names.

In the following example, definitions for the DAVEHIL and PENNER administrator IDs will be exported to the DSM001 tape volume, which the TAPECLASS device class supports. Do not allow any scratch media to be used during this export process. To issue this command, enter:

```
export admin davehil,penner devclass=tapeclass
volumenames=dsm001 scratch=no
```

Exporting Client Node Information

When you issue the EXPORT NODE command, the server exports client node definitions. Each client node definition includes:

- User ID, password, and contact information
- Name of the policy domain to which the client is assigned
- File compression status

- Whether the user has the authority to delete backed up or archived files from server storage
- Whether the client node ID is locked from server access

You can also specify whether to export file data. File data includes file space definitions and authorization rules. You can request that file data be exported in any of the following groupings of files:

- Active and inactive versions of backed up files, archive copies of files, and space-managed files
- Active versions of backed up files, archive copies of files, and space-managed files
- Active and inactive versions of backed up files
- Active versions of backed up files
- Archive copies of files
- Space-managed files

When client file data is exported, the server copies files to export volumes in the order of their physical location in server storage. 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 do the following:

- Export definitions for client nodes and file spaces in the ENGPOLDOM policy domain
- Export any active backup versions of files belonging to these client nodes
- Export this information to scratch volumes in the TAPECLASS device class

To issue this command, enter:

```
export node filespace=* domains=engpoldom
filedata=backupactive devclass=tapeclass
```

In this example, the server exports:

- Definitions of client nodes assigned to ENGPOLDOM
- File space definitions and backup authorizations for each client node in ENGPOLDOM
- Active versions of backed up files belonging to the client nodes assigned to ENGPOLDOM

Exporting Policy Information

When you issue the EXPORT POLICY command, the server exports the following information belonging to each specified policy domain:

- Policy domain definitions
- Policy set definitions, including the active policy set
- Management class definitions, including the default management class
- Backup copy group and archive copy group definitions
- Schedule definitions
- Associations between client nodes and schedules

For example, suppose that you want to export policy and scheduling definitions from the policy domain named ENGPOLDOM. You want to use tape volumes DSM001 and DSM002, which belong to the TAPECLASS device class, but allow the server to use scratch tape volumes if necessary. To issue this command, enter:

```
export policy engpoldom
devclass=tapeclass volumenames=dsm001,dsm002
```

Exporting Server Data

When you issue the EXPORT SERVER command, the server exports all server control information. You can also export file data information with the EXPORT SERVER command.

For example, you want to export server data to four defined tape cartridges, which the TAPECLASS device class supports. You want the server to use scratch volumes if the four volumes are not enough, and so you use the default of SCRATCH=YES. To issue this command, enter:

```
export server devclass=tapeclass
volumenames=dsm001,dsm002,dsm003,dsm004 filedata=all
```

During the export process, the server exports definition information before it exports file data information. This ensures that definition information is stored on the first tape volumes. This process allows you to mount a minimum number of tapes during the import process, if your goal is to copy only control information to the target server.

In the example above, the server exports:

- Administrator definitions
- Client node definitions
- Policy domain, policy set, management class, and copy group definitions
- Schedule definitions and client node associations
- File space definitions
- File space authorization rules
- Backed up, archived, and space-managed files

Importing Data from Sequential Media Volumes

Before you import data to a new target server, you must:

1. Install Tivoli Storage Manager for the target server. This step includes defining disk space for the database and recovery log.

   For information on installing Tivoli Storage Manager, see *Installation Guide*.
2. Define server storage for the target server.

   Because each server operating system handles devices differently, server storage definitions are not exported. Therefore, you must define initial server storage for the target server. The target server must at least be able to use a drive that is compatible with the export media. This task can include defining libraries, drives, device classes, storage pools, and volumes. See the *Administrator's Guide* that applies to the target server.

After Tivoli Storage Manager is installed and set up on the target server, a system administrator can import all server control information or a subset of server control information by specifying one or more of the following import commands:

- IMPORT ADMIN
- IMPORT NODE

- IMPORT POLICY
- IMPORT SERVER

This section guides you through the entire process of importing all server control information and file data from tape volumes to a new target server. This process includes:

- Previewing information before you import data
- Importing definitions
- Tailoring server storage definitions on the target server
- Importing file data

After you understand how to import server control information and file data information, you can import any subset of data to the target server.

Options to Consider Before Importing

The following sections describe options to consider before you import data from sequential media.

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, see "Understanding How Duplicate File Spaces Are Handled" on page 628.

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 describes 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 does the following:

- 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. See "Querying the Activity Log for Export or Import Information" on page 633 for more information.

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.

Step 1: Previewing Information before You Import Data

Before you import any data to the target server, preview each IMPORT command to determine what data you want to import to the target server. You can import all or a subset of export data from tapes.

When you set PREVIEW=YES, tape operators must mount export tape volumes so that the target server can calculate the statistics for the preview.

For example, to preview information for the IMPORT SERVER command, enter:

```
import server devclass=tapeclass preview=yes
volumenames=dsm001,dsm002,dsm003,dsm004
```

Figure 94 on page 625 shows an example of the messages sent to the server console and the activity log.

```
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 94. 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 94 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 use 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. See "Step 3: Tailoring Server Storage Definitions on the Target Server" on page 627 for information on identifying storage pools on the target server.

For information on specifying the PREVIEW parameter, see "Using Preview before Exporting or Importing Data" on page 617. For information on reviewing the results of a preview operation, see "Monitoring Export and Import Processes" on page 631.

Step 2: Importing Definitions

Next, you want to import server control information, which includes:

- Administrator definitions
- Client node definitions
- Policy domain, policy set, management class, and copy group definitions
- Schedule definitions and client node associations

However, do not import file data at this time, because some storage pools named in the copy group definitions may not exist yet on the target server.

Before you import server control information, do the following:

- Read and understand the following information:
  - "Determining Whether to Replace Existing Definitions" on page 623
  - "Understanding How the Server Imports Active Policy Sets"
- Start an administrative client session in console mode to capture import messages to an output file. See "Directing Import Messages to an Output File" on page 627.

Then import the server control information from specified tape volumes. See "Importing Server Control Information" on page 627.

Understanding How the Server Imports Active Policy Sets: When the server imports policy definitions, the following objects are imported to the target server:

- 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.

After $$ACTIVE$$ is imported to the target server, the server activates this policy set. During the activation process, the server validates the policy set by examining the management class and copy group definitions. If any of the following conditions occur, the server issues warning messages during validation:

- The storage destinations specified in the backup and archive copy groups do not refer to defined storage pools.
- The default management class does not contain a backup or archive copy group.

- The current ACTIVE policy set contains management class names that are not defined in the policy set to be activated.
- The current ACTIVE policy set contains copy group names that are not defined in the policy set to be activated.

After each $$ACTIVE$$ policy set has been activated, the server deletes that $$ACTIVE$$ policy set from the target server. To view information about active policy on the target server, you can use the following commands:

- QUERY COPYGROUP
- QUERY DOMAIN
- QUERY MGMTCLASS
- QUERY POLICYSET

Results from issuing the QUERY DOMAIN command show the activated policy set as $$ACTIVE$$. The $$ACTIVE$$ name shows you that the policy set which is currently activated for this domain is the policy set that was active at the time the export was performed.

Directing Import Messages to an Output File: The information generated by the validation process can help you define a storage hierarchy that supports the storage destinations currently defined in the import data.

You can direct import messages to an output file to capture any error messages that are detected during the import process. Do this by starting an administrative client session in console mode before you invoke the import command.

For example, to direct messages to an output file named IMPSERV.OUT, enter:

```
> dsmadmc -consolemode -outfile=impserv.out
```

Importing Server Control Information: Now you are ready to import the server control information. Based on the information generated during the preview operation, you know that all definition information has been stored on the first tape volume named DSM001. Specify that this tape volume can be read by a device belonging to the TAPECLASS device class.

From an administrative client session or from the server console, enter:

```
import server filedata=none devclass=tapeclass volumenames=dsm001
```

Step 3: Tailoring Server Storage Definitions on the Target Server

After you import definition information, use the reports generated by the import process to help you tailor storage for the target server.

To tailor server storage definitions on the target server, complete the following steps:

1. Identify any storage destinations specified in copy groups and management classes that do not match defined storage pools:
   - If the policy definitions you imported included an ACTIVE policy set, that policy set is validated and activated on the target server. Error messages generated during validation include whether any management classes or copy groups refer to storage pools that do not exist on the target server. You have a copy of these messages in a file if you directed console messages to an output file as described in “Directing Import Messages to an Output File.”

- Query management class and copy group definitions to compare the storage destinations specified with the names of existing storage pools on the target server.

  To request detailed reports for all management classes, backup copy groups, and archive copy groups in the ACTIVE policy set, enter these commands:

  ```
  query mgmtclass * active * format=detailed
  query copygroup * active * standard type=backup format=detailed
  query copygroup * active * standard type=archive format=detailed
  ```

2. If storage destinations for management classes and copy groups in the ACTIVE policy set refer to storage pools that are not defined, do one of the following:
   - Define storage pools that match the storage destination names for the management classes and copy groups, as described in "Defining or Updating Primary Storage Pools" on page 220.
   - Change the storage destinations for the management classes and copy groups. Do the following:
     a. Copy the ACTIVE policy set to another policy set
     b. Modify the storage destinations of management classes and copy groups in that policy set, as required
     c. Activate the new policy set

     For information on copying policy sets, see "Defining and Updating a Policy Set" on page 379.

Depending on the amount of client file data that you expect to import, you may want to examine the storage hierarchy to ensure that sufficient storage space is available. Storage pools specified as storage destinations by management classes and copy groups may fill up with data. For example, you may need to define additional storage pools to which data can migrate from the initial storage destinations.

Step 4: 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

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

Understanding How Duplicate File Spaces Are Handled: 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 server console and to the activity log.

For example, if the C_DRIVE and D_DRIVE file space names reside on the target server for node FRED and on the tape volume for FRED, then the server imports the C_DRIVE file space as C_DRIV1 file space and the D_DRIVE file space as D_DRIV1 file space, both assigned to node FRED.

Deciding Whether to Use a Relative Date When Importing File Data: When you import file data, you can keep the original creation date for backup versions and archive copies, or you can specify that the server use an adjusted date.

Because tape volumes containing exported data might not be used for some time, the original dates defined for backup versions and archive copies may be old enough that files are expired immediately when the data is imported to the target server.

To prevent backup versions and archive copies from being expired immediately, specify DATES=RELATIVE on the IMPORT NODE or IMPORT SERVER commands to adjust for the elapsed time since the files were exported to tape.

For example, assume that data exported to tape includes an archive copy archived five days prior to the export operation. If the tape volume resides on the shelf for six months before the data is imported to the target server, the server resets the archival date to five days prior to the import operation.

If you want to keep the original dates set for backup versions and archive copies, use DATES=ABSOLUTE, which is the default. If you use the absolute value, any files whose retention period has passed will be expired shortly after they are imported to the target server.

Issuing an Import Server or Import Node Command: You can import file data, either by issuing the IMPORT SERVER or IMPORT NODE command. When you issue either of these commands, you can specify which type of files should be imported for all client nodes specified and found on the export tapes. You can specify any of the following values to import file data:

All Specifies that all active and inactive versions of backed up files, archive copies of files, and space-managed files for specified client nodes are imported to the target server

None Specifies that no files are imported to the target server; only client node definitions are imported

Archive
Specifies that only archive copies of files are imported to the target server

Backup
Specifies that only backup copies of files, whether active or inactive, are imported to the target server

Backupactive
Specifies that only active versions of backed up files are imported to the target server

Allactive
Specifies that only active versions of backed up files, archive copies of files, and space-managed files are imported to the target server

Spacemanaged
Specifies that only files that have been migrated from a user's local file system (space-managed files) are imported

For example, suppose you want to import all backup versions of files, archive copies of files, and space-managed files to the target server. You do not want to replace any existing server control information during this import operation. Specify the four tape volumes that were identified during the preview operation. These tape volumes can be read by any device in the TAPECLASS device class. To issue this command, enter:

```
import server filedata=all replacedefs=no
devclass=tapeclass volumenames=dsm001,dsm002,dsm003,dsm004
```

You can limit the import to nodes that were assigned to specific policy domains on the source server. For example, suppose you exported from the source server the data for all nodes in all domains. To import to the target server the data only for nodes that were in the ENGDOM on the source server, enter this command:

```
import node filedata=all domains=engdom devclass=tapeclass
volumenames=dsm001,dsm002,dsm003,dsm004
```

If the ENGDOM policy domain exists on the target server, the imported nodes are assigned to that domain. If ENGDOM does not exist on the target server, the imported nodes are assigned to the STANDARD policy domain.

If you do not specify a domain on the IMPORT NODE command, the imported node is assigned to the STANDARD policy domain.

Importing Subsets of Information

You can use an import command to copy a subset of the information from export tapes to the target server. For example, if a tape was created with EXPORT SERVER, you can import only node information from the tape by using IMPORT NODE.

While the server allows you to issue any import command, data cannot be imported to the server if it has not been exported to tape. For example, if a tape is created with the EXPORT POLICY command, an IMPORT NODE command will not find any data on the tape because node information is not a subset of policy information.

Table 37 on page 631 shows the commands you can use to import a subset of exported information to a target server.

Table 37. 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 may 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 does the following:

- The default value is used for the new object's definition
- If the object already exists, the existing parameter is not changed

The server reports on the affected objects to the server console and the activity log during import and export operations. 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. For information on how Tivoli Storage Manager handles duplicate file spaces, see "Understanding How Duplicate File Spaces Are Handled" on page 628.

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*.

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 in two ways:

- 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 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.
- Watch for mount messages, because the server might request mounts of volumes that are not in the library. Check-in of volumes may be required.

Requesting Information about an Export or Import Process

After you issue an EXPORT or IMPORT command, the server starts a background process, assigns a process ID to the operation, and displays the process ID when the operation starts.

You can query an export or import process by specifying the process ID number. For example, to request information about the EXPORT SERVER operation, which started as process 4, enter:

```
query process 4
```

If you issue a preview version of an EXPORT or IMPORT command and then query the process, the server reports the types of objects to be copied, the number of objects to be copied, and the number of bytes to be copied.

When you export or import data and then query the process, the server displays the number and types of objects copied so far, and the total number of bytes that have been transferred, along with information on any media mount requests that may be outstanding for the process.

For information on querying background processes, see "Requesting Information about Server Processes" on page 515.

Viewing Information from the Server Console

When you issue an IMPORT or EXPORT command, either from the server console or from an administrative client, information is displayed on the server console. Figure 95 on page 633 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 95. Sample Export Server Output

Viewing Information from an Administrative Client

Use the console mode from an administrative client to monitor export or import operations or to capture processing messages to an output file. For example, to start an administrative session in console mode, enter:

```
> 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.

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 *Administrator's Reference*.

Querying the Activity Log for Export or Import Information

After an export or import process has completed, you can query the activity log for status information and possible error messages.

To minimize processing time when querying the activity log for export or import information, restrict the search by specifying EXPORT or IMPORT in the SEARCH parameter of the QUERY ACTLOG command.

For example, to determine how much data will be moved after issuing the preview version of the EXPORT SERVER command, query the activity log by entering:

```
query actlog search=export
```

Figure 96 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 96. Sample Activity Log Report on Exported Data

Exporting and Importing Data from Virtual Volumes

You can do all the EXPORT and IMPORT operations described in the previous sequential media sections to virtual volumes. For more information, see "Exporting and Importing Data Using Sequential Media Volumes" on page 617.

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. For details

about how to configure your server to export to or import from virtual volumes, see “Using Virtual Volumes to Store Data on Another Server” on page 601.

Part 5. Protecting the Server

© Copyright IBM Corp. 1993, 2005

Chapter 23. Protecting and Recovering Your Server

Failure or loss of the database, the recovery log, or storage pools can cause loss of client data. This chapter describes how you protect your server and, if necessary, recover your server.

Note: The term *tape* refers to any kind of sequential access, removable media unless otherwise indicated.

See the following sections:

Tasks:
Concepts:
"Levels of Protection" on page 640
"Storage Pool Protection: An Overview" on page 640
"Database and Recovery Log Protection: An Overview" on page 641
"Snapshot Database Protection" on page 644
"Choosing Where to Enable Data Validation" on page 673
Protecting Data:
"Mirroring the Database and Recovery Log" on page 644
"Backing Up Storage Pools" on page 648
"Backing Up the Database" on page 652
"Data Validation During Audit Volume Processing" on page 673
Recovering Data:
"Recovering Your Server Using Database and Storage Pool Backups" on page 661
"Restoring Your Server Using Mirrored Volumes" on page 669
"Restoring Storage Pool Volumes" on page 669
"Auditing a Storage Pool Volume" on page 671
"Correcting Damaged Files" on page 678
"Restoring a Library Manager Database" on page 684
"Restoring a Library Client Database" on page 685
Scenarios:
"Backup and Recovery Scenarios" on page 680

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

© Copyright IBM Corp. 1993, 2005

Levels of Protection

For the best protection of your data, you should use all of the following:

- Backups of your storage pools
- Mirrored copies of your database and recovery log, with the recovery log mode set to roll-forward
- Full and incremental backups of your database

As an adjunct to full and incremental database backups, you can also use snapshot database backups.

Attention: ADSM Version 1 provided database salvage commands in case of a catastrophic error. Although these commands are still available, you should use the current database backup and recovery functions for the best server protection. Do not use the database salvage commands without help from an IBM service representative.

Storage Pool Protection: An Overview

If one or more storage pool volumes is lost or damaged, the client data may be permanently lost. However, you can back up storage pools to sequential access copy storage pools and move the volumes offsite. If data is lost or damaged, you can restore individual volumes or entire storage pools from the copy storage pools. The server tries to access the file from a copy storage pool if the primary copy of the file cannot be obtained for one of the following reasons:

- The primary file copy has been previously marked damaged (for information about damaged files, see "Correcting Damaged Files" on page 678).
- The primary file is stored on a volume that UNAVAILABLE or DESTROYED.
- The primary file is stored on an offline volume.
- The primary file is located in a storage pool that is UNAVAILABLE, and the operation is for restore, retrieve, or recall of files to a user, or export of file data.

For details, see "Restoring Storage Pools" on page 666, "Backing Up Storage Pools" on page 648, "Recovering a Lost or Damaged Storage Pool Volume" on page 683, and "Maintaining the Integrity of Files" on page 678.

How Restore Processing Works

Two commands let you restore files from copy storage pools:

RESTORE STGPOOL
: Restores all storage pool files that have been identified as having read errors. These files are known as *damaged* files or *unreadable* files. This command also restores all files on any volumes that have been designated as *destroyed* by using the UPDATE VOLUME command. See "Restoring Storage Pools" on page 666 for details.

RESTORE VOLUME
: Recreates files that reside on a volume or volumes in the same primary storage pool. You can use this command to recreate files for one or more volumes that have been lost or damaged. See "Restoring Storage Pool Volumes" on page 669 for details.

Tivoli Storage Manager uses database information to determine which files should be restored for a volume or storage pool. As a result, restore processing does not require that the original volumes be accessed. For example, if a primary storage

pool volume is damaged, you can use the RESTORE VOLUME command to recreate files that were stored on that volume, even if the volume itself is not readable. However, if you delete the damaged files (DISCARDDATA=YES on the DELETE VOLUME command), the server removes from the database references to the files on the primary storage pool volume and to copies of the files on copy storage pool volumes. You could not restore those files.

Restore processing copies files from a copy storage pool onto new primary storage pool volumes. The server then deletes database references to files on the original primary storage pool volumes. A primary storage pool volume will become empty if all files that were stored on that volume are restored to other volumes. In this case, the server automatically deletes the empty volume from the database.

How the Destroyed Volume Access Mode Works

The *destroyed* volume access mode permits the restoration of entire volumes. This mode designates primary volumes for which files are to be restored. 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 — This command automatically changes the access mode of specified volumes to the destroyed volume access mode using a volume list provided as part of the command.
- UPDATE VOLUME — Before using this command to restore volumes in a storage pool, you must update the access mode of the volumes to destroyed.

The destroyed designation for volumes is important during restore processing, particularly when the RESTORE STGPOOL command is used to restore a large number of primary storage pool volumes after a major disaster:

- You can 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.
- After you have identified the primary volumes to be restored and set their access mode to destroyed, you can add new volumes to the storage pool. The new volumes are used to contain the files as they are restored from the copy storage pool volumes. The new volumes can also be used for new files that end users back up, archive, or migrate.
- The designation of destroyed volumes permits tracking the files that must still be restored from copy storage pools. If restore processing is ended before completion for any reason, you can restart the restore. Only the files that still reside on destroyed volumes would need to be restored.

Database and Recovery Log Protection: An Overview

The database contains information about the client data in your storage pools. The recovery log contains records of changes to the database. If you lose the recovery log, you lose the changes that have been made since the last database backup. If you lose the database, you lose all your client data.

You have several ways to protect this information:

- Mirror the database, or the recovery log, or both.
- Back up the database to tape or remote virtual volumes. See “Using Virtual Volumes to Store Data on Another Server” on page 601.

- Back up the database to tape or remote virtual volumes. See "Using Virtual Volumes to Store Data on Another Server" on page 601 for more information. In the recovery log, save all the changes made to the database since that backup (this is called *roll-forward* mode).

Mirroring

You can prevent the loss of the database or recovery log due to a hardware failure on a single drive, by mirroring drives. Mirroring simultaneously writes the same data to multiple disks. However, mirroring does not protect against a disaster or a hardware failure that affects multiple drives or causes the loss of the entire system. While Tivoli Storage Manager is running, you can dynamically start or stop mirroring and change the capacity of the database.

Mirroring provides the following benefits:

- Protection against database and recovery log media failures
- Uninterrupted operations if a database or recovery log volume fails
- Avoidance of costly database recoveries

However, there are also costs:

- Mirroring doubles the required DASD for those volumes that are mirrored
- Mirroring results in decreased performance

Database and Recovery Log Protection

Tivoli Storage Manager can perform full and incremental database backups to tape while the server is running and available to clients. With the server running in *normal* mode, the backup media can then be stored onsite or offsite and can be used to recover the database up to the point of the backup. You can run full or incremental backups as often as needed to ensure that the database can be restored to an acceptable point-in-time.

You can provide even more complete protection if you specify *roll-forward* mode. With *roll-forward* mode and an intact recovery log, you can recover the database up to its most current state (the point at which the database was lost).

For the fastest recovery time and greatest availability of the database, mirror both the database and recovery log, and periodically back up the database. When operating in roll-forward mode, mirroring better ensures that you have an intact recovery log, which is necessary to restore the database to its most current state.

Attention: If the encryption key is not available, data cannot be restored or retrieved under any circumstance. When using ENABLECLIENTENCRYPTKEY for encryption, the encryption key is stored on the server database. This means that for objects using this method, the server database must exist and have the proper values for the objects for a proper restore. Ensure that you back the server database up frequently to prevent data loss. See *Tivoli Storage Manager Using the Application Program Interface* for more information about encryption keys.

Normal Mode versus Roll-Forward Mode

Roll-forward mode offers the greatest protection for your data. However, there are costs to roll-forward mode. The following tables describe the protection afforded by each mode and the requirements for each mode.

Quality of Protection	
Normal Mode	**Roll-forward Mode**
Recover to a point-in-time of the latest full or incremental backup only.	Recover to a point-in-time of the latest full or incremental backup or, with an intact recovery log, to the most current state.
Recover the loss of client data up to the time when that data has been: • Backed up since the last database backup. • Moved due to storage pool migration, reclamation, or move data operations since the last database backup and then overwritten.	With an intact recovery log, recover to the most current state with no loss of client data.
You must restore the entire database even if only one volume is damaged.	You can restore a single volume.
Not preferable if the server supports HSM clients..	Preferable if the server supports HSM clients (space-managed files should be protected as fully as possible from hardware failure).

Storage Requirements	
Normal Mode	**Roll-forward Mode**
Does not require a recovery log to restore to a point-in-time. The recovery log keeps only uncommitted transactions, and its size is not affected by normal mode.	Requires an intact recovery log to restore to the most current state. The recovery log keeps all transactions since the last database backup. In this mode you should significantly increase the recovery log size. However: • Frequent database backups reduce recovery log storage requirements (after a backup is completed, recovery log records preceding the backup are deleted). • Mirroring the recovery log requires much less space than mirroring the database.
For the greatest availability, you should mirror the database and recovery log or place them on devices that guarantee availability.	You should mirror the recovery log to recover to the most current state. **Note:** Unlike mirroring the database, roll-forward recovery does not provide continuous operations after a media failure. This is because the database must be brought down to perform the recovery.

The following table compares four typical data recovery configurations, two for roll-forward mode and two for normal mode. In all four cases, the storage pools and the database are backed up. The benefits and costs are:

Mirroring
: Whether the database and recovery log are mirrored. Mirroring costs additional disk space.

Coverage
: How completely you can recover your data. Roll-forward recovery cannot be done if the recovery log is not intact. However, roll-forward mode does support point-in-time recovery.

Speed to Recover
: How quickly data can be recovered.

Mode	Mirroring	Quality of Protection	Speed to Recover
Roll-Forward	Log and database	Greatest	Fastest
	Log Only	Medium	Moderate
Normal	Log and database	Medium	Moderate
	None	Least	Slowest

Attention: If the log mode is set to roll-forward after a point-in-time database restoration, a database backup starts when the server is brought up for the first time. This can cause loss of data: A tape can have current data on it, but because of the point-in-time restoration, it can be marked as scratch. When the server starts for the first time, it may use this tape to write the database backup, thus destroying the original data on this tape.

This situation could occur if roll-forward mode is enabled, but the administrator restored the database as if the server was operating in normal mode, not roll-forward mode. For example: The database is to be backed up at midnight everyday Monday through Friday. On Friday, the database was restored to a point-in-time of midnight Wednesday. Thursday's database backup was not used; this tape exists and does contain valid data. But because the database was restored to Wednesday at midnight, the Thursday's tape was marked as scratch. This tape was then inadvertently chosen and written with the database backup information. Therefore, the data for Thursday was lost.

Snapshot Database Protection

A snapshot database backup is a full database backup that does not interrupt the current full and incremental backup series. Snapshot database tapes can then be taken off-site for recovery purposes and therefore kept separate from the normal full and incremental backup tapes. For information about doing a snapshot of the database, see "Doing Snapshot Database Backups" on page 661.

Mirroring the Database and Recovery Log

Mirroring can be crucial in the recovery process. Consider the following scenario: Because of a sudden power outage, a partial page write occurs. The recovery log is corrupted and not completely readable. Without mirroring, recovery operations cannot complete when the server is restarted. However, if the recovery log is mirrored and a partial write is detected, a mirror volume can be used to construct valid images of the missing data.

Note: Mirroring the Tivoli Storage Manager database and log should be considered when retention protection is enabled. If a database restore is needed, the database can be brought back to the current point in time with no data loss.

This section explains how to:

- Allocate disk volumes to mirror the database and recovery log
- Define database or recovery log mirrored volume copies
- Specify mirroring and database page shadowing server options
- Request information about mirrored volumes

Task	Required Privilege Class
Define database and recovery log volumes	System or unrestricted storage
Query mirrored volumes	Any administrator

Separating Disk Volume Copies On Separate Physical Disks When Mirroring the Database and Recovery Log

By separating volume copies on different physical devices, you protect the server from media failure and increase the availability of the database and recovery log. If you cannot assign each volume copy to its own physical disk, allocate them as shown in Table 38.

Table 38. Separating Volume Copies

Physical Disk	Database Volume	Recovery Log Volume
Physical Disk 1	Database volume copy 1	Recovery log volume copy 3
Physical Disk 2	Database volume copy 2	Recovery log volume copy 1
Physical Disk 3	Database volume copy 3	Recovery log volume copy 2

Mirrored volumes must have at least the same capacity as the original volumes.

Defining Database or Recovery Log Mirrored Volume Copies

To mirror the database or recovery log, define a volume copy for each volume in the database or recovery log.

For example, the database consists of five volumes named VOL1, VOL2, VOL3, VOL4, and VOL5. To mirror the database, you must have five volumes that match the original volumes in size. Figure 97 shows a mirrored database in which VOL1–VOL5 are mirrored by VOLA–VOLE.

Figure 97. Mirrored Volumes

Use the DSMFMT command to format the space. For example, to format VOLA, a 25MB database volume, enter:

```
./dsmfmt -m -db vola 25
```

Then define the group of mirrored volumes. For example, you might enter the following commands:

```
define dbcopy vol1 vola

define dbcopy vol2 volb
```

```
define dbcopy vol3 volc

define dbcopy vol4 vold

define dbcopy vol5 vole
```

After a volume copy is defined, the volume copy is synchronized with the original volume. This process can range from minutes to hours, depending on the size of the volumes and performance of your system. After synchronization is complete, the volume copies are mirror images of each other.

Specifying Mirroring and Database Page Shadowing Server Options

Four server options let you specify the level of protection, recoverability, and performance for mirrored volumes:

- MIRRORREAD specifies how mirrored volumes are accessed when the server reads the recovery log or a database page during normal processing. You may specify MIRRORREAD LOG for reading recovery log pages, or MIRRORREAD DB for reading database pages. MIRRORREAD LOG (or DB) NORMAL specifies that only one mirrored volume is read to obtain the desired page. MIRRORREAD LOG (or DB) VERIFY specifies that all mirrored volumes for a page be read, compared, and re-synchronized if necessary. MIRRORREAD LOG (or DB) VERIFY can decrease server performance as each mirrored volume for the page is accessed on every read.
- MIRRORWRITE specifies how mirrored volumes are written to. You may issue MIRRORWRITE LOG or DB, and then specify that write operations for the database and the recovery log be specified as SEQUENTIAL or PARALLEL:
  - A PARALLEL specification offers better performance but at the potential cost of recoverability. Pages are written to all copies at about the same time. If a system outage results in a partial page write and the outage affects both mirrored copies, then both copies could be corrupted.
  - A SEQUENTIAL specification offers improved recoverability but at the cost of performance. Pages are written to one copy at a time. If a system outage results in a partial page write, only one copy is affected. However, because a successful I/O must be completed after the write to the first copy but before the write to the second copy, performance can be affected.
- DBPAGESHADOW=YES mirrors the latest batch of pages written to a database. In this way if an outage occurs that affects both mirrored volumes, the server can recover pages that have been partially written. If no name is specified in the DBPAGESHADOWFILE option, a dbpgshdw.bdt file will be created and used. If the DBPAGESHADOWFILE option specifies a file name, that file name will be used.
- DBPAGESHADOWFILE specifies the name of the database page shadowing file.

  DBPAGESHADOW and DBPAGESHADOWFILE coordinate with the MIRRORWRITE server option and its specifications of DB and SEQUENTIAL or PARALLEL like this:

DBPAGESHADOW Setting	Number of Database Volumes with Mirrors Defined	Number of Database Volumes with Mirrors Synchronized and Available	MIRRORWRITE Database Setting	DBPAGESHADOW Behavior
DBPAGESHADOW Setting	**Number of Database Volumes with Mirrors Defined**	**Number of Database Volumes with Mirrors Synchronized and Available**	**MIRRORWRITE Database Setting**	**DBPAGESHADOW Behavior**
NO	Some database volumes	Some database volumes	PARALLEL or SEQUENTIAL	No database page shadowing will be performed.
YES	Some database volumes	Some database volumes	PARALLEL or SEQUENTIAL	Database page shadowing will be performed.
YES	All database volumes	Some database volumes	PARALLEL or SEQUENTIAL	Database page shadowing will be performed.
YES	All database volumes	All database volumes	PARALLEL	Database page shadowing will be performed. These settings provide additional protection to the database when in PARALLEL write mode.
YES	All database volumes	All database volumes	SEQUENTIAL	Database page shadowing will **not** be performed. These settings provide no additional protection to the database when in SEQUENTIAL write mode.
YES	No database volumes	No database volumes	PARALLEL or SEQUENTIAL	Database page shadowing will be performed.

Requesting Information about Mirrored Volumes

You can request information about mirrored database or recovery log volumes by using the QUERY DBVOLUME and QUERY LOGVOLUME commands. For example:

```
query dbvolume
```

The following type of information is displayed:

```
Volume Name    Copy    Volume Name    Copy    Volume Name    Copy
(Copy 1)       Status  (Copy 2)       Status  (Copy 3)       Status
-----------    ------  -------------  ------  -------------  ------
VOL1           Sync'd  VOLA           Sync'd                 Undef-
VOL2           Sync'd  VOLB           Sync'd                 ined
VOL3           Sync'd  VOLC           Sync'd
VOL4           Sync'd  VOLD           Sync'd

VOL5           Sync'd  VOLE           Sync'd
```

- Each pair of vertical columns displays an image of the database or recovery log. For example, VOLA, VOLB, VOLC, VOLD, and VOLE (Copy 2) represent one image of the database.
- Each horizontal row displays a *group of mirrored volumes*. For example, VOL1, and VOLA represent the two volume copies.

Backing Up Storage Pools

Task	Required Privilege Class
Define, back up, or restore storage pools Restore volumes	System, unrestricted storage, or restricted storage (only for those pools to which you are authorized)

You can back up primary storage pools to copy storage pools to improve data availability. When you back up a primary storage pool, you create backup copies of client files that are stored in primary storage pools in copy storage pools. By using copy storage pools, you maintain multiple copies of files and reduce the potential for data loss due to media failure. If the primary file is not available or becomes corrupted, the server accesses and uses the duplicate file from a copy storage pool.

Figure 98. Copy Storage Pools

Primary storage pools should be backed up each day to the same copy storage pool. Backing up to the same copy storage pool ensures that files do not need to be recopied if they have migrated to the next pool.

For example, to back up the ARCHIVEPOOL primary storage pool to the DISASTER-RECOVERY copy storage pool, enter:

```
backup stgpool archivepool disaster-recovery maxprocess=4
```

The only files backed up to the DISASTER-RECOVERY pool are files for which a copy does not already exist in the copy storage pool.

Note: To back up a primary storage pool, the DATAFORMAT must be NATIVE or NONBLOCK.

The command example above uses four parallel processes to perform an incremental backup of the primary storage pool to the copy storage pool (MAXPROCESS=4). Set the MAXPROCESS parameter in the BACKUP STGPOOL command to the number of mount points or drives that can be dedicated to this operation.

Because the backup copies are made incrementally, you can cancel the backup process. Reissuing the BACKUP STGPOOL command lets the backup continue from the spot the backup was canceled.

You can back up multiple primary storage pools to one copy storage pool. If multiple copies are necessary, you can also back up a primary storage pool to multiple copy storage pools. However, you should back up the entire primary storage pool hierarchy to the same copy storage pool for easier management of storage volumes. See "Using Multiple Copy Storage Pools" on page 651.

See the following sections:

"Scheduling Storage Pool Backups"
"Example: Simple Hierarchy with One Copy Storage Pool" on page 650
"Using Simultaneous Write to Copy Storage Pools" on page 650
"Using Multiple Copy Storage Pools" on page 651
"Delaying Reuse of Volumes for Recovery Purposes" on page 651

For recovery scenarios that use backed-up copies of storage pools, see "Recovering to a Point-in-Time from a Disaster" on page 681 and "Recovering a Lost or Damaged Storage Pool Volume" on page 683.

Notes:

1. 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, reevaluate your database size requirements.
2. If a copy is to be made in a copy storage pool and a copy already exists with the same insertion date, no action is taken.
3. 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.
4. Files in a copy storage pool do not migrate to another storage pool.
5. After a file is copied to a copy storage 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.

Scheduling Storage Pool Backups

For the best protection, primary storage pools should be backed up regularly, preferably each day. You can define schedules to begin backups of files in the

primary storage pools. For example, to back up the BACKUPPOOL, ARCHIVEPOOL, and TAPEPOOL storage pools every night, schedule the following commands:

```
backup stgpool backuppool disaster-recovery maxprocess=4

backup stgpool archivepool disaster-recovery maxprocess=4

backup stgpool tapepool disaster-recovery maxprocess=4
```

See Chapter 18, "Automating Server Operations," on page 471 for information about scheduling commands.

If you schedule storage pool backups and migrations and have enough disk storage, you can copy most files from the disk storage pool before they are migrated to tape and thus avoid unnecessary tape mounts. Here is the sequence:

1. Clients back up or archive data to disk
2. You issue or schedule BACKUP STGPOOL commands to back up the primary storage pools to copy storage pools
3. Data migrates from disk storage pools to primary tape storage pools

Example: Simple Hierarchy with One Copy Storage Pool

Assume that you have two primary storage pools: one random access storage pool (DISKPOOL) and one tape storage pool (TAPEPOOL, with device class TAPECLASS). Files stored in DISKPOOL are migrated to TAPEPOOL. You want to back up the files in both primary storage pools to a copy storage pool.

To schedule daily incremental backups of the primary storage pools, do the following:

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
   ```

   Notes:

   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.

   To minimize tape mounts, back up the disk storage pool first, then the tape storage pool.

   For more information about scheduling, see Chapter 18, "Automating Server Operations," on page 471.

Using Simultaneous Write to Copy Storage Pools

You can set up a primary storage pool so that when a client backs up, archives, or migrates a file, the file is written to the primary storage pool and is simultaneously stored into each copy storage pool specified for the primary storage pool.

Use of the simultaneous write function is not intended to replace regular backups of storage pools. If you use the function to simultaneously write to copy storage pools, ensure that the copy of each primary storage pool is complete by regularly issuing the BACKUP STGPOOL command. See "Using the Simultaneous Write Function to Protect Your Data" on page 249.

Using Multiple Copy Storage Pools

Occasionally, when Tivoli Storage Manager restores data, there is some duplication of restored files. This can occur if primary volumes are not available, and Tivoli Storage Manager does not have a complete copy storage pool from which to perform the restore. Then, Tivoli Storage Manager must use volumes from multiple copy storage 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 available to the server, or ensure that only one copy storage pool has an access of read/write during the restore operation.

Duplication of restored files only occurs when these conditions exist:

- Primary volumes are unavailable or offsite.
- Multiple copy storage pools are available.
- Copy storage pools do not contain all of the files that are in the primary storage pools.

The following example explains this scenario:

The primary storage pool Main contains volumes Main1, Main2, and Main3.

- Main1 contains files File11, File12, File13
- Main2 contains files File14, File15, File16
- Main3 contains files File17, File18, File19

The copy storage pool DuplicateA contains volumes DupA1, DupA2, and DupA3.

- DupA1 contains copies of File11, File12
- DupA2 contains copies of File13, File14
- DupA3 contains copies of File15, File16, File17, File18 (File19 is missing because BACKUP STGPOOL was run on the primary pool before the primary pool contained File 19.)

The copy storage pool DuplicateB contains volumes DupB1 and DupB2.

- DupB1 contains copies of File11, File12
- DupB2 contains copies of File13, File14, File15, File16, File17, File18, File19

If you have not designated copy storage pool DuplicateB as the only copy storage pool to have read/write access for the restore operation, then Tivoli Storage Manager can choose the copy storage pool DuplicateA, and use volumes DupA1, DupA2, and DupA3. Because copy storage pool DuplicateA does not include file File19, Tivoli Storage Manager would then use volume DupB2 from the copy storage pool DuplicateB. The program does not track the restoration of individual files, so File15, File16, File17, and File18 will be restored a second time, and duplicate copies will be generated when volume DupB2 is processed.

Delaying Reuse of Volumes for Recovery Purposes

When you define or update a sequential access storage pool, you can use the REUSEDELAY parameter. This parameter specifies the number of days that must

elapse before a volume can be reused or returned to scratch status after all files have been expired, deleted, or moved from the volume. When you delay reuse of such volumes and they no longer contain any files, they enter the *pending* state. Volumes remain in the pending state for as long as specified with the REUSEDELAY parameter for the storage pool to which the volume belongs.

Delaying reuse of volumes can be helpful under certain conditions for disaster recovery. When files are expired, deleted, or moved from a volume, they are not actually erased from the volumes: The database references to these files are removed. Thus the file data may still exist on sequential volumes if the volumes are not immediately reused.

A disaster may force you to restore the database using a database backup that is not the most recent backup. In this case, some files may not be recoverable because the server cannot find them on current volumes. However, the files may exist on volumes that are in pending state. You may be able to use the volumes in pending state to recover data by doing the following:

1. Restore the database to a point-in-time prior to file expiration.
2. Use a primary or copy storage pool volume that has not been rewritten and contains the expired file at the time of database backup.

If you back up your primary storage pools, set the REUSEDELAY parameter for the primary storage pools to 0 to efficiently reuse primary scratch volumes. For your copy storage pools, you should delay reuse of volumes for as long as you keep your oldest database backup.

For an example of using database backup and delaying volume reuse, see "Protecting Your Database and Storage Pools" on page 680. For information about expiration, see "Running Expiration Processing to Delete Expired Files" on page 391.

Backing Up the Database

Backing up the database is a simple operation. You can back up the database with full and incremental backups or by taking a snapshot of a specific point-in-time of the database; these are called snapshot database backups. (See "Doing Full and Incremental Backups" on page 660 and "Doing Snapshot Database Backups" on page 661 for more information.) Before your first backup, you must do some or all of the following steps:

- Define device classes for backups
- Set the recovery log mode
- Schedule database backups
- Estimate the recovery log size
- Automate database backups to occur according to a defined schedule or when the recovery log utilization reaches a specified percentage.

To restore your database, you must have copies of (or be able to create) the following information:

- Volume history file
- Device configuration file
- Server options file
- Database and recovery log set up (the output from detailed queries of your database and recovery log volumes).

Defining Device Classes for Backups

You can use existing device classes for backups or define new ones. You can also specify different device classes for incremental backups and for full backups. For example, you might want to write full backups to tape and incremental backups to disk. Specifying a device class with a device type of FILE is useful if an incremental backup is run based on a database backup trigger. You should do this only if you are also backing up the files to tape and taking them off site. Otherwise, in a disaster you can only restore the full backup.

You can also reserve a device class, and therefore a device, for automatic backups only. In this way, the server does not try to back up the database with no device available. If a database backup shares a device class with a low priority operation, such as reclamation, and all the devices are in use, the lower priority operation is automatically canceled. This frees a device for the database backup.

Note: Device class definitions are saved in the device configuration files (see "Saving the Device Configuration File" on page 658).

Setting the Recovery Log Mode

You can set the recovery log mode to either *normal* or *roll-forward*. See "Database and Recovery Log Protection" on page 642 for a description of the two modes and for a comparison their benefits and costs.

If you do not set the recovery log mode, the server runs in normal mode. To set the log mode to roll-forward, enter:

```
set logmode rollforward
```

Note: The log mode is not in roll-forward mode until you perform the first full database backup after entering this command.

To set the log mode back to normal, enter:

```
set logmode normal
```

Note: Using the roll-forward log mode should be considered from a server when the retention protection mode is enabled. If a database restore is needed, the database can be brought back to the current point in time with no data loss.

Estimating the Size of the Recovery Log

The number of transactions affect how large you should make your recovery log. As you add more clients and increase concurrent transactions, you can extend the size of the log. In roll-forward mode you should also consider how often you perform database backups. In this mode, the recovery log keeps all transactions since the last database backup and typically requires much more space than normal mode does.

To determine the size that the recovery log should be in roll-forward mode, you must know how much recovery log space is used between database backups. For example, if you perform daily incremental backups, check your daily usage over a period of time. You can use the following procedure to make your estimate:

1. Set the log mode to normal. In this way you are less likely to exceed your log space if your initial setting is too low for roll-forward mode.
2. After a scheduled database backup, reset the statistic on the amount of recovery log space used since the last reset by using the following command:

```
reset logconsumption
```

3. Just before the next scheduled database backup, display the current recovery log statistics by using the following command:

   ```
   query log format=detailed
   ```

   Record the *cumulative consumption* value, which shows the space, in megabytes, used since the statistic was last reset.
4. Repeat steps 2 on page 653 and 3 for at least one week.
5. Increase the highest cumulative consumption value by 30 percent. Set your recovery log size to this increased value to account for periods of unusually high activity.

   For example, over a period of a week the highest cumulative consumption value was 500MB. If you set your recovery log to 650MB, you should have enough space between daily backups.

For information on how to adjust the recovery log size, see "Increasing the Size of the Database or Recovery Log" on page 496 or "Decreasing the Size of the Database or Recovery Log" on page 501.

Note: If the recovery log runs out of space, you may not be able to start the server for normal operation. You can create an additional recovery log volume if needed to start the server and perform a database backup. For example, to create a 5MB volume A00, issue the following command:

```
> dsmserv extend log a00 5mb
```

Specify volume sizes in multiples of 4MB plus 1MB for overhead.

Scheduling Database Backups

Database backups require devices, media, and time. Consider scheduling backups to occur at certain times of the day and after activities such as the following:

- Major client backup or archive activities
- Storage pool migration and reclamation
- Storage pool backups
- MOVE DATA or DELETE VOLUME commands

Depending on the frequency of these activities and the amount of client data, you might back up your storage pools daily and then immediately back up the database.

Consider the following when you decide what kind of backups to do and when to do them:

- Full backups take longer than incremental backups
- Full backups have shorter recovery times than incremental backups (you must load only one set of volumes to restore the entire database)
- Full backups are required:
  - For the first backup
  - If there have been 32 incremental backups since the last full backup
  - After changing the log mode to roll-forward
  - After changing the database size (an extend or reduce operation)

Automating Database Backups

In roll-forward mode, you can set a database backup to occur automatically when the recovery log utilization reaches a defined percentage. Once the backup is complete, the server automatically deletes any unnecessary recovery log records, thus reducing the recovery log utilization. You can also control automatic database backups by how much log space will be freed and how much time has elapsed since the last backup. You might want to automatically trigger database backups if you have already scheduled them. However, while the automatically triggered database backups are occurring, the recovery log could grow faster than expected or the server may not be able to significantly reduce the recovery log utilization as a result of the backup. You should try to coordinate the recovery log size and scheduled database backups. A database backup has a higher priority than most operations, and backup based on a trigger could occur during high server activity and affect your other operations. Adjust the recovery log size and database backup trigger parameters to avoid triggering backups at non-scheduled times.

By setting a database backup trigger you can reduce the likelihood that the recovery log will run out of space before the next backup.

If the log mode is changed from normal to roll-forward, the next database backup must be a full backup. If a database backup trigger is defined when you set the log mode to roll-forward, the full backup is done automatically. The server does not start saving log records for roll-forward recovery until this full backup completes successfully.

You can determine the size of your recovery log by completing the steps in "Estimating the Size of the Recovery Log" on page 653. Define your database backup trigger according to the size of your recovery log. For example, assume that your recovery log size is 650MB. The recovery log utilization percentage is usually less than 500MB between database backups. You want to trigger a backup only in unusual circumstances. Therefore, set the trigger to at least 75 percent (approximately 500MB). To set the trigger to 75 percent and run 20 incremental backups to every full backup, enter:

```
define dbbackuptrigger logfullpct=75 devclass=tapeclass
 numincremental=20 mininterval=120 minlogfreepct=10
```

Each incremental backup, whether automatic or manual, is added to the count of incremental backups. Each full backup, whether automatic or manual, resets the count for incremental backups to 0. If you specify a NUMINCREMENTAL value of 0, the server automatically runs only full backups.

Note: If you issue a BACKUP DB command with the TYPE=INCREMENTAL parameter, the server performs an incremental backup of the database regardless of the NUMINCREMENTAL setting. For example, if you set NUMINCREMENTAL to 5, there will be 5 incremental backups following the last full backup. If you then issue BACKUP DB TYPE=INCREMENTAL, an incremental backup is still done, and the incremental backup counter is set to 6. This occurs if the BACKUP DB command is issued either by an administrator or through an administrative schedule.

After you set the database backup trigger, you might find that automatic backups occur too often. Check the backup trigger parameters by entering:

```
query dbbackuptrigger format=detailed
```

The following information is displayed:

```
             Full Device Class: TAPECLASS
      Incremental Device Class: TAPECLASS
           Log Full Percentage: 75
    Incrementals Between Fulls: 20
       Minimum Backup Interval: 120
  Minimum Log Percentage Freed: 10
Last Update by (administrator): SERVER_CONSOLE
         Last Update Date/Time: 02/27/2002 12:57:52
```

This information shows that the database will be backed up if the log is at least 75 percent full and either the last database backup was at least two hours (120 minutes) ago, or there is at least 10 percent of the log that will be freed after the backup completes. If automatic backups are occurring too often, you could increase the log full percentage, the minimum interval, or the minimum amount of log space to be freed. You could increase the log full percentage to 80, increase the minimum interval to eight hours (480 minutes), and increase the minimum amount of log space to be freed to 30 percent by entering:

```
update dbbackuptrigger logfullpct=80 mininterval=480 minlogfreepct=30
```

If the database backup trigger automatically runs backups more often than you want and the setting is high (for example, 90 percent or higher), you should probably increase the recovery log size. If you no longer want to use the database backup trigger, enter:

```
delete dbbackuptrigger
```

After you delete the database backup trigger, the server no longer runs automatic database backups.

Note: If you delete the trigger and stay in roll-forward mode, transactions fail when the log fills. Therefore, you should change the log mode to normal. Remember, however, that normal mode does not let you perform roll-forward recovery. Increase the recovery log size if you want roll-forward recovery.

Saving the Volume History File

Volume history information is stored in the database, but during a database restore, it is not available from there. To perform a restore, therefore, the server must get the information from the volume history file. It is very important to save your volume history file so that you do not have to manually examine every volume.

The following volume information is stored in the database:

- Sequential access storage pool volumes that have been added, reused (through reclamation or move data operations), or deleted (during delete volume or reclamation operations)
- Full and incremental database backup volume information
- Export volumes for administrator, node, policy, and server data
- Snapshot database volume information
- Backup set volume information.

The server updates the volume history file as volumes are added. However, you must periodically run a delete operation to discard outdated information about volumes (see “Deleting Volume History Information” on page 657 for details).

To ensure the availability of volume history information, it is extremely important to do one of the following:

- 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.

Note: You can recover the database without a volume history file. However, because you must examine every volume that may contain database backup information, this is a time-consuming and error-prone task.

The VOLUMEHISTORY server option lets you specify backup volume history files. Then, whenever the server updates volume information in the database, it also updates the same information in the backup files.

You can also back up the volume history information at any time, by entering:

```
backup volhistory
```

If you do not specify file names, the server backs up the volume history information to all files specified with the VOLUMEHISTORY server option.

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

- Not halt the server for a few minutes after issuing the BACKUP VOLHISTORY command.
- Examine the volume history file to see if the file is updated.

Deleting Volume History Information

You should periodically delete outdated information from the volume history file. For example, if you keep backups for seven days, information older than that is not needed. When information about database backup volumes or export volumes is deleted, the volumes return to scratch status. For scratch volumes of device type FILE, the files are deleted. When information about storage pools volumes is deleted, the volumes themselves are not affected.

To display volume history information up to yesterday, enter:

```
query volhistory enddate=today-1
```

To delete information that is seven days old or older, enter:

```
delete volhistory type=all todate=today-8
```

Notes:

1. Existing volume history files are *not* automatically updated with the DELETE VOLHISTORY command.
2. Do not delete sequential volume history information until you no longer need that information. For example, do not delete dump volume information or storage volume reuse information, unless you have backed up or dumped the database at a later time than that specified for the delete operation.
3. Do not delete the volume history information for database dump, database backup, or export volumes that reside in automated libraries, unless you want to return the volumes to scratch status. When the DELETE VOLHISTORY command removes volume information for such volumes, they automatically

return to scratch status. The volumes are then available for reuse by the server and the information stored on them may be overwritten.

Saving the Device Configuration File

Make a copy of your device configuration file and save it.

The device configuration file contains information required to read backup data. This information includes the following:

- Devices class definitions
- Library definitions
- Drive definitions
- Path definitions
- Server definitions

This information is stored in the database, but during a database restore, it is not available from there. To perform a restore, therefore, the server must get the information from the device configuration file. When device information is updated in the database, it is also updated in the device configuration file. The device information must match the devices configured on the system where the restore will be performed. You may have to edit those commands in an existing file so that they match.

Only path definitions with SRCTYPE=SERVER are backed up to the device configuration file. Paths of SRCTYPE=DATAMOVER are not written out to the file.

To ensure the availability of the device configuration information, it is extremely important that you do one of the following:

- Store at least one backup copy of the device configuration file on a disk separate from the database
- Store your device configuration file offsite with your volume history file and database backups
- Store a printout of the information that is stored offsite
- Store a remote copy, for example, on an NFS-mounted file system

The DEVCONFIG server option lets you specify backup device configuration files (for details, see the *Administrator's Reference*). After the server is restarted, whenever the server updates device configuration information in the database, it also updates the same information in the backup files.

During a database restore operation, the server tries to open the first device configuration file in the order in which the files occur in the server options. If it cannot read that file, it searches for the next usable device configuration file. If none can be found, you must recreate the file. See "Recreating a Device Configuration File" on page 659 for details. After the database has been restored, you may have to update the device configuration.

You can also back up the device configuration information at any time, by entering:

```
backup devconfig
```

If you do not specify file names, the device configuration file is backed up to *all* files specified with the DEVCONFIG server option.

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

- Not halt the server for a few minutes after issuing the BACKUP DEVCONFIG command.
- Examine the device configuration file to see if the file is updated.

If you lose the device configuration file and need it to restore the database, you must recreate it manually. See "Recreating a Device Configuration File" for details.

If you are using automated tape libraries, volume location information is also saved in the device configuration file. The file is updated whenever CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME, and AUDIT LIBRARY commands are issued, and the information is saved as comments (/* ..... */). This information is used during restore or load operations to locate a volume in an automated library. If you must recreate the device configuration file, you will be unable to recreate the volume location information. Therefore, you must define your library as a manual library and manually mount the volumes during server processing. 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 may 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.

Updating the Device Configuration File

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.

If a disaster occurs, you may have to restore Tivoli Storage Manager by using devices other than those that are included in the device configuration file. In such a case, you will have to update the device configuration files manually with information about the new devices.

Definitions for paths are included when SRCTYPE=SERVER.

Library volume location information is updated in the device configuration file whenever CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME, and AUDIT LIBRARY commands are issued for SCSI libraries.

Recreating a Device Configuration File

The following commands read and execute the device configuration file:

- DSMSERV RESTORE DB

- DSMSERV LOADDB
- DSMSERV DISPLAY DBBACKUPVOLUME

If no device configuration file is found, you must recreate it before you can start the restore operation. The device configuration file must follow these conventions:

- The commands must be in this order:
  - DEFINE SERVER (if you are using virtual volumes)
  - DEFINE DEVCLASS
  - DEFINE LIBRARY
  - DEFINE DRIVE
  - DEFINE PATH

  You must provide those definitions needed to mount the volumes read by the command that you issued. If you are restoring or loading from a FILE device class, you will need only the DEFINE DEVCLASS command.
- 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 may 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.
- You can use command defaults.
- The file can include blank lines.
- A single line can be up to 240 characters.
- The file can include continuation characters and comments as described in the *Administrator's Reference*.

The following shows an example of a device configuration file:

```
/*  Tivoli Storage Manager Device Configuration */
define devclass tapeclass devtype=8mm library=manuallib
define library manuallib libtype=manual
define drive manuallib drive02
define path server1 drive02 srctype=server destype=drive
 library=manuallib device=/dev/mt2
```

Saving the Server Options

You should make a copy of your server options file and save it.

Saving the Database and Recovery Log Information

You should make copies of this output and save them. The database and recovery log setup information is the output from detailed queries of your database and recovery log volumes.

Doing Full and Incremental Backups

The first backup of your database must be a full backup. You can run up to 32 incremental backups between full backups.

To perform a full backup of your database to the TAPECLASS device class, enter:

```
backup db type=full devclass=tapeclass
```

In this example, the backup data is written to scratch volumes. You can also specify volumes by name. After a full backup, you can perform incremental backups, which copy only the changes to the database since the previous backup.

To do an incremental backup of the database to the TAPECLASS device class, enter:

```
backup db type=incremental devclass=tapeclass
```

Doing Snapshot Database Backups

A snapshot database backup is a full database backup that does not interrupt the current full and incremental backup series. Snapshot database tapes can then be taken off-site for recovery purposes and therefore kept separate from the normal full and incremental backup tapes. Snapshot database backups enhance the protection of your server and its data while maintaining the full and incremental database backup series. Although snapshot database backups cannot restore a database or a database volume to its most current state, you can use them to restore a database to a specific point-in-time.

Snapshot database backups:

- Copy the complete contents of a database, just like a full database backup.
- Create a new database backup series without interrupting the existing full and incremental backup series for the database.
- Do not truncate the server recovery log when the server is running in roll-forward mode.

Use the BACKUP DB command to perform a snapshot database backup. New volume history entries are created for the snapshot database volumes.

To perform a snapshot database backup to the TAPECLASS device class, enter:

```
backup db type=dbsnapshot devclass=tapeclass
```

Note: Snapshot database backups should be used as an adjunct to full and incremental backups. When the server is in roll-forward mode, and a snapshot database backup is performed, the recovery log keeps growing. When full and incremental backups are performed with roll-forward mode enabled, the recovery log is restarted each time a full backup is performed.

Recovering Your Server Using Database and Storage Pool Backups

This section explains how to recover by using backups of the database and storage pools. Figure 99 on page 662 shows the situation presented in the two scenarios in this section: an installation has lost its server, including the database and recovery log, and its onsite storage pools.

Figure 99. Recovery from a Disaster

The following topics are included:

- Restoring to a point-in-time
- Restoring to the most current state

To perform a restore, you should have the following information, preferably stored offsite (see Figure 99):

- A full database backup
- Any incremental database backups between the last full backup and the point-in-time to which you are recovering
- Copy storage pool volumes
- On tape or diskette, or as printouts:
  - Server options file
  - Volume history file
  - Device configuration file with the applicable device information (library, drive, and device class definitions)
  - Database and recovery log setup (the output from detailed queries of your database and recovery log volumes)

Note: To perform a DSMSERV RESTORE DB operation, the database backup volumes must be in a library of a library type of MANUAL or SCSI.

Restoring a Database to a Point-in-Time

Point-in-time recovery is normally used for situations such as disaster recovery or to remove the effects of errors that can cause inconsistencies in the database.

You can use either full and incremental backups or snapshot database backups to restore a database to a point-in-time.

For a scenario of recovering to a point-in-time, see "Backup and Recovery Scenarios" on page 680.

Here is the procedure for restoring the database:

1. Rename and save a copy of the volume history file if it exists. After the database is restored, any volume history information pointed to by the server options is lost. You will need this information to identify the volumes to be audited. If you do not have a volume history file, see "Point-in-Time Restore Without a Volume History File" on page 665.
2. If the device configuration file is unavailable, recreate it manually (see "Recreating a Device Configuration File" on page 659). Put the existing or recreated device configuration file in the server work library. Do the same with the server options file. Have available your outputs from your detailed queries about your database and recovery log setup information.

   You may need to modify the device configuration file based on the hardware available at the recovery site. For example, the recovery site might require a different device class, library, and drive definitions. For more information, see "Updating the Device Configuration File" on page 659.
3. If the original database or recovery log volumes were lost, issue the DSMSERV FORMAT utility to initialize the database and recovery log. For example:

   ```
   dsmserv format 1 log1 1 dbvol1
   ```

   Attention: Do not start the server until *after* you restore the database (the next step). Starting the server before the restore would destroy any existing volume history files.
4. Issue the DSMSERV RESTORE DB utility. For example, to restore the database to a backup series that was created on April 19, 1999, enter:

   ```
   dsmserv restore db todate=04/19/1999
   ```

   The server does the following:

   a. Reads the volume history file to locate the last full backup that occurred on or before the specified date and time.

      Note: If the volume history file is not available, you must mount tape volumes in the correct order or specify their order on the DSMSERV RESTORE DB utility.

   b. Using the device configuration file, requests a mount of the first volume, which should contain the beginning of the full backup.
   c. Restores the backup data from the first volume.
   d. Continues to request mounts and to restore data from the backup volumes that contain the full backup and any incremental backups that occurred on or before the date specified.

   From the old volume history information (generated by the QUERY VOLHISTORY command) you need a list of all the volumes that were reused (STGREUSE), added (STGNEW), and deleted (STGDELETE) since the original backup. Use this list to perform the rest of this procedure.

   It may also be necessary to update the device configurations in the restored database.
5. Audit all disk volumes, all reused volumes, and any deleted volumes located by the AUDIT VOLUME command using the FIX=YES parameter.

   This process identifies files recorded in the database that can no longer be found on the volume. If a copy of the file is in a copy storage pool, the file on the audited volume is marked as damaged. Otherwise, the file is deleted from the database and is lost.

6. If the audit detects any damaged files, issue the RESTORE STGPOOL command to restore those files after you have audited the volumes in the storage pool. Include the FIX=YES parameter on the AUDIT VOLUME command to delete database entries for files not found in the copy storage pool.
7. 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.
8. Redefine any storage pool volumes that were added since the database backup.

Notes:

1. Some files may be lost if they were moved since the backup (due to migration, reclamation, or move data requests) and the space occupied by those files has been reused. You can minimize this loss by using the REUSEDELAY parameter when defining or updating sequential access storage pools. This parameter delays volumes from being returned to scratch or being reused. See "Delaying Reuse of Volumes for Recovery Purposes" on page 651 for more information on the REUSEDELAY parameter.
2. By backing up your storage pool and your database, you reduce the risk of losing data. To further minimize loss of data, you can:
   - Mark the backup volumes in the copy storage pool as OFFSITE and move them to an offsite location.

     In this way the backup volumes are preserved and are not reused or mounted until they are brought onsite. Ensure that you mark the volumes as OFFSITE before you back up the database.

     To avoid having to mark volumes as offsite or physically move volumes:
     - Specify a device class of DEVTYPE=SERVER in your database backup.
     - Back up a primary storage pool to a copy storage pool associated with a device class of DEVTYPE=SERVER.
   - Back up the database immediately after you back up the storage pools.
   - Turn off migration and reclamation while you back up the database.
   - Do not perform any MOVE DATA operations while you back up the database.
   - Use the REUSEDELAY parameter's interval to prevent your copy storage pool volumes from being reused or deleted before they might be needed.
3. If your old volume history file shows that any of the copy storage pool volumes needed to restore your storage pools have been reused (STGREUSE) or deleted (STGDELETE), you may not be able to restore all your files. You can avoid this problem by including the REUSEDELAY parameter when you define your copy storage pools.
4. After a restore, the volume inventories for Tivoli Storage Manager and for your tape management system may be inconsistent. For example, after a database backup, a new volume is added to Tivoli Storage Manager. The tape management system inventory records the volume as belonging to Tivoli Storage Manager. If the database is restored from the backup, Tivoli Storage Manager has no record of the added volume, but the tape management system does. You must synchronize these inventories. Similarly, the volume inventories for Tivoli Storage Manager and for any automated libraries may also be inconsistent. If they are, issue the AUDIT LIBRARY command to synchronize these inventories.

Point-in-Time Restore Without a Volume History File

You can use either full and incremental backups or snapshot database backups to restore a database to a point-in-time.

If you are doing a point-in-time restore and a volume history file is not available, you must enter the volume names in the DSMSERV RESTORE DB utility in the sequence in which they were written to. First, however, issue the DSMSERV DISPLAY DBBACKUPVOLUME utility to read your backup volumes and display the information needed to arrange them in order (backup series, backup operation, and volume sequence). For example:

```
dsmserv display dbbackupvolume devclass=tapeclass
 volumenames=dsm012,dsm023,dsm037,dsm038,dsm058,dsm087
```

For example, the most recent backup series consists of three operations:

0 A full backup on three volumes in the sequence dsm023, dsm037, and dsm087

1 An incremental backup on one volume, dsm012

2 An incremental backup on two volumes in the sequence dsm038 and dsm058

You would issue three commands in the following order:

```
dsmserv restore db volumenames=dsm023,dsm037,dsm087
 devclass=tapeclass commit=no
dsmserv restore db volumenames=dsm012
 devclass=tapeclass commit=no
dsmserv restore db volumenames=dsm038,dsm058
 devclass=tapeclass commit=no
```

Attention: If the original database or recovery log volumes are available, you issue only the DSMSERV RESTORE DB utility. However, if those volumes have been lost, you must first issue the DSMSERV FORMAT command to initialize the database and recovery log, then issue the DSMSERV RESTORE DB utility.

Storage Pool Backups in a Point-of-Time Restore

The following example shows the importance of storage pool backups with a point-in-time restore. In this example, the storage pool was not backed up with the BACKUP STGPOOL command.

9:30 a.m.
Client A backs up its data to Volume 1.

Noon The system administrator backs up the database.

1:30 p.m.
Client A's files on Volume 1 (disk), is migrated to tape (Volume 2).

3:00 p.m.
Client B backs up its data to Volume 1.

The server places Client B's files in the location that contained Client A's files prior to the migration.

3:30 p.m.
The server goes down.

3:40 p.m.
The system administrator reloads the noon version of the database by using the DSMSERV RESTORE DB utility.

4:40 p.m.

Volume 1 is audited. The following then occurs:

1. The server compares the information on Volume 1 and with the restored database (which matches the database at noon).
2. The audit does not find Client A's files on Volume 1 where the reloaded database indicates they should be. Therefore, the server deletes these Client A file references.
3. The database has no record that Client A's files are on Volume 2, and the files are, in effect, lost.
4. The database has no record that Client B's files are on Volume 1, and the files are, in effect, lost.

If roll-forward recovery had been used, the database would have been rolled forward to 3:30 p.m. when the server went down. In this case, neither Client A's files nor Client B's files would have been lost. If a point-in-time restore of the database had been performed and the storage pool had been backed up, Client A's files would not have been deleted by the volume audit. Those files could have been restored with a RESTORE VOLUME or RESTORE STGPOOL command. Client B's files would still have been lost, however.

Restoring a Database to its Most Current State

You can use roll-forward recovery to restore a database to its most current state if:

- The server has been in roll-forward mode continuously from the time of the last full backup to the time the database was damaged or lost.
- The last backup series created for the database is available. A backup series consists of a full backup, all applicable incremental backups, and all recovery log records for database changes since the last backup in the series was run.

You can only use full and incremental backups with roll-forward mode enabled to restore a database to its most current state. Snapshot database backups are complete database copies of a point-in-time.

To restore the database to its most current state, enter:

```
dsmserv restore db
```

Attention: If the original database or recovery log volumes are available, you issue only the DSMSERV RESTORE DB utility. However, if those volumes have been lost, you must first issue the DSMSERV FORMAT utility to initialize the database and recovery log, then issue the DSMSERV RESTORE DB utility.

Note: Roll-forward recovery does not apply if all recovery log volumes are lost. However, with the server running in roll-forward mode, you can still perform point-in-time recovery in such a case.

Restoring Storage Pools

You can recreate files in a primary storage pool by using duplicate copies in copy storage pools. The files must have been copied to the copy storage pools by using the BACKUP STGPOOL command.

Task	Required Privilege Class
Restoring storage pools	System, unrestricted storage, or restricted storage

The RESTORE STGPOOL command restores specified primary storage pools that have files with the following problems:

- The primary copy of the file has been identified as having read errors during a previous operation. Files with read errors are marked as damaged.
- The primary copy of the file resides on a volume that has an access mode of DESTROYED. For how the access mode of a volume changes to the DESTROYED access mode, see "How Restore Processing Works" on page 640.

When you restore a storage pool, be prepared to provide the following information:

Primary storage pool
Specifies the name of the primary storage pool that is being restored.

Copy storage pool
Specifies the name of the copy storage pool from which the files are to be restored. This information is optional. If you do not specify a copy storage pool, the server restores the files from any copy storage pool where it can find them.

New storage pool
Specifies the name of the new primary storage pool to which to restore the files. This information is optional. If you do not specify a new storage pool, the server restores the files to the original primary storage pool.

Maximum number of processes
Specifies the maximum number of parallel processes that are used for restoring files.

Preview
Specifies whether you want to preview the restore operation without actually restoring data.

See "Correcting Damaged Files" on page 678 and "Backup and Recovery Scenarios" on page 680 for examples of using the RESTORE STGPOOL command.

What Happens When a Storage Pool Is Restored

When you restore a storage pool, Tivoli Storage Manager determines which files are in that storage pool. Using file copies from a copy storage pool, Tivoli Storage Manager restores the files that were in the storage pool to the same or a different storage pool.

Cached Files: Cached copies of files in a disk storage pool are never restored. References to any cached files that have been identified as having read errors or cached files that reside on a *destroyed* volume will be removed from the database during restore processing.

The RESTORE STGPOOL command with the PREVIEW=YES parameter can be used to identify volumes that contain damaged primary files. During restore processing, a message is issued for every volume in the restored storage pool that contains damaged, noncached files. To identify the specific files that are damaged on these volumes, use the QUERY CONTENT command.

After the files are restored, the old references to these files in the primary storage pool are deleted from the database. This means that Tivoli Storage Manager now locates these files on the volumes to which they were restored, rather than on the volumes on which they were previously stored. If a destroyed volume becomes

empty because all files have been restored to other locations, the destroyed volume is automatically deleted from the database.

The RESTORE STGPOOL command generates a background process that can be canceled with the CANCEL PROCESS command. If a RESTORE STGPOOL background process is canceled, some files may have already been restored prior to the cancellation. To display information about background processes, use the QUERY PROCESS command.

The RESTORE STGPOOL command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

Restoring Files to a Storage Pool with Collocation Enabled

When restoring to a primary storage pool that has collocation enabled, the server restores files by collocation group, by client node or by client file space. This process preserves the collocation of client files. However, if the copy storage pool being used to restore files does not have collocation enabled, restore processing can be slow.

If you need to use a copy storage pool that is not collocated to restore files to a primary storage pool that is collocated, you can improve performance by:

1. Restoring the files first to a random access storage pool (on disk).
2. Allowing or forcing 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.

When a Storage Pool Restoration Is Incomplete

The restoration of a storage pool volume may be incomplete. Use the QUERY CONTENT command to get more information on the remaining files on volumes for which restoration was incomplete. The restoration may be incomplete for one or more of the following reasons:

- Either files were never backed up, or the backup copies were marked as damaged.
- A copy storage pool was specified on the RESTORE command, but files were backed up to a different copy storage pool. If you suspect this is a problem, use the RESTORE command again without specifying a copy storage pool from which to restore files. The PREVIEW option can be used on the second RESTORE command, if you do not actually want to restore files.
- Volumes in the copy storage pool needed to perform the restore operation are offsite or unavailable. Check the activity log for messages that occurred during restore processing.
- Backup file copies in copy storage pools were moved or deleted by other processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool volumes while restore processing is in progress:
  - MOVE DATA
  - DELETE VOLUME (DISCARDDATA=YES)
  - AUDIT VOLUME (FIX=YES)
- You can prevent reclamation processing for your copy storage pools by setting the RECLAIM parameter to 100 with the UPDATE STGPOOL command.

Restoring Your Server Using Mirrored Volumes

If a mirrored volume fails due to media failure, you can restore the volume by taking the following steps:

1. View the status of the database and recovery log volumes by using QUERY DBVOLUME or QUERY LOGVOLUME commands.
2. If necessary, place the failing volume offline by using DELETE DBVOLUME or DELETE LOGVOLUME commands. The server usually does this automatically.
3. Fix the failing physical device.
4. Allocate space to be used for a new volume by using the DSMFMT utility.
5. Bring the volume online by using DEFINE DBCOPY or DEFINE LOGCOPY commands.

After a database or recovery log volume copy is defined, the server synchronizes the volume copy with its associated database or recovery log volume.

Restoring Storage Pool Volumes

You can recreate files in primary storage pool volumes by using copies in a copy storage pool.

Task	Required Privilege Class
Restore volumes in any storage pool for which they have authority	System, unrestricted storage, or restricted storage

Use the RESTORE VOLUME command to restore all files that are stored in the same primary storage pool and that were previously backed up to copy storage pools. When you use the RESTORE VOLUME command, be prepared to supply some or all of the following information:

Volume name
Specifies the name of the volume in the primary storage pool for which to restore files.

Tip: To restore more than one volume in the same primary storage pool, issue this command once and specify a list of volumes to be restored. When you specify more than one volume, Tivoli Storage Manager attempts to minimize volume mounts for the copy storage pool.

Copy storage pool name
Specifies the name of the copy pool from which the files are to be restored. This information is optional. If you do not specify a particular copy storage pool, the files are restored from any copy storage pool where it can find them.

New storage pool
Specifies the name of the new primary storage pool to which to restore the files. This information is optional. If you do not specify a new storage pool, the files are restored to the original primary storage pool.

Maximum number of processes
Specifies the maximum number of parallel processes that are used for restoring files.

Preview
Specifies whether you want to preview the restore operation without actually restoring data.

See "Recovering a Lost or Damaged Storage Pool Volume" on page 683 for an example of using the RESTORE VOLUME command.

What Happens When a Volume Is Restored

When you restore a volume, the server obtains a copy of each file that was on the volume from a copy storage pool, and then stores the files on a different volume.

Cached Files: Cached copies of files in a disk storage pool are never restored. References to any cached files that reside on a volume that is being restored are removed from the database during restore processing.

After files are restored, the old references to these files in the primary storage pool are deleted from the database. Tivoli Storage Manager now locates these files on the volumes to which they were restored, rather than on the volume on which they were previously stored.

The RESTORE VOLUME command changes the access mode of the volumes being restored to *destroyed*. When the restoration is complete (when all files on the volume are restored to other locations), the destroyed volume is empty and is then automatically deleted from the database.

The RESTORE VOLUME command generates a background process that can be canceled with the CANCEL PROCESS command. If a RESTORE VOLUME background process is canceled, some files may have already been restored prior to the cancellation. To display information on background processes, use the QUERY PROCESS command.

The RESTORE VOLUME command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

When a Volume Restoration Is Incomplete

The restoration of a volume may be incomplete. Use the QUERY CONTENT command to get more information on the remaining files on volumes for which restoration was incomplete. The restoration may be incomplete for one or more of the following reasons:

- Files were either never backed up or the backup copies are marked as damaged.
- A copy storage pool was specified on the RESTORE command, but files were backed up to a different copy storage pool. If you suspect this is a problem, use the RESTORE command again without specifying a copy storage pool from which to restore files. The PREVIEW option can be used on the second RESTORE command, if you do not actually want to restore files.
- Volumes in the copy storage pool needed to perform the restore operation are offsite or unavailable. Check the activity log for messages that occurred during restore processing.
- Backup file copies in copy storage pools were moved or deleted by other processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool volumes while restore processing is in progress:
  - MOVE DATA
  - DELETE VOLUME (DISCARDDATA=YES)

– AUDIT VOLUME (FIX=YES)

You can prevent reclamation processing for your copy storage pools by setting the RECLAIM parameter to 100 with the UPDATE STGPOOL command.

Auditing a Storage Pool Volume

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

The server database contains information about files on storage pool volumes. If there are inconsistencies between the information in the database about files and the files actually stored in a storage pool volume, users may be unable to access their files.

To ensure that all files are accessible on volumes in a storage pool, audit any volumes you suspect may have problems by using the AUDIT VOLUME command. You have the option of auditing multiple volumes using a time range criteria, or auditing all volumes in a storage pool.

You should audit a volume when the following conditions are true:

- The volume is damaged.
- The volume has not been accessed for a long period of time, for example, after six months
- A read or write error occurs while accessing the volume
- The database has been restored to an earlier point-in-time, and the volume is either a disk volume or a volume that was identified as being reused or deleted since the database backup

If a storage pool has data validation enabled, run an audit for the volumes in the storage pool to have the server validate the data.

What Happens When You Audit Storage Pool Volumes

When you audit a volume, a background process is started. During the auditing process, the server:

- 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.
- Records results of the audit in the activity log.

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:

- The type of storage pool to which the volume is assigned
- The FIX option of the AUDIT VOLUME command
- The location of file copies (whether a copy of the file exists in a copy storage pool)

To display the results of a volume audit after it has completed, use the QUERY ACTLOG command. See "Requesting Information from the Activity Log" on page 524.

Volumes in a Primary Storage Pool

For a volume in a primary storage pool, the values for the FIX parameter on an AUDIT VOLUME command 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:

- 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.

Volumes in a Copy Storage Pool

For volumes in a copy storage pool, the values for the FIX parameter on an AUDIT VOLUME command 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.

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. This is helpful if you have introduced new hardware devices. The validation assures that the data is not corrupt as it moves through the hardware, and then is written to the volume in the storage pool. You can use the DEFINE STGPOOL or UPDATE STGPOOL commands to enable data validation for storage pools.

When you enable data validation for an existing storage pool, the server validates data that is written from that time forward. The server does not validate existing data which was written to the storage pool before data validation was enabled.

When data validation is enabled for storage pools, the server generates a cyclic redundancy check (CRC) value and stores it with the data when it is written to the storage pool. The server validates the data when it audits the volume, by generating a cyclic redundancy check and comparing this value with the CRC value stored with the data. If the CRC values do not match, then the server processes the volume in the same manner as a standard audit volume operation. This process can depend on the following:

- The type of storage pool to which the volume is assigned
- The FIX option of the AUDIT VOLUME command
- The location of file copies (whether a copy of the file exists in a copy storage pool)

See "Volumes in a Primary Storage Pool" on page 672 and "Volumes in a Copy Storage Pool" on page 672 for details on how the server handles inconsistencies detected during an audit volume process. Check the activity log for details about the audit operation.

The server removes the CRC values before it returns the data to the client node.

Choosing Where to Enable Data Validation

Data validation is available for nodes and storage pools. The forms of validation are independent of each other. Figure 100 on page 674 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 100 on page 674 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 100. Data Transfer Eligible for Data Validation

Table 39 provides information that relates to Figure 100. This information explains the type of data being transferred and the appropriate command to issue.

Table 39. Setting Data Validation

Numbers in Figure 100	Where to Set Data Validation	Type of Data Transferred	Command	Command Parameter
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 DATAONLY
3	Server definition (storage sgent only)	Metadata	DEFINE SERVER UPDATE SERVER	VALIDATEPROTOCOL=ALL
4	Storage pool definition issued on the Tivoli Storage Manager server	File Data	DEFINE STGPOOL UPDATE STGPOOL	CRCDATA=YES[1]
5	Storage pool definition issued on the Tivoli Storage Manager server	File Data	DEFINE STGPOOL UPDATE STGPOOL	CRCDATA=YES[1]

[1] This parameter is not available for primary sequential access storage pools that use the following data formats: NETAPPDUMP, CELERRADUMP, NDMPDUMP.

Note: The storage agent reads the VALIDATEPROTOCOL setting for the client from the Tivoli Storage Manager server.

Figure 101 on page 675 is similar to the previous figure, however note that the top section encompassing 1, 2, and 3 is shaded. All three of these data validations are related to the VALIDATEPROTOCOL parameter. What is significant about this validation is that it is active only during the client session. After

validation, the client and server discard the CRC values generated in the current session. This is In contrast to storage pool validation, 4 and 5, which is always active as long as the storage pool CRCDATA setting is equal to YES.

The validation of data transfer between the storage pool and the storage agent 4 is managed by the storage pool CRCDATA setting defined by the Tivoli Storage Manager server. Even though the flow of data is between the storage agent and the storage pool, data validation is determined by the storage pool definition. Therefore, if you always want your storage pool data validated, set your primary storage pool CRCDATA setting to YES.

Figure 101. Protocol Data Validation versus Storage Pool Data Validation

If the network is unstable, you may decide to only enable data validation for nodes. Tivoli Storage Manager generates a cyclic redundancy check when the data is sent over the network to the server. Certain nodes may have more critical data than others and may require the assurance of data validation. When you identify the nodes that require data validation, you can choose to have only the user's data validated or all the data validated. Tivoli Storage Manager validates both the file data and the file metadata when you choose to validate all data. See "Validating a Node's Data During a Client Session" on page 412.

When you enable data validation for a server-to-server exchange or between a storage agent and server, the server must validate all data. You can enable data validation by using the DEFINE SERVER or UPDATE SERVER command. For a server-to-server exchange, see "Using Virtual Volumes to Store Data on Another Server" on page 601 for more information. For data that is exchanged between a storage agent and the server, refer to the *IBM Tivoli Storage Manager Storage Agent User's Guide* for the storage agent's operating system.

If the network is fairly stable but your site is perhaps using new hardware devices, you may decide to only enable data validation for storage pools. When the server sends data to the storage pool, the server generates cyclic redundancy checking, and stores the CRC value with the data. The server validates the CRC value when the server audits the volume. Later, you may decide that data validation for storage pools is no longer required after the devices prove to be stable. Refer to "Auditing a Storage Pool Volume" on page 671 for more information on data validation for storage pools.

Performance Considerations

Consider the impact on performance when you decide whether data validation is necessary for storage pools. Data validation impacts performance because the server requires additional CPU overhead to calculate and compare CRC values. 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.

Managing Storage Pool Data Validation

The AUDIT VOLUME command has additional parameters that allow you to specify an audit for data written to volumes within a range of days, or to run an audit for a given storage pool.

You can manage when the validation of data in storage pools occurs by scheduling the audit volume operation. You can choose a method suitable to your environment, for example:

- The administrator can select volumes at random to audit. A random selection does not require significant resources or cause much contention for server resources but can provide assurance that the data is valid.
- Schedule a daily audit of all volumes written in the last day. This method validates data written to a given storage pool on a daily basis.
- Audit volumes in storage pools only for client nodes that are considered to be critical users.

Auditing a Volume in a Disk Storage Pool

To audit disk volume /dev/vol1 and have only summary messages sent to the activity log and server console, enter:

```
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, enter:

```
query process
```

The following figure displays 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
```

To display the results of a volume audit after it has completed, you can issue the QUERY ACTLOG command.

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 begins the audit process with the first volume on which the first file resides. For example, Figure 102 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 102. Tape Volumes with Files A, B, C, D, and E

If you request that the server audit volume VOL3, the server first accesses volume VOL2, because File D begins at VOL2. When volume VOL2 is accessed, the server *only* audits File D. It does not audit the other files on this volume.

Because File D spans multiple volumes, the server accesses volumes VOL2, VOL3, VOL4, and VOL5 to ensure that there are no inconsistencies between the database and the storage pool volumes.

For volumes that require manual mount and dismount operations, the audit process can require significant manual intervention.

Auditing a Single Volume in a Sequential Access Storage Pool

To audit a single volume in a sequential storage pool, request that the server skip any files that span multiple volumes. This option is useful when the volume you want to audit contains part of a file, the rest of which resides on a different, damaged volume.

For example, to audit only volume VOL5 in the example in Figure 102 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. 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=032002 todate=032202
```

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.

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

Auditing Volumes in a Specific Storage Pool

You can limit the audit to volumes in a specified storage pool. For example, you can audit the volumes in storage pool BKPOOL1 by issuing the following command:

```
audit volume stgpool=bkppool1
```

Defining a Schedule to Audit Volumes on a Regular Basis

You can define administrative schedules to have the server audit volumes on a regular basis. The following example can be used if your critical users store data in storage pool STPOOL3 and you want all volumes in the storage pool audited every 2 days.

```
define schedule crcstg1 type=administrative
cmd='audit volume stgpool=STPOOL3' active=yes starttime=21:00 period=2
```

This command will audit all volumes in the STPOOL3 storage pool every two days. The audit operation will begin at 9:00 p.m.

Correcting Damaged Files

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, 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, the server sends the user the undamaged copy.

Files that are marked as damaged cannot be:

- Restored, retrieved, or recalled
- Moved by migration, reclamation, or the MOVE DATA command
- Backed up during a BACKUP STGPOOL operation if the primary file is damaged
- Restored during a RESTORE STGPOOL or RESTORE VOLUME operation if the backup copy in a copy storage pool is damaged
- Migrated or reclaimed during migration and reclamation

Maintaining the Integrity of Files

To maintain the data integrity of user files, you can:

1. Detect damaged files before the users do.

   The AUDIT VOLUME command marks a file as damaged if a read error is detected for the file. If an undamaged copy is in an onsite copy storage pool, it is used to provide client access to the file. See “Data Validation During Audit Volume Processing” on page 673.
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, the primary file can be corrected using the RESTORE VOLUME or RESTORE STGPOOL command. For an example, see "Restoring Damaged Files."

4. Regularly run commands to identify files that are marked as damaged:
   - The RESTORE STGPOOL command displays the name of each volume in the restored storage pool that contains one or more damaged primary files. Use this command with the preview option to identify primary volumes with damaged files without actually performing the restore operation.
   - The QUERY CONTENT command with the DAMAGED option lets you display damaged files on a specific volume.

For an example of how to use these commands, see "Restoring Damaged Files."

Restoring Damaged Files

If you use copy storage pools, you can restore damaged client files. You can also check storage pools for damaged files and restore the files. This section explains how to restore damaged files based on the scenario in "Example: Simple Hierarchy with One Copy Storage Pool" on page 650.

If a client tries to access a file stored in TAPEPOOL and a read error occurs, the file in TAPEPOOL is automatically marked as damaged. Future accesses to the file automatically use the copy in COPYPOOL as long as the copy in TAPEPOOL is marked as damaged.

To restore any *damaged* files in TAPEPOOL, you can define a schedule that issues the following command periodically:

```
restore stgpool tapepool
```

You can check for and replace any files that develop data-integrity problems in TAPEPOOL or in COPYPOOL. For example, every three months, query the volumes in TAPEPOOL and COPYPOOL by entering the following commands:

```
query volume stgpool=tapepool
query volume stgpool=copypool
```

Then issue the following command for each volume in TAPEPOOL and COPYPOOL:

```
audit volume <volname> fix=yes
```

If a read error occurs on a file in TAPEPOOL, that file is marked *damaged* and an error message is produced. If a read error occurs on file in COPYPOOL, that file is deleted and a message is produced.

Restore *damaged* primary files by entering:

```
restore stgpool tapepool
```

Finally, create new copies in COPYPOOL by entering:

```
backup stgpool tapepool copypool
```

Backup and Recovery Scenarios

This section presents scenarios for protecting and recovering a Tivoli Storage Manager server. You can modify the procedures to meet your needs.

These scenarios assume a storage hierarchy consisting of:

- The default random access storage pools (BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL)
- TAPEPOOL, a tape storage pool

Protecting Your Database and Storage Pools

A company's standard procedures include the following:

- Perform reclamation of its copy storage pool, once a week. 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.
- Back up its storage pools every night.
- Perform a full backup of the database once a week and incremental backups on the other days.
- Ship the database and copy storage pool volumes to an offsite location every day.

To protect client data, the administrator does the following:

1. Creates 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. Performs the first backup of the primary storage pools.

   Note: The first backup of a primary storage pool is a full backup and, depending on the size of the storage pool, could take a long time.
3. Defines schedules for the following daily operations:
   a. Incremental backups of the primary storage pools each night by issuing:

      ```
      backup stgpool backuppool disaster-recovery maxprocess=2
      backup stgpool archivepool disaster-recovery maxprocess=2
      backup stgpool spacemgpool disaster-recovery maxprocess=2
      backup stgpool tapepool disaster-recovery maxprocess=2
      ```

      These commands use multiple, parallel processes to perform an incremental backup of each primary storage pool to the copy pool. Only those files for which a copy does not already exist in the copy pool are backed up.

      Note: Migration should be turned off during the rest of the day. You could add a schedule to migrate from disk to tape at this point. In this way, the backups are done while the files are still on disk.

b. Change the access mode to OFFSITE for volumes that have read-write or read-only access, are onsite, and are at least partially filled. This is done by entering:

```
update volume * access=offsite location='vault site info'
wherestgpool=disaster-recovery whereaccess=readwrite,readonly
wherestatus=filling,full
```

c. Back up the database by entering:

```
backup db type=incremental devclass=tapeclass scratch=yes
```

4. Does the following operations nightly after the scheduled operations have completed:
   a. Backs up the volume history and device configuration files. If they have changed, back up the server options files and the database and recovery log setup information.
   b. Moves the 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. Identifies offsite volumes that should be returned onsite by using the QUERY VOLUME command:

      ```
      query volume stgpool=disaster-recovery access=offsite status=empty
      ```

      These volumes, which have become empty through expiration, reclamation, and file space deletion, have waited the delay time specified by the REUSEDELAY parameter. The administrator periodically returns outdated backup database volumes. These volumes are displayed with the QUERY VOLHISTORY command and can be released for reuse with the DELETE VOLHISTORY command.
5. Brings the volumes identified in step 4c onsite and updates their access to read-write.

Recovering to a Point-in-Time from a Disaster

In this scenario, the processor on which Tivoli Storage Manager resides, the database, and all onsite storage pool volumes are destroyed by fire. An administrator restores the server to the point-in-time of the last backup. You can use either full and incremental backups or snapshot database backups to restore a database to a point-in-time.

Do the following:

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. For example, to initialize the database and recovery log, enter:

   ```
   dsmserv format 1 log1 1 dbvol1
   ```

2. Move the latest backup and all of the DISASTER-RECOVERY volumes onsite from the offsite location.

   Note: Do not change the access mode of these volumes until after you have completed step 7 on page 682.
3. If a current, undamaged volume history file exists, save it.
4. Restore the volume history and device configuration files, the server options and the database and recovery log setup. For example, the recovery site might require different device class, library, and drive definitions. For more information, see “Updating the Device Configuration File” on page 659.

5. Restore the database from the latest backup level by issuing the DSMSERV RESTORE DB utility (see "Recovering Your Server Using Database and Storage Pool Backups" on page 661).
6. Change the access mode of all the existing primary storage pool volumes in the damaged storage pools to DESTROYED by entering:
```
update volume * access=destroyed wherestgpool=backuppool
update volume * access=destroyed wherestgpool=archivepool
update volume * access=destroyed wherestgpool=spacemgpool
update volume * access=destroyed wherestgpool=tapepool
```
7. Issue the QUERY VOLUME command to identify any volumes in the DISASTER-RECOVERY storage pool that were onsite at the time of the disaster. Any volumes that were onsite would have been destroyed in the disaster and could not be used for restore processing. Delete each of these volumes from the database by using the DELETE VOLUME command with the 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 by entering:
```
update volume * access=readwrite wherestgpool=disaster-recovery
```

   Note: At this point, clients can access files. If a client tries to access a file that was stored on a destroyed volume, the retrieval request goes to the copy storage pool. In this way, clients can access their files without waiting for the primary storage pool to be restored. When you update volumes brought from offsite to change their access, you greatly speed recovery time.
9. Define new volumes in the primary storage pool so the files on the damaged volumes can be restored to the new volumes. The new volumes also let clients backup, archive, or migrate files to the server. You do not need to perform this step if you use only scratch volumes in the storage pool.
10. Restore files in the primary storage pool from the copies located in the DISASTER-RECOVERY pool by entering:
```
restore stgpool backuppool maxprocess=2
restore stgpool archivepool maxprocess=2
restore stgpool spacemgpool maxprocess=2
restore stgpool tapepool maxprocess=2
```
    These commands use multiple parallel processes to restore files to primary storage pools. After all the files have been restored for a destroyed volume, that volume is automatically deleted from the database. See "When a Storage Pool Restoration Is Incomplete" on page 668 for what to do if one or more volumes cannot be fully restored.
11. 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.

Point-in-Time Restores In a Shared Library Environment

A point-in-time restore for a library manager server or a library client server requires additional steps to ensure the consistency of the volume inventories of the affected servers. This section describes the procedures for the two possible scenarios.

Point-in-Time Restore of a Library Manager Server: A point-in-time restore of a library manager server could create inconsistencies between the volume inventories

of the library manager and library client servers. The restore removes all library client server transactions that occurred after the point in time from the volume inventory of the library manager server. The volume inventory of the library client server, however, still contains those transactions. New transactions could then be written to these volumes, resulting in a loss of client data. To prevent this problem, do the following after the restore:

1. Halt further transactions on the library manager server: Disable all schedules, migration and reclamations on the library client and library manager servers.
2. Audit all libraries on all library client servers. The audits will re-enter those volume transactions that were removed by the restore on the library manager server. You should audit the library clients from the oldest to the newest servers. Use the volume history file from the library client and library manager servers to resolve any conflicts.
3. Delete the volumes from the library clients that do not own the volumes.
4. Resume transactions by enabling all schedules, migration, and reclamations on the library client and library manager servers.

Point-in-Time Restore of a Library Client Server: A point-in-time restore of a library client server could cause volumes to be removed from the volume inventory of a library client server and later overwritten. If a library client server acquired scratch volumes after the point-in-time to which the server is restored, these volumes would be set to private in the volume inventories of the library client and library manager servers. After the restore, the volume inventory of the library client server can be regressed to a point-in-time before the volumes were acquired, thus removing them from the inventory. These volumes would still exist in the volume inventory of the library manager server as private volumes owned by the client.

The restored volume inventory of the library client server and the volume inventory of the library manager server would be inconsistent. The volume inventory of the library client server must be synchronized with the volume inventory of the library manager server in order to return those volumes to scratch and enable them to be overwritten. To synchronize the inventories, do the following:

1. Audit the library on the library client server to synchronize the volume inventories of the library client and library manager servers.
2. To resolve any remaining volume ownership concerns, refer to the volume history and issue the UPDATE VOLUME command as needed.

Recovering a Lost or Damaged Storage Pool Volume

If a company makes the preparations described in “Protecting Your Database and Storage Pools” on page 680, it can recover from a media loss. In the following scenario, an operator inadvertently destroys a tape volume (DSM087) belonging to the TAPEPOOL storage pool. An administrator performs the following actions to recover the data stored on the destroyed volume by using the offsite copy storage pool:

1. Determine the copy pool volumes that contain the backup copies of the files that were stored on the volume that was destroyed by entering:

   ```
   restore volume dsm087 preview=yes
   ```

   This command produces a list of offsite volumes that contain the backed up copies of the files that were on tape volume DSM087.
2. Set the access mode of the copy volumes identified as UNAVAILABLE to prevent reclamation.

Note: This precaution prevents the movement of files stored on these volumes until volume DSM087 is restored.

3. Bring the identified volumes to the onsite location and set their access mode to READONLY to prevent accidental writes. If these offsite volumes are being used in an automated library, the volumes must be checked into the library when they are brought back onsite.
4. Restore the destroyed files by entering:

   ```
   restore volume dsm087
   ```

   This command sets the access mode of DSM087 to DESTROYED and attempts to restore all the files that were stored on volume DSM087. The files are not actually restored to volume DSM087, but to another volume in the TAPEPOOL storage pool. All references to the files on DSM087 are deleted from the database and the volume itself is deleted from the database.
5. Set the access mode of the volumes used to restore DSM087 to OFFSITE using the UPDATE VOLUME command.
6. Set the access mode of the restored volumes, that are now onsite, to READWRITE.
7. Return the volumes to the offsite location. If the offsite volumes used for the restoration were checked into an automated library, these volumes must be checked out of the automated library when the restoration process is complete.

Restoring a Library Manager Database

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. If the library manager's database becomes corrupted, it may be restored following these steps:

1. Rename and save a copy of the volume history file if it exists.

   After the database is restored, any volume history information pointed to by the server options is lost. You will need this information to identify the volumes to be audited.
2. Put the device configuration file and the server options file in the server working directory.

   If the device configuration file is unavailable, recreate it manually. For information about recreating a device configuration file, see "Recreating a Device Configuration File" on page 659.
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 volumes are present.

   If the original database or recovery log volumes were lost, issue the DSMSERV FORMAT utility to initialize the database and recovery log.
5. Issue the DSMSERV RESTORE DB utility.
6. Start the library manager.
7. Issue an AUDIT LIBRARY command from each library client for each shared library.
8. Create a list from the old volume history information (generated by the QUERY VOLHISTORY command) that shows all of the volumes that were reused (STGREUSE), added (STGNEW), and deleted (STGDELETE) since the original backup. Use this list to perform the rest of this procedure.

9. Audit all disk volumes, all reused volumes, and any deleted volumes located by the AUDIT VOLUME command using the FIX=YES parameter.
10. Issue the RESTORE STGPOOL command to restore those files detected as damaged by the audit. Include the FIX=YES parameter on the AUDIT VOLUME command to delete database entries for files not found in the copy storage pool.
11. 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.
12. Redefine any storage pool volumes that were added since the database backup.

Restoring a Library Client Database

In a Tivoli Storage Manager shared library environment, the servers that share a library and rely on a library manager to coordinate and manage the library's usage are known as library clients. Each library client maintains a database of volume usage and volume history. If the library client's database becomes corrupted, it may be restored following these steps:

1. Rename and save a copy of the volume history file if it exists.

   After the database is restored, any volume history information pointed to by the server options is lost. You will need this information to identify the volumes to be audited.
2. Put the device configuration file and the server options file in the server working directory.

   If the device configuration file is unavailable, you may recreate it manually. For information about recreating a device configuration file, see "Recreating a Device Configuration File" on page 659.
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 volumes are present.

   If the original database or recovery log volumes were lost, issue the DSMSERV FORMAT utility to initialize the database and recovery log.
5. Issue the DSMSERV RESTORE DB utility.
6. Create a list from the old volume history information (generated by the QUERY VOLHISTORY command) that shows all of the volumes that were reused (STGREUSE), added (STGNEW), and deleted (STGDELETE) since the original backup. Use this list to perform the rest of this procedure.
7. Audit all disk volumes, all reused volumes, and any deleted volumes located by the AUDIT VOLUME command using the FIX=YES parameter.
8. Issue the RESTORE STGPOOL command to restore those files detected as damaged by the audit. Include the FIX=YES parameter on the AUDIT VOLUME command to delete database entries for files not found in the copy storage pool.
9. Mark as destroyed any volumes that cannot be located, and recover those volumes from copy storage pool backups. If no backups are available, delete the volumes from the database by using the DELETE VOLUME command with the DISCARDDATA=YES parameter.
10. Issue the AUDIT LIBRARY command for all shared libraries on this library client.

11. Redefine any storage pool volumes that were added since the database backup.

Chapter 24. Using Disaster Recovery Manager

You can use the disaster recovery manager (DRM) function to do any one or all the following:

- Prepare a disaster recovery plan that can help you to recover your applications in the case of a disaster. You can recover at an alternate site, on replacement computer hardware, and with people who are not familiar with the applications.
- Manage your offsite recovery media.
- Store your client recovery information.

You can also use the disaster recovery plan for audits to certify the recoverability of the server.

Note: DRM is function within the product, Tivoli Storage Manager Extended Edition. For more information, see "Registering Licensed Features" on page 458.

Before using this chapter, you should be familiar with Chapter 23, "Protecting and Recovering Your Server," on page 639.

This chapter contains the following sections:

Tasks:
"Querying Defaults for the Disaster Recovery Plan File" on page 688
"Specifying Recovery Instructions for Your Site" on page 692
"Specifying Information About Your Server and Client Node Machines" on page 693
"Specifying Recovery Media for Client Machines" on page 695
"Creating and Storing the Disaster Recovery Plan" on page 696
"Managing Disaster Recovery Plan Files Stored on Target Servers" on page 697
"Moving Backup Media" on page 699
"Summary of Disaster Recovery Manager Daily Tasks" on page 704
"Staying Prepared for a Disaster" on page 705
"Recovering From a Disaster" on page 706
Disaster Recovery Reference:
"Disaster Recovery Manager Checklist" on page 714
"The Disaster Recovery Plan File" on page 717

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the Administration Center. For more information about using the Administration Center, see "Managing Servers with the New Administration Center" on page 36.

© Copyright IBM Corp. 1993, 2005

Task	Required Privilege Class
All DRM tasks unless otherwise noted.	System

To recover from a disaster, you must know the location of offsite recovery media. DRM helps you to determine which volumes to move offsite and back onsite and, tracks the location of the volumes.

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. However, you can override these default settings. 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*
       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

The following table describes how to set defaults for the disaster recovery plan file.

Table 40. 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 40. 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` **Notes:** 1. To remove any specified copy storage pool names, and thus select all copy storage pools, specify a null string (”) in SET DRMCOPYSTGPOOL. 2. 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
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 as */u/recovery/plans/rpp*, 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

Table 40. Defaults for the Disaster Recovery Plan File (continued)

Process	Default

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.MODERECOVERY.DRMODE and RECOVERY.NMODE 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 as */u/server/recoveryplans/*, enter:
	`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

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

The following table describes how to set defaults for offsite recovery media management.

Table 41. 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

Table 41. Defaults for Offsite Recovery Media Management (continued)

Process	Default
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, to set the file name as */drm/orm/exec.cmds* enter: `set drmcmdfilename /drm/orm/exec.cmds` **To override the default:** CMDFILENAME parameter on MOVE DRMEDIA or QUERY DRMEDIA
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`

Table 41. Defaults for Offsite Recovery Media Management (continued)

Process	Default
Whether to process backup volumes of the FILE device type	At installation, MOVE DRMEDIA and QUERY DRMEDIA will not process backup volumes that are associated with a device type of FILE. **The default at installation:** Backup volumes of the FILE device type are not processed **To change the default:** SET DRMFILEPROCESS To allow processing, enter: `set drmfileprocess yes`
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 disaster recovery plan includes instructions that you create. Enter your instructions in flat files that have the following names:

- *prefix*.RECOVERY.INSTRUCTIONS.GENERAL
- *prefix*.RECOVERY.INSTRUCTIONS.OFFSITE
- *prefix*.RECOVERY.INSTRUCTIONS.INSTALL
- *prefix*.RECOVERY.INSTRUCTIONS.DATABASE
- *prefix*.RECOVERY.INSTRUCTIONS.STGPOOL

Note: The files created for the recovery instructions must be physical sequential files.

RECOVERY.INSTRUCTIONS.GENERAL

Include information such as administrator names, telephone numbers, and location of passwords. For example:

```
Recovery Instructions for Tivoli Storage Manager Server ACMESRV on system ZEUS
Joe Smith (wk 002-000-1111 hm 002-003-0000):  primary system programmer
Sally Doe (wk 002-000-1112 hm 002-005-0000):  primary recovery administrator
Jane Smith (wk 002-000-1113 hm 002-004-0000): responsible manager

Security Considerations:
Joe Smith has the password for the Admin ID ACMEADM. If Joe is unavailable,
you need to either issue SET AUTHENTICATION OFF or define a new
administrative user ID at the replacement Tivoli Storage Manager server console.
```

RECOVERY.INSTRUCTIONS.OFFSITE

Include information such as the offsite vault location, courier's 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.
Our courier's name 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 and copy storage pool volumes required. Fred
is committed to returning these volumes to us in less than 12 hours.
```

RECOVERY.INSTRUCTIONS.INSTALL
Include information about server installation and the location of installation volumes. For example:

```
You will need to reinstall the Tivoli Storage Manager server and administrative
client after installing the Sun Solaris operating system.
The install volume for the Tivoli Storage Manager server is INS001. If that is
lost, you will need to contact Copy4You Software, at 1-800-000-0000, and obtain
a new copy. Another possibility is the local IBM Branch office at
555-7777.
```

RECOVERY.INSTRUCTIONS.DATABASE
Include information about how to recover the database and about how much hardware space requirements. For example:

```
You will need to find replacement disk space for the server database. We
have an agreement with Joe Replace that in the event of a disaster, he
will provide us with disk space.
```

RECOVERY.INSTRUCTIONS.STGPOOL
Include information on primary storage pool recovery instructions. For example:

```
Do not worry about the archive storage pools during this disaster recovery.
Focus on migration and backup storage pools.
The most important storage pool is XYZZZZ.
```

Specifying Information About Your Server and Client Node Machines

You need information about your server machine to rebuild its replacement. You also need information about client node machines to rebuild or restore them. Follow this procedure to specify that information and store it in the server database:

Server Machine

1. Specify server machine information:

   Issue 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:

   ```
   define machine tsm1 adsmserver=yes priority=1
   ```

2. **Client Machines**Specify the client node location and business priority:

   Issue 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:

   Issue 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 709. 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. 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
     ```

Figure 103 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   {
       }
```

Figure 103. Example of Awk Script File to Insert Machine Characteristics

d. The *machchar.awk* script is then run from an AIX prompt as follows:

```
nawk -f machchar.awk -v machine=acctrcv clientinfo.txt > clientinfo.mac
```

e. To add the machine characteristics, start an administrative client and run the macro. For example:

```
> dsmadmc -id=xxx -pw=xxx macro clientinfo.mac
```

You can view your machine characteristics by issuing the QUERY MACHINE command with FORMAT=CHARACTERISTICS parameter.

f. To specify recovery instructions for your client machine, use this same awk script process but with the RECOVERYINSTRUCTIONS parameter.

Specifying Recovery Media for Client Machines

Follow these steps to specify the bootable media needed to reinitialize or reinstall an operating system on a client machine and to associate machines with media. You can also associate non-executable media such as application user guides with client machines.

1. Define the bootable media. For example, define the media named TELLERWRKSTNIMAGE which is for AIX Version 4.3, contains the required volumes named AIX001, AIX002, and AIX003, and is located in Building 21.

   ```
   define recoverymedia tellerwrkstnimage type=boot
     volumenames=aix001,aix002,aix003 product="AIX 4.3"
     location="Building 21"
   ```

   You should define the recovery media after a client machine configuration changes. For example, after you have installed a new level of AIX on a client machine and created a bootable image using **mksysb**, issue the DEFINE RECOVERYMEDIA command to define the new **mksysb** volumes.

   To query your recovery media definitions, issue the QUERY RECOVERYMEDIA command with the FORMAT=DETAILED parameter.

2. Associate one or more machines with recovery media. Use the association information to identify the boot media to use in the replacement machines. For example, to associate machine MACH255 with recovery media TELLERWRKSTNIMAGE, issue the following command:

   ```
   define recmedmachassociation tellerwrkstnimage mach255
   ```

3. When the boot media is moved offsite, update its location. For example, to update the location of boot media TELLERWRKSTNIMAGE to the offsite location IRONVAULT, issue the following command:

```
update recoverymedia tellerwrkstnimage location=ironvault
```

You can define media that contain softcopy manuals that you would need during recovery. For example, to define a CD-ROM containing the AIX 4.3 manuals that are on volume CD0001, enter:

```
define recoverymedia aix43manuals type=other volumes=cd0001
  description="AIX 4.3 Bookshelf"
```

Creating and Storing the Disaster Recovery Plan

You can create a disaster recovery plan file and store the file locally or on another server.

The recovery plan contains the following information:

- The recovery procedure
- A list of required database and storage pool backup 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 717.

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 diskettes, on NFS-mounted disk space that is located offsite, or on a remote server.

Before creating a disaster recovery plan, back up your storage pools then backup the database. See "Backing Up Storage Pools" on page 648 and "Backing Up the Database" on page 652 for details about these procedures.

If you manually send backup media offsite, see "Moving Backup Volumes Offsite" on page 701. If you use virtual volumes, see "Using Virtual Volumes to Store Data on Another Server" on page 601.

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 471).

Note: DRM creates a plan that assumes that the latest database full plus incremental series would be used to restore the database. However, you may want to use DBSNAPSHOT backups for disaster recovery and retain your full plus incremental backup series on site to recover from possible availability problems. In this case, you must specify the use of DBSNAPSHOT backups in the PREPARE command. For example:

```
prepare source=dbsnapshot
```

Storing the Disaster Recovery Plan Locally

When you create a recovery plan file but do not specify a device class, the file is stored locally in a file system. If you store the file locally, you can specify a storage location. For example, to store the recovery plan file locally in the */u/server/recoveryplans/* directory, enter:

```
prepare planprefix=/u/server/recoveryplans/
```

Recovery plan files that are stored locally are not automatically expired. You should periodically delete down-level recovery plan files manually.

DRM appends to the file name the date and time (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. 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

First, 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 602 for details). For example, assume a device class named TARGETCLASS is defined on the source server where you create the recovery plan file. Then to create the plan file, enter:

```
prepare devclass=targetclass
```

The recovery plan file is written as an object on the target server, and a volume history record is created on the source server. For more about recovery plan files that are stored on target servers, see "Displaying Information about Recovery Plan Files."

Managing Disaster Recovery Plan Files Stored on Target Servers

The following sections describe how you can view information about disaster recovery plan files 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):

- **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
```

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
```

- **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. For example,

- **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
```

Notes:

1. You cannot issue these commands from a server console.
2. An output delay can occur when the plan file is located on tape.

See "The Disaster Recovery Plan File" on page 717 for an example of the contents of a recovery plan file.

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.

For example, a recovery plan file named *marketing.20000831.045000* was created using the device class of TARGETCLASS and on a source server whose node name at the target server is BRANCH8. You want to restore the file and direct the output to *rpf.out*:

- **From the source server:** Enter,

```
query rpfcontent marketing.20000831.045000
  devclass=targetclass > rpf.out
```

- **From the target server:** Enter,

```
query rpfcontent marketing.20000831.045000
  nodename=branch8 > rpf.out
```

To display a list of recovery plan files, use the QUERY RPFILE command. See "Displaying Information about Recovery Plan Files" on page 697 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. 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
```

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 697.

Deleting Recovery Plan Files Manually

You can delete volume history records containing information about recovery plan file objects. When the records are deleted from the source server and the grace period is reached, the objects are deleted from the target server.

Note: The record for the latest recovery plan file is not deleted.

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:

```
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 will need database backup volumes and copy storage pool volumes. To prepare for a disaster, you will need to perform the following daily tasks:

1. Move new backup media offsite and update the database with their locations. See "Moving Backup Volumes Offsite" on page 701 for details.
2. Return expired or reclaimed backup media onsite and update the database with their locations. See "Moving Backup Volumes Onsite" on page 702 for details.

Task	Required Privilege Class
Send backup volumes offsite and back onsite	Unrestricted storage or operator

Offsite recovery media management does not process virtual volumes. To display all virtual copy storage pool and database backup volumes that have their backup objects on the remote target server, issue the following command:

```
query drmedia * wherestate=remote
```

The disaster recovery plan includes backup volume location information and can provide a list of offsite volumes required to restore a server.

The following diagram shows the typical life cycle of the recovery media:

Figure 104. Recovery Media Life Cycle

DRM assigns the following states to volumes. The location of a volume is known at each state.

MOUNTABLE
The volume contains valid data, and Tivoli Storage Manager can access it.

NOTMOUNTABLE
The volume contains valid data and is onsite, but Tivoli Storage Manager cannot access it.

COURIER
The volume contains valid data and is in transit to the vault.

VAULT
The volume contains valid data and is at the vault.

VAULTRETRIEVE
The volume, which is located at the offsite vault, no longer contains valid data and is to be returned to the site. For more information on reclamation

of offsite copy storage pool volumes, see "Reclamation of Offsite Volumes" on page 274. For information on expiration of database backup volumes, see step 1 on page 702.

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 and scratch copy storage pool volumes are deleted from the database. For private copy storage pool volumes, the access mode is updated to READWRITE.

Moving Backup Volumes Offsite

After you have 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. Do the following to identify the database backup and copy storage pool volumes and move them offsite:

1. Identify the copy storage pool and database backup volumes to be moved offsite:

   ```
   query drmedia * wherestate=mountable
   ```

   DRM displays information similar to the following:

   ```
   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 by issuing the following command:

   ```
   move drmedia * wherestate=mountable
   ```

   For all volumes in the MOUNTABLE state, DRM does the following:

   - Updates the volume state to NOTMOUNTABLE and the volume location according to the SET DRMNOTMOUNTABLENAME. If this command has not been issued, the default location is NOTMOUNTABLE.
   - For a copy storage pool volume, updates the access mode to unavailable.
   - For a volume in an automated library, checks the volume out of the library.

   Notes:

   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
      ```

      Next, 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.

   b. 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 has not yet been issued, the default location is COURIER. For more information, see "Specifying Defaults for Offsite Recovery Media Management" on page 690

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 has not yet been issued, the default location is VAULT. For more information, see "Specifying Defaults for Offsite Recovery Media Management" on page 690.

5. To display a list of volumes that contain valid data at the vault, issue the following command:

   ```
   query drmedia wherestate=vault
   ```

   DRM displays information similar to the following:

   ```
   Volume Name       State           Last Update          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 transition the volumes from NOTMOUNTABLE state to VAULT state, issue the following command:

   ```
   move drmedia * wherestate=notmountable tostate=vault
   ```

   For all volumes in the NOTMOUNTABLE state, DRM updates the volume state to VAULT and the volume location according to the SET DRMVAULTNAME command. If the SET command has not yet been issued, the default location is VAULT.

See "Staying Prepared for a Disaster" on page 705 for an example that demonstrates sending server backup volumes offsite using MOVE DRMEDIA and QUERY DRMEDIA commands.

Moving Backup Volumes Onsite

Use the following procedure to expire the non-virtual database backup volumes and return the volumes back onsite for reuse or disposal.

1. To specify the number of days before a database backup series is expired, issue the SET DRMDBBACKUPEXPIREDAYS command. To ensure that the database can be returned to an earlier level and database references to files in the copy

storage pool are still valid, specify the same value for the REUSEDELAY parameter in your copy storage pool definition.

The following example sets the number of days to 30.

```
set drmdbbackupexpiredays 30
```

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.

Note: 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 backup volume onsite for reuse or disposal. A copy storage pool backup volume can be moved onsite 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 onsite if the database backup series is EXPIRED according to the rules outlined in 1 on page 702. To determine which volumes to retrieve, issue the following command:

   ```
   query drmedia * wherestate=vaultretrieve
   ```

   The server dynamically determines which volumes can be moved back onsite. 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 following 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 690.

4. When the courier delivers the volumes, acknowledge that the courier has returned the volumes onsite, by issuing:

   ```
   move drmedia * wherestate=courierretrieve
   ```

   The server does the following for all volumes in the COURIERRETRIEVE state:

   - The volumes are now onsite and can be reused or disposed.
   - The database backup volumes are deleted from the volume history table.
   - For scratch copy storage pool volumes, the record in the database is deleted. For private copy storage pool volumes, the access is updated 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 transition 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:

   - The volumes are now onsite and can be reused or disposed.
   - The database backup volumes are deleted from the volume history table.
   - For scratch copy storage pool volumes, the record in the database is deleted. For private copy storage pool volumes, the access is updated to read/write.

Summary of Disaster Recovery Manager Daily Tasks

This section summarizes the use of DRM during routine operations and during disaster recovery.

Setup

1. License DRM
2. Ensure the device configuration and volume history files exist.
3. Back up the storage pools.
4. Do a full backup the database (for example, a database snapshot backup).
5. Define site-specific server recovery instructions.
6. Describe priority client machines.
7. Generate the disaster recovery plan.

Daily Preparation Operations

Day 1

1. Back up client files.
2. Back up the primary storage pools.
3. Back up the database (for example, a database snapshot backup).
4. Mark the backup volumes as unavailable to Tivoli Storage Manager.
5. Send the backup volumes and disaster recovery plan file to the vault.
6. Generate the disaster recovery plan.

Day 2

1. Back up client files
2. Back up the primary storage pools.
3. Back up the database (for example, a database snapshot backup).
4. Mark the backup volumes as unavailable to Tivoli Storage Manager.
5. Send the backup volumes and disaster recovery plan file to the vault.
6. Generate the disaster recovery plan.

Day 3

1. Automatic storage pool reclamation processing occurs.
2. Back up client files.
3. Back up the primary storage pools.
4. Back up the database (for example, a database snapshot backup).
5. Send the backup volumes and a list of expired volumes to be reclaimed to the vault.
6. The vault acknowledges receipt of the volumes sent on the previous day.
7. Generate the disaster recovery plan.

Disaster and Recovery

Day 4

The server and the client machines are destroyed.

1. Restore the server using the latest recovery plan.
2. Identify the top priority client nodes at the disaster site.
3. Restore client machine files from the copy storage pools.
4. Restore the primary storage pools.
5. Move database backup and copy storage pool volumes to the vault.

Daily Operations

Day 5

1. Back up client files.
2. Back up the primary storage pools.
3. Back up the database (for example, a database snapshot backup).
4. Send the backup volumes and a list of expired volumes to be reclaimed to the vault.
5. Generate the disaster recovery plan.

Staying Prepared for a Disaster

This section provides an overview and a scenario of the tasks required to stay prepared for a disaster. The steps are performed by the onsite Tivoli Storage Manager administrator unless otherwise indicated.

1. Record the following information in the RECOVERY.INSTRUCTIONS stanza source files:
   - Software license numbers
   - Sources of replacement hardware
   - Any recovery steps specific to your installation
2. Store the following information in the database:
   - Server and client node machine information (DEFINE MACHINE, DEFINE MACHINENODE ASSOCIATION, and INSERT MACHINE)
   - The location of the boot recovery media (DEFINE RECOVERYMEDIA)
3. Schedule automatic nightly backups to occur in the following order:
   a. Primary Storage Pools
   b. Database
4. Daily, create a list of the previous night's database and storage pool backup volumes to be sent offsite:

   ```
   query drmedia * wherestate=mountable
   ```

   a. Check the volumes out of the library:

      ```
      move drmedia * wherestate=mountable
      ```

   b. Send the volumes offsite and record that the volumes were given to the courier:

      ```
      move drmedia * wherestate=notmountable
      ```
5. Create a new recovery plan:

   ```
   prepare
   ```
6. Copy the recovery plan file to a diskette to be given 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 and storage pool backup tapes, the recovery plan file diskette, and the list of volumes to be returned from the vault.
9. The courier gives you any tapes that were on the previous day's return from the vault list.

Update the state of these tapes and check them into the library:

```
move drmedia * wherestate=courierretrieve cmdf=/drm/checkin.libvol
  cmd="checkin libvol libauto &vol  status=scratch"
```

The volume records for the tapes that were in the COURIERRETRIEVE state are deleted from the database. The MOVE DRMEDIA command also generates the CHECKIN LIBVOL command for each tape processed in the file */drm/checkin.libvol*. For example:

```
checkin libvol libauto tape01 status=scratch
checkin libvol libauto tape02 status=scratch
...
```

Note: An administrator can run the MACRO command by specifying /drm/checkin.libvol.

```
> dsmadmc -id=xxxxx -pa=yyyyyy MACRO /drm/checkin.libvol
```

10. The courier takes the database and storage pool backup tapes, the recovery plan diskette, and the list of volumes to return from the vault.
11. Call the vault and verify that the backup tapes arrived and are secure, and that the tapes to be returned to the site have been given to the courier.
12. Set the location of the volumes sent to the vault:

    ```
    move drmedia * wherestate=courier
    ```

13. Set the location of the volumes given to the courier by the vault:

    ```
    move drmedia * wherestate=vaultretrieve
    ```

Recovering From a Disaster

This section provides an overview of the tasks involved in recovering the server and clients. It also presents scenarios of both procedures.

Recovering the Server: Here are guidelines for recovering your server:

1. Obtain the latest disaster recovery plan file.
2. Break out the file to view, update, print, or run as macros or scripts (for example, batch programs or batch files).
3. Obtain the backup 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 724).

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.

Server Recovery Scenario

Here is the procedure for a complete recovery of the server after a disaster has destroyed it. In this example virtual volumes are not used. The steps are performed by the onsite administrator unless otherwise indicated.

1. Review the recovery steps described in the RECOVERY.INSTRUCTIONS.GENERAL stanza of the plan.
2. Request the server backup tapes from the offsite vault.
3. Break out the recovery plan file stanzas into multiple files (see "Breaking Out a Disaster Recovery Plan File" on page 717.) 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 backup volumes.
5. Find a replacement server machine. 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. Restore the operating system and the Tivoli Storage Manager server software on the replacement server machine. The media names and their locations are specified in the RECOVERY.INSTRUCTIONS.INSTALL stanza and the MACHINE.RECOVERY.MEDIA.REQUIRED stanza. Ensure that the environment is the same as when the disaster recovery plan file was created. The environment includes:
   - The directory structure of the Tivoli Storage Manager server executable and disk formatting utility
   - The directory structure for Tivoli Storage Manager server configuration files (disk log, volume history file, device configuration file, and server options file)
   - The directory structure and the files created when the disaster recovery plan file was split into multiple files
7. Review the Tivoli Storage Manager macros contained in the recovery plan.

   If, at the time of the disaster, the courier had not picked up the previous night's database and storage pool incremental backup volumes but they were not destroyed, remove the entry for the storage pool backup volumes from the COPYSTGPOOL.VOLUMES.DESTROYED file.
8. If some required storage pool backup volumes could not be retrieved from the vault, remove the volume entries from the COPYSTGPOOL.VOLUMES.AVAILABLE file.
9. If all primary volumes were destroyed, no changes are required to the PRIMARY.VOLUMES script and Tivoli Storage Manager macro files.
10. 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 659.

11. 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 are some sample steps from a recovery script:

    a. Copy the Tivoli Storage Manager server options file from the DSMSERV.OPT file to its original location.
    b. Copy the volume history file required by database restore processing from the VOLUME.HISTORY.FILE file to its original location.

       Note: Use this copy of the volume history file unless you have a more recent copy (after the disaster occurred).
    c. Copy the device configuration file required by database restore processing from the DEVICE.CONFIGURATION.FILE file to its original location.
    d. Create the Tivoli Storage Manager server recovery log and database volumes using DSMFMT.
    e. Issue DSMSERV FORMAT command to format the recovery log and database files.
    f. Issue the DSMSERV RESTORE DB command.
    g. Start the server.
    h. Register Tivoli Storage Manager server licenses.
    i. Mark copy storage pool volumes retrieved from vault as available.
    j. Mark copy storage pool volumes that cannot be obtained as unavailable.
    k. Mark primary storage pool volumes as *destroyed*.

    Notes:

    a. Due to changes in hardware configuration during recovery, you might have to update the device configuration file located in the restored Tivoli Storage Manager database (see "Updating the Device Configuration File" on page 659).
    b. You can mount copy storage 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.
    c. Use the AUDIT LIBRARY command to ensure that the restored Tivoli Storage Manager database is consistent with the automated library volumes.
12. If client machines are not damaged, invoke the RECOVERY.SCRIPT.NORMAL.MODE script file to restore the server primary storage pools. If client machines are damaged, you may want to delay this action until after all clients are recovered.

    Note: This action is optional because Tivoli Storage Manager can access the copy storage 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 machines 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.

13. Collect the database backup and copy storage 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 used in the recovery already have the correct ORMSTATE.

14. Issue the BACKUP DB command to back up the newly restored database.
15. Issue the following command to check the volumes out of the library:

```
move drmedia * wherestate=mountable
```

16. Create a list of the volumes to be given to the courier:

```
query drmedia * wherestate=notmountable
```

17. Give the volumes to the courier and issue the following command:

```
move drmedia * wherestate=notmountable
```

18. Issue the PREPARE command.

Client Recovery Scenario

The following scenario demonstrates the recovery of clients.

1. To view a list of client machines that were lost in building 21 and their restore priority, issue the following command:

```
query machine building=021 format=detailed
```

    DRM displays information similar to the following:

```
          Machine Name: POLARIS
      Machine Priority: 1
              Building: 21
                 Floor: 2
                  Room: 1
               Server?: No
           Description: Payroll
             Node Name: POLARIS
   Recovery Media Name: MKSYSB1
      Characteristics?: Yes
Recovery Instructions?: Yes
```

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. For example:

   `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.

Recovering When Using Different Hardware at the Recovery Site

You may have to recover your system using hardware that is different from that used when you backed up your database and created disaster recovery plan file. Before restoring the database, update the device configuration file included in the recovery plan file. After restoring the database, update the device configuration on the database.

This section describes a number of such situations in detail. If the hardware environment is different at the recovery site, you must update the device configuration file. Tivoli Storage Manager uses the device configuration file to access the devices that are needed to read the database backup volumes. The RECOVERY.VOLUMES.REQUIRED stanza in the plan file identifies the volumes that are needed to restore the database.

Automated SCSI Library at the Original Site and a Manual SCSI Library at the Recovery Site

Ensure that the DEFINE DRIVE and DEFINE LIBRARY commands in the device configuration file are valid for the new hardware configuration. For example, if an automated tape library was used originally and cannot be used at the recovery site, update the device configuration file. Include the DEFINE LIBRARY and DEFINE DRIVE commands that are needed to define the manual drive to be used. In this case, you must manually mount the backup volumes.

Note: If you are using an automated library, you may also need to update the device configuration file to specify the location of the database backup volume.

Here is an example of an original device configuration file, which describes an automated tape library:

```
/*  Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
  mountlimit=2 mountwait=60 mountretention=60
  prefix=tsm library=auto8mmlib

define library auto8mmlib libtype=scsi

define drive auto8mmlib 8mm_tape0 element=82 online=yes

define drive auto8mmlib 8mm_tape1 element=83 online=yes

define path server1 auto8mmlib srctype=server desttype=library
 device=/dev/rmt/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.

Automated SCSI Library at the Original and Recovery Sites

Manually place the database backup volumes in the automated library and note the element numbers where you place them. Then update the comments in the device configuration file to identify the locations of those volumes.

Note: You may also need to audit the library after the database is restored in order to update the server inventory of the library volumes.

Here is an example of an original device configuration file, which describes an automated tape library:

```
/*  Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
  mountlimit=2 mountwait=60 mountretention=60
  prefix=tsm library=auto8mmlib

define library auto8mmlib libtype=scsi

define drive auto8mmlib 8mm_tape0 element=82 online=yes

define drive auto8mmlib 8mm_tape1 element=83 online=yes

define path server1 auto8mmlib srctype=server desttype=library
 device=/dev/rmt/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 informatiion to restore the database restore. Comments that no longer apply at the recovery site are removed.

Managing Copy Storage Pool Volumes at the Recovery Site

The RECOVERY.VOLUMES.REQUIRED stanza in the recovery plan file identifies the required copy storage pool volumes. 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. These volumes must be available to the restored server. After the database is restored, you can handle copy storage 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.

Note: 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 42. Checklist

Activity	Start Date	End Date	Status	Person Resp.	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					
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 REUSEDELAY and DRMDBBACKUPEXPIREDAYS are the same.					
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					
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 – Tape/diskette 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?					

Table 42. Checklist (continued)

Activity	Start Date	End Date	Status	Person Resp.	Backup Person
Determine where you want to create the user-specified recovery instructions What is the prefix of the instructions pathname?					
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					
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 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 42. Checklist (continued)

Activity	Start Date	End Date	Status	Person Resp.	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 primary storage pools • Back up the Tivoli Storage Manager database • QUERY DRMEDIA to list the backup 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.

Breaking Out a Disaster Recovery Plan File

You can use an awk script or an editor to break out the stanzas into individual files. A sample procedure, *planexpl.awk.smp*, is shipped with DRM and is located in */opt/tivoli/tsm/server/bin* or wherever the server resides. You can modify this procedure for your installation. Store a copy of the procedure offsite for recovery.

Structure of the Disaster Recovery Plan File

The disaster recovery plan is divided into the following types of stanzas:

Command stanzas
Consist of scripts (for example, batch programs or batch files) and Tivoli Storage Manager macros. You can view, print, and update these stanzas, and run them during recovery.

Note: The RECOVERY.SCRIPT.NORMAL.MODE and RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE stanzas contain the commands that invoke the scripts and macros contained in the other stanzas.

Instruction stanzas
Consist of recovery instructions specific to your site. You can view, print, and update these stanzas, and use them during recovery.

Server requirements stanzas
Include the database and recovery log requirements, device and volume requirements, and license information. You can view and print these stanzas, and use them during recovery.

Configuration file stanzas
Consist of the volume history, device configuration, and server options files.

Machine and recovery media stanzas
Consist of machine recovery instructions and information about machine hardware, software, and recovery media. You can print and update these stanzas, and use them during server recovery.

Table 43 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 43. 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

Table 43. Administrative Tasks Associated with the Disaster Recovery Plan File (continued)

Stanza Name	Tasks
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)
LOGANDDB.VOLUMES.CREATE script	**During disaster recovery:** Edit and run (optional)
LOG.VOLUMES	**During disaster recovery:** Optionally edit/copy
DB.VOLUMES	**During disaster recovery:** Optionally edit/copy
LOGANDDB.VOLUMES.INSTALL script	**During disaster recovery:** Edit and run (optional)
LICENSE.REGISTRATION macro	**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.CREATE script	**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)

Table 43. Administrative Tasks Associated with the Disaster Recovery Plan File (continued)

Stanza Name	Tasks
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.

Description and Table of Contents Stanzas

PLANFILE.DESCRIPTION

Identifies the server for this recovery plan, and the date and time the plan is created.

```
begin PLANFILE.DESCRIPTION

Recovery Plan for Server SERVER1
Created by DRM PREPARE on 12/14/2004 10:03:08 AM
DRM PLANPREFIX /opt/tivoli/tsm/server/bin/
Storage Management Server for Linux/xxxx - Version 5, Release 2, Level 4.0

end PLANFILE.DESCRIPTION
```

Figure 105. Description Stanza

PLANFILE.TABLE.OF.CONTENTS

Lists the stanzas documented in this plan.

```
 begin PLANFILE.TABLE.OF.CONTENTS

PLANFILE.DESCRIPTION
PLANFILE.TABLE.OF.CONTENTS

Server Recovery Stanzas:
  SERVER.REQUIREMENTS
  RECOVERY.INSTRUCTIONS.GENERAL
  RECOVERY.INSTRUCTIONS.OFFSITE
  RECOVERY.INSTRUCTIONS.INSTALL
  RECOVERY.VOLUMES.REQUIRED
  RECOVERY.DEVICES.REQUIRED
  RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script
  RECOVERY.SCRIPT.NORMAL.MODE script
  LOGANDDB.VOLUMES.CREATE script
  LOG.VOLUMES
  DB.VOLUMES
  LOGANDDB.VOLUMES.INSTALL script
  LICENSE.REGISTRATION macro
  COPYSTGPOOL.VOLUMES.AVAILABLE macro
  COPYSTGPOOL.VOLUMES.DESTROYED macro
  PRIMARY.VOLUMES.DESTROYED macro
  PRIMARY.VOLUMES.REPLACEMENT.CREATE script
  PRIMARY.VOLUMES.REPLACEMENT macro
  STGPOOLS.RESTORE macro
  VOLUME.HISTORY.FILE
  DEVICE.CONFIGURATION.FILE
  DSMSERV.OPT.FILE

Machine Description Stanzas:
  MACHINE.GENERAL.INFORMATION
  MACHINE.RECOVERY.INSTRUCTIONS
  MACHINE.CHARACTERISTICS
  MACHINE.RECOVERY.MEDIA.REQUIRED

end PLANFILE.TABLE.OF.CONTENTS
```

Figure 106. Table of Contents Stanza

Server Requirements Stanza

SERVER.REQUIREMENTS

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 volumes. This stanza also identifies the directory where the server executable resided when the server was started. If the server executable 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
- LOGANDDB.VOLUMES.CREATE
- LOGANDDB.VOLUMES.INSTALL
- PRIMARY.VOLUMES.REPLACEMENT.CREATE

```
 begin SERVER.REQUIREMENTS

Database Requirements Summary:

    Available Space (MB): 20
  Assigned Capacity (MB): 20
        Pct. Utilization: 2.2
Maximum Pct. Utilization: 2.2
        Physical Volumes: 2

Recovery Log Requirements Summary:

    Available Space (MB): 20
  Assigned Capacity (MB): 20
        Pct. Utilization: 4.4
Maximum Pct. Utilization: 4.8
        Physical Volumes: 2
Server Executable Directory: /opt/tivoli/tsm/server/bin
end SERVER.REQUIREMENTS
```

Figure 107. Server Requirements Stanza

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 692 for details.

Note: In the following descriptions, *prefix* represents the prefix portion of the file name. See "Specifying Defaults for the Disaster Recovery Plan File" on page 688 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
```

Figure 108. Recovery Instructions General Stanza

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
```

Figure 109. Recovery Instructions Offsite Stanza

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 Linux/xxx.
 The Linux 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 Linux and installation
 is Fred Myers.

end RECOVERY.INSTRUCTIONS.INSTALL
```

Figure 110. Recovery Instructions Install Stanza

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

RECOVERY.VOLUMES.REQUIRED

Provides a list of the database backup and copy storage 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 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 = OffsiteVault Inc.
  Device Class = LIB8MM
  Volume Name =
   TPBK08
 Location = OffsiteVault Inc.
  Device Class = LIB8MM
  Volume Name =
   TPBK06

Volumes required for storage pool restore
 Location = OffsiteVault Inc.
  Copy Storage Pool = CSTORAGEPF
  Device Class = LIB8MM
  Volume Name =
   TPBK05
   TPBK07

end RECOVERY.VOLUMES.REQUIRED
```

Figure 111. Volume Requirements Stanza

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: LIB8MM
        Device Access Strategy: Sequential
           Storage Pool Count: 2
                  Device Type: 8MM
                       Format: DRIVE
         Est/Max Capacity (MB): 4.0
                  Mount Limit: 2
              Mount Wait (min): 60
         Mount Retention (min): 10
                 Label Prefix: TIVSM
                      Library: RLLIB
                    Directory:
Last Update by (administrator): Bill
         Last Update Date/Time: 12/11/2000 10:18:34

end RECOVERY.DEVICES.REQUIRED
```

Figure 112. Volume and Device Requirements Stanzas

Disaster Recovery Mode Stanza

RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE

Contains a script with the commands needed to recover the server. You can use the script as a guide and run the commands from a command line. Or you can copy it

to a file, modify it and the files it refers to, and run the script. You may need to modify the script because of differences between the original and the replacement systems. At the completion of these steps, client requests for file restores are satisfied directly from copy storage pool volumes.

The disaster recovery plan issues commands using the administrative client. The disaster recovery plan file issues commands using the administrative client. Ensure that the path to the administrative client is established before running the script. For example, set the shell variable PATH or update the scripts with the path specification for the administrative client.

The commands in the script do the following:

- Restore the server options, volume history, and device configuration information files.
- Invoke the scripts contained in the LOGANDDB.VOLUMES.CREATE and LOGANDDB.VOLUMES.INSTALL stanzas.

  Attention: When this script runs, any log volumes or database volumes with the same names as those named in the plan are *removed* (see LOGANDDB.VOLUMES.CREATE under "Create and Install Database and Recovery Log Volumes Stanzas" on page 729). In most disaster recoveries, the Tivoli Storage Manager server is installed on a new machine. When this script is run, it is assumed that there is no Tivoli Storage Manager data in the log or database volumes. Tivoli Storage Manager installation includes the creation of database and recovery log volumes. If you have created a log volume or a database volume (for example, for testing), and you want to preserve the contents, you must take some action such as renaming the volume or copying the contents before executing this script.
- Invoke the macros contained in the following stanzas:
  - LICENSE.REGISTRATION
  - COPYSTGPOOL.VOLUMES.AVAILABLE
  - COPYSTGPOOL.VOLUMES.DESTROYED
  - PRIMARY.VOLUMES.DESTROYED.

To help understand the operations being performed in this script, see "Backup and Recovery Scenarios" on page 680.

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/ivoli/tsm/client/admin/bin*.

For example, to invoke this script using an administrator ID of *don*, password of *mox*, server name of *prodtsm*, enter the following command:

```
planprefix/RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE don mox prodtsm
```

For more information, see the entry for the recovery plan prefix in Table 40 on page 688.

```
 begin RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script

#!/bin/bash
set -x

 # Purpose: This script contains the steps required to recover the server
 #   to the point where client restore requests can be satisfied
 #   directly from available copy storage pool volumes.
 # Note: This script assumes that all volumes necessary for the restore have
 #   been retrieved from the vault and are available. This script assumes
 #   the recovery  environment is compatible (essentially the same) as the
 #   original.  Any deviations require modification to this script and the
 #   macros and shell scripts it runs.  Alternatively, you can use this
 #   script as a guide, and manually execute each step.

if [ -z "$1" -o -z "$2" -o -z "$3" ]
then
  print "Specify the following positional parameters:"
  print "administrative client ID, password, and server ID."
  print "Script stopped."
  exit
fi
 # Set the  server working directory
cd /opt/tivoli/tsm/server/bin

 # Restore server options, volume history, device configuration files.
cp /prepare/DSMSERV.OPT.FILE \
    /opt/tivoli/tsm/server/bin/dsmserv.optx
cp /prepare/VOLUME.HISTORY.FILE \
    /opt/tivoli/tsm/server/bin/volhistory.txtx
cp /prepare/DEVICE.CONFIGURATION.FILE \
    /opt/tivoli/tsm/server/bin/devconfig.txtx

export DSMSERV_CONFIG=/opt/tivoli/tsm/server/bin/dsmserv.optx

export DSMSERV_DIR=/opt/adsmserv/bin
```

Figure 113. Disaster Recovery Mode Script (Part 1 of 2)

```
 # Create and format log and database files.
/prepare/LOGANDDB.VOLUMES.CREATE 2>&1 \
| tee /prepare/LOGANDDB.VOLUMES.CREATE.log

 # Initialize the log and database files.
/prepare/LOGANDDB.VOLUMES.INSTALL 2>&1 \
| tee /prepare/LOGANDDB.VOLUMES.INSTALL.log

 # Restore the  server database to latest version backed up per the
 # volume history file.
/opt/tivoli/tsm/server/bin/dsmserv restore db todate=08/11/2000 totime=10:20:22

 # Start the server.
nohup /opt/tivoli/tsm/server/bin/dsmserv &
print Please start new  server console with command dsmadmc -CONSOLE.
print Press enter to continue recovery script execution.
read pause

 # Register  Server Licenses.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT
  -OUTFILE=/prepare/LICENSE.REGISTRATION.log
    macro /prepare/LICENSE.REGISTRATION.mac

 # Tell  Server these copy storage pool volumes are available for use.
 # Recovery Administrator: Remove from macro any volumes not obtained from vault.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/COPYSTGPOOL.VOLUMES.AVAILABLE.log \
     macro /prepare/COPYSTGPOOL.VOLUMES.AVAILABLE

 # Volumes in this macro were not marked as 'offsite' at the time
 # PREPARE ran. They were likely destroyed in the disaster.
 # Recovery Administrator: Remove from macro any volumes not destroyed.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/COPYSTGPOOL.VOLUMES.DESTROYED.log \
     macro /prepare/COPYSTGPOOL.VOLUMES.DESTROYED
 # Mark primary storage pool volumes as ACCESS=DESTROYED.
 # Recovery administrator: Remove from macro any volumes not destroyed.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/PRIMARY.VOLUMES.DESTROYED.log \
     macro /prepare/PRIMARY.VOLUMES.DESTROYED

end RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script
```

Figure 113. Disaster Recovery Mode Script (Part 2 of 2)

Normal Mode Stanza

RECOVERY.SCRIPT.NORMAL.MODE

Contains a script with the commands needed to restore the server primary storage pools. 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. The disaster recovery plan file issues commands using the administrative client. Ensure that the path to the administrative client is established before running the script. For example, set the shell variable PATH or update the scripts with the path specification for the administrative client.

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 script contained in the PRIMARY.VOLUMES.REPLACEMENT.CREATE stanza: It also invokes the macros contained in the following stanzas:

PRIMARY.VOLUMES.REPLACEMENT
STGPOOLS.RESTORE

To help understand the operations being performed in this script, see "Backup and Recovery Scenarios" on page 680.

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 40 on page 688.

```
begin RECOVERY.SCRIPT.NORMAL.MODE script
#!/bin/bash
set -x

 # Purpose: This script contains the steps required to recover the server
 #          primary storage pools. This mode allows you to return the
 #          copy storage pool volumes to the vault and to run the
 #          server as normal.
 # Note: This script assumes that all volumes necessary for the restore
 #   have been retrieved from the vault and are available. This script
 #   assumes the recovery  environment is compatible (essentially the
 #   same) as the original. Any deviations require modification to this
 #   script and the macros and shell scripts it runs. Alternatively,
 #   you can use this script as a guide, and manually execute each step.

if [ -z "$1" -o -z "$2" -o -z "$3" ]
then
  print "Specify the following positional parameters:"
  print "administrative client ID, password, and server ID."
  print "Script stopped."
  exit
fi

 # Create replacement volumes in the primary storage pools (If any
 # are implemented as disk but not logical volume.)
 # Recovery administrator: Edit script for your replacement volumes.
/prepare/PRIMARY.VOLUMES.REPLACEMENT.CREATE 2>&1 \
| tee /prepare/PRIMARY.VOLUMES.REPLACEMENT.CREATE.log

 # Define replacement volumes in the primary storage pools. Must
 # have different name than original.
 # Recovery administrator: Edit macro for your replacement volumes.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/PRIMARY.VOLUMES.REPLACEMENT.log \
     macro /prepare/PRIMARY.VOLUMES.REPLACEMENT

 # Restore the primary storage pools from the copy storage pools.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/STGPOOLS.RESTORE.log \
     macro /prepare/STGPOOLS.RESTORE

end RECOVERY.SCRIPT.NORMAL.MODE script
```

Figure 114. Normal Mode Script

Create and Install Database and Recovery Log Volumes Stanzas

LOGANDDB.VOLUMES.CREATE

Contains a script with the commands needed to recreate the database and log volumes. You can use the script as a guide and issue the commands as needed from a command line, or you can copy it to a file, modify it, and run it. This script is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

If you used raw partitions for log or database volumes, you must create these volumes before running the LOGANDDB.VOLUMES.INSTALL script.If you want to use regular files instead of raw partitions, modify this script to include the appropriate DSMFMT commands.

The plan assumes that the volume formatting command (DSMFMT) resides in the same directory as the server executable indicated in the stanza SERVER.REQUIREMENTS.

```
begin LOGANDDB.VOLUMES.CREATE script
#!/bin/bash
set -x
 # Purpose:  Create log and database volumes.
 # Recovery Administrator: Run this to format  server log and
 #  database volumes.
  print Remove database volume /opt/tivoli/tsm/server/bin/db01x.
 rm -f /opt/tivoli/tsm/server/bin/db01x

  print Create  database volume /opt/tivoli/tsm/server/bin/db01x 12M
 /opt/tivoli/tsm/server/bin/dsmfmt -m -db /opt/tivoli/tsm/server/bin/db01x 12M

  print Remove database volume /opt/tivoli/tsm/server/bin/db02x.
 rm -f /opt/tivoli/tsm/server/bin/db02x

  print Create  database volume /opt/tivoli/tsm/server/bin/db02x 8M
 /opt/tivoli/tsm/server/bin/dsmfmt -m -db /opt/tivoli/tsm/server/bin/db02x 8

  print Remove log volume /opt/tivoli/tsm/server/bin/lg01x.
 rm -f /opt/tivoli/tsm/server/bin/lg01x

  print Create  log volume /opt/tivoli/tsm/server/bin/lg01x 12M
 /opt/tivoli/tsm/server/bin/dsmfmt -m -log /opt/tivoli/tsm/server/bin/lg01x 12M

  print Remove log volume /opt/tivoli/tsm/server/bin/lg02x.
 rm -f /opt/tivoli/tsm/server/bin/..lg02x

  print Create  log volume /opt/tivoli/tsm/server/bin/lg02x 8M
 /opt/tivoli/tsm/server/bin/dsmfmt -m -log /opt/tivoli/tsm/server/bin/lg02x 8

end LOGANDDB.VOLUMES.CREATE script
```

Figure 115. Create Database and Recovery Log Volumes Stanza

LOG.VOLUMES

Contains the names of the log volumes to be initialized. The contents of this stanza must be placed into a separate file to be used by the LOGANDDB.VOLUMES.INSTALL script.

```
begin LOG.VOLUMES
/opt/tivoli/tsm/server/bin/lg01x
/opt/tivoli/tsm/server/bin/lg02x

end LOG.VOLUMES
```

Figure 116. Recovery Log Volumes Stanza

DB.VOLUMES

Contains the names of the database volumes to be initialized. The contents of this stanza must be placed into a separate file to be used by the LOGANDDB.VOLUMES.INSTALL script.

```
begin DB.VOLUMES

/opt/tivoli/tsm/server/bin/db01x
/opt/tivoli/tsm/server/bin/db02x

end DB.VOLUMES
```

Figure 117. Database Volume Stanza

LOGANDDB.VOLUMES.INSTALL

Contains a script with the commands required to initialize the database and log volumes. This script is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

```
 begin LOGANDDB.VOLUMES.INSTALL script

#!/bin/bash
set -x

 # Purpose: Initialize the log and database volumes.
 # Recovery Administrator: Run this to initialize an  server.

 /opt/tivoli/tsm/server/bin/dsmserv install \
   2 FILE:/prepare/LOG.VOLUMES \
   2 FILE:/prepare/DB.VOLUMES

end LOGANDDB.VOLUMES.INSTALL script
```

Figure 118. Install Database and Recovery Log Volumes Stanza

License Registration Stanza

LICENSE.REGISTRATION

Contains a macro to register your server licenses. 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 recreate the      */
 /*  licenses that existed in the  server.                              */
 /* Recovery Administrator: Review licenses and add or delete licenses  */
 /*  as necessary.                                                      */

register license file(50client.lic)
register license file(network.lic)
register license file(drm.lic)

end LICENSE.REGISTRATION macro
```

Figure 119. License Registration Macro Stanza

Copy Storage Pool Volumes Stanzas

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             */
 /*   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 TPBK05 acc=READW wherestg=CSTORAGEPF
 upd vol TPBK07 acc=READW wherestg=CSTORAGEPF

end COPYSTGPOOL.VOLUMES.AVAILABLE macro
```

Figure 120. Copy Storage Pool Volumes Available Stanza

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
```

Figure 121. Copy Storage Pool Volumes Destroyed Stanza

Primary Storage Volumes Stanzas

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         */
 /*   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.                      */

 upd vol /opt/tivoli/tsm/server/bin/bk02 acc=DESTROYED wherestg=BACKUPPOOL
 upd vol /opt/tivoli/tsm/server/bin/bk01x acc=DESTROYED wherestg=BACKUPPOOL
 upd vol /opt/tivoli/tsm/server/bin/bk03 acc=DESTROYED wherestg= BACKUPPOOLF
 upd vol BACK4X acc=DESTROYED wherestg=BACKUPPOOLT

end PRIMARY.VOLUMES.DESTROYED macro
```

Figure 122. Primary Storage Volumes Destroyed Stanza

PRIMARY.VOLUMES.REPLACEMENT.CREATE

Contains a script with the commands needed to recreate the primary disk storage pool volumes. You can use the script as a guide and run the commands from a command line, or you can copy thew script to a file, modify it, and run it. This script is invoked by the RECOVERY.SCRIPT.NORMAL.MODE script.

If you used raw partitions for the primary storage pool volumes, create the replacement volumes by running the PRIMARY.VOLUMES.REPLACEMENT macro. To use regular files instead of raw partitions, modify this script to include the appropriate DSMFMT commands.

The plan file assumes that the volume formatting program (DSMFMT) resides in the same directory as the server executable indicated in the stanza SERVER.REQUIREMENTS.

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 retrievability of volume names that require renaming in the stanzas. Before using the volume names, change these names to new names that are valid for the device class and valid 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.

  Notes:

  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 will be entries for the same volumes in PRIMARY.VOLUMES.REPLACEMENT.

This stanza does not include primary storage pool virtual volumes, because these volumes are considered offsite and have not been destroyed in a disaster.

```
begin PRIMARY.VOLUMES.REPLACEMENT.CREATE script

#!/bin/bash
set -x

 # Purpose: Create replacement volumes for primary storage pools that
 #   use device class DISK.
 # 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.

   print Replace /opt/tivoli/tsm/server/bin/bk02 DISK 16M in BACKUPPOOL
 /opt/tivoli/tsm/server/bin/dsmfmt -m -data /opt/tivoli/tsm/server/bin/bk02@ 16

   print Replace /opt/tivoli/tsm/server/bin/bk01x DISK 5M in BACKUPPOOL
 /opt/tivoli/tsm/server/bin/dsmfmt -m -data /opt/tivoli/tsm/server/bin/bk01x@ 5

end PRIMARY.VOLUMES.REPLACEMENT.CREATE script
```

Figure 123. Primary Storage Volumes Replacement Stanza

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:

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 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.

  Notes:

  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.CREATE and 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.

```
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 /opt/tivoli/tsm/server/bin/bk02 DISK 16M in BACKUPPOOL */
 def vol BACKUPPOOL /opt/tivoli/tsm/server/bin/bk02@ acc=READW

   /* Replace /opt/tivoli/tsm/server/bin/bk01x DISK 5M in BACKUPPOOL */
 def vol BACKUPPOOL f;/bk01x@ acc=READW

   /* Replace /opt/tivoli/tsm/server/bin/bk03 FILES 4M in BACKUPPOOLF */
 def vol BACKUPPOOLF /opt/tivoli/tsm/server/bin/bk03@ acc=READW

   /* Replace BACK4X COOL8MM 0M in BACKUPPOOLT */
 def vol BACKUPPOOLT BACK4X@ acc=READW

end PRIMARY.VOLUMES.REPLACEMENT macro
```

Figure 124. Primary Storage Volumes Replacement Stanza

Storage Pools Restore Stanza

STGPOOLS.RESTORE

Contains a macro to restore the primary storage pools. You can use it 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.

```
begin 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 ARCHIVEPOOL
 restore stgp BACKUPPOOL
 restore stgp BACKUPPOOLF
 restore stgp BACKUPPOOLT
 restore stgp SPACEMGPOOL

end STGPOOLS.RESTORE macro
```

Figure 125. Storage Pools Restore Stanza

Configuration Stanzas

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 (for example, '.' or '/'), 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*. For example, if *volhprefix* is /opt/tivoli/tsm/server/bin, the file name is /opt/tivoli/tsm/server/bin/drmvolh.txt.

Note: The *volhprefix* is set based on the following:

- If the environmental variable DSMSERV_DIR has been defined, it is used as the *volhprefix*.
- If the environmental variable DSMSERV_DIR has not been defined, 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.

```
 begin VOLUME.HISTORY.FILE

*************************************************************************
*
*               Tivoli Storage Manager Sequential Volume Usage History
*                                     Updated 02/11/2000 10:20:34
*
*     Operation         Volume   Backup Backup Volume Device      Volume
*     Date/Time          Type    Series Oper.   Seq   Class Name  Name
*************************************************************************
 2000/08/11 10:18:43  STGNEW          0      0      0 COOL8MM     BACK4X
 2000/08/11 10:18:43  STGNEW          0      0      0 FILES       BK03
 2000/08/11 10:18:46  STGNEW          0      0      0 LIB8MM      TPBK05
* Location for volume TPBK06 is: 'Ironvault Inc.'
 2000/08/11 10:19:23  BACKUPFULL      1      0      1 LIB8MM      TPBK06
 2000/08/11 10:20:03  STGNEW          0      0      0 LIB8MM      TPBK07
 2000/08/11 10:20:22  BACKUPINCR      1      1      1 LIB8MM      TPBK08

end VOLUME.HISTORY.FILE
```

Figure 126. Volume History 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 659.

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 (for example, '.' or '/'), 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.

```
begin DEVICE.CONFIGURATION.FILE

/* Tivoli Storage Manager Device Configuration */
DEFINE DEVCLASS COOL8MM DEVTYPE=8MM FORMAT=DRIVE MOUNTLIMIT=1 MOUNTWAIT=60
MOUNTRETENTION=60 PREFIX=TIVSM LIBRARY=ITSML
DEFINE DEVCLASS FILES DEVTYPE=FILE MAXCAPACITY=4096K MOUNTLIMIT=2 +
DIRECTORY=/opt/tivoli/tsm/server/bin
DEFINE DEVCLASS FILESSM DEVTYPE=FILE MAXCAPACITY=100K MOUNTLIMIT=2 +
DIRECTORY=/opt/tivoli/tsm/server/bin
DEFINE DEVCLASS LIB8MM DEVTYPE=8MM FORMAT=DRIVE MOUNTLIMIT=1 MOUNTWAIT=60+
MOUNTRETENTION=60 PREFIX=TIVSM LIBRARY=RLLIB
end DEVICE.CONFIGURATION.FILE
```

Figure 127. Device Configuration File Stanza

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.

```
begin DSMSERV.OPT.FILE

* Server options file located in /opt/tivoli/tsm/server/bin/dsmserv.optx
TCPPort 1509
VOLUMEHISTORY /opt/tivoli/tsm/server/bin/volhistory.txtx
DEVCONFIG     /opt/tivoli/tsm/server/bin/devconfig.txtx
* The following option was added by PREPARE.
DISABLESCHEDS YES

end DSMSERV.OPT.FILE
```

Figure 128. Server Options File Stanza

License Information Stanza

LICENSE.INFORMATION

Contains a copy of the latest license audit results and the server license terms.

```
begin LICENSE.INFORMATION
                                Last License Audit: 12/30/2000 10:25:34
                           Registered Client Nodes: 1
                             Licensed Client Nodes: 51
                 Are network connections in use ?: Yes
                Are network connections licensed ?: Yes
Are Open Systems Environment clients registered ?: No
  Are Open Systems Environment clients licensed ?: No
                      Is space management in use ?: No
                    Is space management licensed ?: No
            Is disaster recovery manager in use ?: Yes
          Is disaster recovery manager licensed ?: Yes
    Are Server-to-Server Virtual Volumes in use ?: No
  Are Server-to-Server Virtual Volumes licensed ?: Yes
            Is Advanced Device Support required ?: No
            Is Advanced Device Support licensed ?: No
                        Server License Compliance: Valid

end LICENSE.INFORMATION
```

Figure 129. License Information Stanza

Machine Files Stanza

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
```

Figure 130. Machine General Information Stanza

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
```

Figure 131. Machine Recovery Instructions Stanza

MACHINE.RECOVERY.CHARACTERISTICS

Provides the hardware and software characteristics for the server machine.This stanza is included in the plan file if the machine characteristics are saved in the database.

```
begin MACHINE.CHARACTERISTICS
Purpose: Hardware and software characteristics of machine DSMSRV1.
 Filesystem           1k-blocks     Used Available Use% Mounted on
/dev/ida/c0d0p8        1032056    953908     25724  98% /
/dev/ida/c0d0p1         518000     49500    442188  11% /boot
/dev/ida/c0d0p3        5670432   3840664   1541720  72% /devel
/dev/ida/c0d0p6        9573932   1341124   7746476  15% /drivers
/dev/ida/c0d0p2        3096288   2751896    187108  94% /home
/dev/ida/c0d0p5        2064160    698048   1261260  36% /pkgs
none                    191664         0    191664   0% /dev/shm
/dev/ida/c0d0p7        4104144   1369168   2526492  36% /usr
/dev/ida/c0d0p10        518000     44280    447408  10% /var
/dev/ida/c0d0p11       5835056        20   5538628   1% /webend MACHINE.CHARACTERISTICS
```

Figure 132. Machine Recovery Characteristics Stanza

MACHINE.RECOVERY.MEDIA

Provides information about the media (for example, boot media) needed for rebuilding the machine that contains the server. This stanza is included in the plan file if recovery media information is saved in the database and it has been associated with the machine that contains the server.

```
begin MACHINE.RECOVERY.MEDIA.REQUIRED
 Purpose: Recovery media for machine DSMSRV1.
  Recovery Media Name: DSMSRV
                 Type: Other
        Volume Names:
            Location: IRONMNT
         Description: Server Installation CD
             Product:
  Product Information:

  Recovery Media Name: LINUXOS
                 Type: Other
        Volume Names:
            Location: IRONMNT
         Description:  Linux/xxxx  Installation CD
             Product:
  Product Information:
end MACHINE.RECOVERY.MEDIA.REQUIRED
```

Figure 133. Machine Recovery Media Stanza

Appendix A. External Media Management Interface Description

This appendix contains Programming Interface information for the interface that IBM Tivoli Storage Manager provides to external media management programs. The interface 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. See "Configuring Libraries Controlled by Media Manager Programs" on page 131. Refer to your media manager's documentation for that product's setup information and instructions for operational usage.

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 uni-directional pipes and maps them to stdin and stdout during the CreateProcess call. When a standard handle is redirected to refer to a file or a pipe, the handle can only be used by the ReadFile and WriteFile functions. This precludes normal C functions such as gets or printf. Since the server will never terminate the external program process, it is imperative that the external program recognize a read or write failure on the pipes and exit the process. In addition, the external program should exit the process if it reads an unrecognized command.

The external program may obtain values for the read and write handles using the following calls:

```
readPipe=GetStdHandle(STD_INPUT-HANDLE) and writePipe=GetStdHandle(STD_OUTPUT_HANDLE)
```

© Copyright IBM Corp. 1993, 2005

Processing during Server Initialization

Ensure that the external media management program cooperates with the server during the server's initialization. For each external library defined to the server, the following must occur during server initialization:

1. The server loads the external program (CreateProcess) in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string, in text form, into the standard input (stdin) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string, in text form, into its standard output (stdout) stream.
4. The server closes the pipes.
5. When the agent detects that the pipes are closed, it performs any necessary cleanup and calls the stdlib exit routine.

Processing for Mount Requests

To process the mount request:

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 the release request:

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 execution when performed on volumes in external libraries. The move commands will 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. As 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 will result in QUERY requests being sent to the agent. During certain types of processing, Tivoli Storage Manager may need to know if a volume is present in a library. The external agent should verify that the volume is physically present in the library.

1. The server loads the external program in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string (in text form) into the standard input (stdin) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string (in text form) into its standard output (stdout) stream.
4. The server sends the BEGIN BATCH request (stdin).
5. The agent sends the BEGIN BATCH response (stdout).
6. The server sends 1 to n volume requests (n > 1). These can be any number of QUERY or EJECT requests. For each request, the agent will send the applicable QUERY response or EJECT response.
7. The server sends the END BATCH request (stdin).
8. The agent sends the END BATCH response (stdout), performs any necessary cleanup, and calls the stdlib exit routine.

Error Handling

If the server encounters an error during processing, it will close the stdin and stdout streams to the agent exit. The agent will detect this when it tries 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 format of the Begin Batch Request is:

```
BEGIN BATCH
```

Format of the external program response:

```
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
```

Format of the external program response:

```
END BATCH COMPLETE, RESULT=resultCode
```

where:

resultCode
One of the following:
- SUCCESS
- INTERNAL_ERROR

Volume Query Request

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.

Format of the external program response:

```
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, the server sends an initialization request to the external media management program for each EXTERNAL library. The external program must process 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 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.

Format of the external program response:

```
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.

Format of the request:

```
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).

Format of the external program response:

```
RELEASE libraryname volname COMPLETE, RESULT=resultcode
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

volname
Specifies the name of the volume returned to scratch (released).

resultcode
One of the following:
- SUCCESS
- VOLUME_UNKNOWN
- VOLUME_UNAVAILABLE
- INTERNAL_ERROR

Volume Mount Request

When the server requires a volume mount, the server starts the external media management program, issues a request to initialize, then issues a request to mount a volume. The external program is responsible for verifying that this request is coming from Tivoli Storage Manager and not from an unauthorized system.

The volume mounted by the external media management program must be a tape with a standard IBM label that matches the external volume label. When the external program completes the mount request, the program must send a response. If the mount was successful, the external program must remain active. If the mount failed, the external program must end immediately by using the stdlib exit routine.

Format of the request:

```
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 required for the volume. Possible values are `READONLY` and `READWRITE`.

devicetypes
Specifies a list of device types that can be used to satisfy the request for the volume and the FORMAT specified in the device class. The most preferred device type is first in the list. Items are separated by commas, with no intervening spaces. Possible values are:

- 3590
- 3590E
- 3590H
- 3592
- 4MM_DDS1
- 4MM_DDS1C
- 4MM_DDS2
- 4MM_DDS2C
- 4MM_DDS3
- 4MM_DDS3C
- 4MM_HP_DDS4
- 4MM_DDS4
- 4MM_HP_DDS4C
- 4MM_DDS4C
- 4MM_DDS5C
- 8MM_SAIT
- 8MM_VXA2
- 8MM_AITC
- 8MM_AIT
- 8MM_8200
- 8MM_ELIANT
- 8MM_8500
- 8MM_8500C
- 8MM_8205
- 8MM_8900
- 8MM_M2
- DLT_2000
- DLT_4000
- DLT_7000
- DLT_8000
- SDLT
- SDLT320
- DLT1
- DLT2
- SDLT600

- LTO_ULTRIUM
- LTO_ULTRIUM2
- LTO_ULTRIUM3
- QIC_525
- QIC_IBM1000
- IBM_QIC4GBC
- QIC_12GBC
- QIC_5010C
- QIC_20GBC
- QIC_25GBC
- QIC_30GBC
- QIC_50GBC
- QIC_70GBC
- REMOVABLEFILE
- M8100
- STK_SD3
- STK_9840
- STK_T9840C
- STK_9940
- STK_9940B
- STK_9840_VOLSAFE
- STK_T9840C_VOLSAFE
- STK_9940_VOLSAFE
- STK_9940B_VOLSAFE

timelimit
Specifies the maximum number of minutes that the server waits for the volume to be mounted. If the mount request is not completed within this time, the external manager responds with the result code TIMED_OUT.

userid
Specifies the user ID of the process that needs access to the drive.

volumenumber
The *volumenumber* is 1.

'location'
Specifies the value of the location field from the Tivoli Storage Manager inventory (for example, 'Room 617 Floor 2'). One blank character is inserted between the volume number and the left single quotation mark in the location information. If no location information is associated with a volume, nothing is passed to the exit. If no volume information exists, the single quotation marks are not passed. Also, if volume information is passed, then probably the volume has been ejected from the library and needs to be returned to the library before the mount operation can proceed. The location information should be posted by the agent so that the operator can obtain the volume and return it to the library.

Format of the external program response:

```
MOUNT libraryname volname COMPLETE ON specialfile, RESULT=resultcode
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

volname
Specifies the name of the volume mounted for the request.

specialfile
The fully qualified path name of the device special file for the drive in which the volume was mounted. If the mount request fails, the value should be set to /dev/null.

The external program must ensure that the special file is closed before the response is returned to the server.

resultcode
One of the following:
- SUCCESS
- DRIVE_ERROR
- LIBRARY_ERROR
- VOLUME_UNKNOWN
- VOLUME_UNAVAILABLE
- CANCELLED
- TIMED_OUT
- INTERNAL_ERROR

Volume Dismount Request

When a successful mount operation completes, the external process must wait for a request to dismount the volume. When the dismount operation completes, the external program must send a response to the server.

After the dismount response is sent, the external process ends immediately by using the stdlib exit routine.

Format of the request:

```
DISMOUNT libraryname volname
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

volname
Specifies the name of the volume to be dismounted.

Format of the external program response:

```
DISMOUNT libraryname volname COMPLETE, RESULT=resultcode
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to Tivoli Storage Manager.

volname
Specifies the name of the volume dismounted.

resultcode
One of the following:
- SUCCESS
- DRIVE_ERROR
- LIBRARY_ERROR
- INTERNAL_ERROR

Appendix B. User Exit and File Exit Receivers

This appendix contains samples of the user exit receiver for event logging. The data structure of the user exit receivers also 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. You can also use Tivoli Storage Manager commands to control event logging. See "Logging IBM Tivoli Storage Manager Events to Receivers" on page 526 and *Administrator's Reference* for details. The samples for the C, H, and make files are shipped with the server code in the */opt/tivoli/tsm/server/bin* directory.

Notes:

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.

© Copyright IBM Corp. 1993, 2005

Sample User Exit Declarations

```
/*****************************************************************
 * Name:            userExitSample.h
 * Description:     Declarations for a user-exit
 * Environment:     Linux/i386
 *****************************************************************/

#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
```

Figure 134. Sample User Exit Declarations (Part 1 of 3)

```
/**********************************************
 * 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 via *
 *    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)*/
```

Figure 134. Sample User Exit Declarations (Part 2 of 3)

```
  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
```

Figure 134. Sample User Exit Declarations (Part 3 of 3)

Sample User Exit Program

```
/*********************************************************************
 * Name:            userExitSample.c
 * Description:     Example user-exit program invoked by the server
 * Environment:     Linux/i386
 *********************************************************************/

#include <stdio.h>
#include "userExitSample.h"

/**************************************
 *** Do not modify below this line. ***
 **************************************/

extern void adsmV3UserExit( void *anEvent );

/************
 *** Main ***
 ************/

int main(int argc, char *argv[])
{
/* Do nothing, main() is never invoked, but stub is needed */

exit(0);  /* For picky compilers */

} /* End of main() */

/*******************************************************************
 * Procedure:  adsmV3UserExit
 * If the user-exit is specified on the server, a valid and
 * appropriate event causes an elEventRecvData structure (see
 * userExitSample.h) to be passed to adsmV3UserExit that returns a void.
 * INPUT :   A (void *) to the elEventRecvData structure
 * RETURNS:  Nothing
 *******************************************************************/

void adsmV3UserExit( void *anEvent )
{
/* Typecast the event data passed */
elEventRecvData *eventData = (elEventRecvData *)anEvent;
```

Figure 135. Sample User Exit Program (Part 1 of 2)

```
 /*************************************
 *** 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() */
```

Figure 135. Sample User Exit Program (Part 2 of 2)

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 44. 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	Node name
0178-0193	Communications method name
0195-0258	Owner name
0260-0323	High-level internet address (n.n.n.n)
0325-0356	Port number from high-level internet address
0358-0387	Schedule name
0389-0418	Domain name
0420-2019	Event text
2020-2499	Unused spaces

Table 44. Readable Text File Exit (FILETEXTEXIT) Format (continued)

Column	Description
2500	New line character

Appendix C. Setting up the Secure Sockets Layer (SSL)

This section contains instructions for:

- Setting up SSL for Integrated Solutions Console
- Setting up LDAP over SSL

Setting up SSL for Integrated Solutions Console

This section describes the overall tasks required to configure SSL for Integrated Solutions Console. After you complete this procedure, all requests, starting with the portal login, are encrypted.

To set up SSL for Integrated Solutions Console, perform the following procedure:

1. Configure the Web server to support HTTPS. If this is a production environment, you need to obtain a certificate from a certificate authority. For testing purposes, you can use the process described in step 2 to generate a self-signed certificate.
2. Create the key files by opening the IKEYMAN utility located in the <isc_root>/AppServer/bin directory, where <isc_root> is the root directory of the Integrated Solutions Console runtime installation (/opt/IBM/ISC601, by default).
   a. Open it from the command line as ikeyman.sh.
   b. From the menu bar, select **Key Database File -> New**.
   c. Ensure that the Key database type is set to JKS.
   d. Enter ISCV6ServerKeyFile.jks as the file name for the key file.
   e. Enter the directory that will hold the key file as the location, for example: <isc_root>/AppServer/etc.
   f. Click **OK**.
   g. A password prompt appears. Enter a password and repeat to confirm. This password will be required to read from or write to this file in the future, so do not forget it. The password strength is determined by the variety of the characters used in the password.
   h. Click **OK**.
   i. A list of signer certificates should appear which represents the identities of a selection of CAs. This list is not needed, so it can be deleted by selecting each certificate, clicking **Delete** and confirming.
   j. A self-signed certificate can now be generated. From the menu bar, select **Create -> New Self-Signed Certificate**.
   k. A window will appear requesting information in order to generate the certificate. Enter ISCV6IntSec as the Key Label. Ideally, spaces should not be used in the key label.
   l. Select **X509 V3** as the Version and **1024** as the Key Size.
   m. The first two fields of the Distinguished Name, Common Name and Organization, are mandatory. The ISC server name might be entered as the Common Name, for example: ISCsrv01.
   n. While Organization Unit, Locality and State/Province are optional fields, it is recommended that appropriate values be entered; for example: ITSO, Tucson, AZ.

© Copyright IBM Corp. 1993, 2005

o. Select the appropriate Country.
p. Enter **365** as the Validity Period.
q. Click **OK**.
r. The new self-signed certificate should be added to the Personal Certificates list.
s. From the menu bar, select **Key Database File -> Close**. This will close the current key store.
t. From the menu bar, select **Key Database File -> New**.
u. Ensure that the Key database type is set to JKS, and enter ISCV6ServerTrustFile.jks as the file name. Be sure that this file name is the same one used earlier.
v. Enter the directory that will hold the key file as the location, for example: <isc_root>/AppServer/etc.
w. Click **OK**.
x. Enter a password and repeat to confirm. This password will be required to read from or write to this file in the future, so do not forget it.
y. Click **OK**.
z. From the menu bar, select **Key Database File -> Close**.

3. Create an addSSLentry.jacl script in <isc_root>/AppServer/bin to run via wsadmin to enable the ISC to use these keys:
   a. The following sample script creates a new SSL entry. Note that the file names and passwords are hard coded in the file; you have to modify the values or change the script to accept parameters.

```
# new SSL entry in the SSL repertoire
# setting the security object
set security_root [$AdminConfig list Security]
# setting the variables for the entry
set ssl_alias "DefaultNode/DefaultSSLSettings"
set ssl_clientAuthentication [list clientAuthentication false]
set ssl_enableCryptoHardwareSupport
  [list enableCryptoHardwareSupport false]
set ssl_keyFileFormat [list keyFileFormat "JKS"]
set ssl_keyFileName [list keyFileName "<isc_root>/AppServer/
  etc/ISCV6ServerKeyFile.jks"]
set ssl_keyFilePassword [list keyFilePassword "password"]
  set ssl_securityLevel [list securityLevel "HIGH"]
set ssl_trustFileFormat [list trustFileFormat "JKS"]
set ssl_trustFileName [list trustFileName"<isc_root>/AppServer/
etc/ISCV6ServerTrustFile.jks"]
set ssl_trustFilePassword [list trustFilePassword "password"]
# this long line puts the attributes for the object
   together from the variables and values
set ssl_def [list $ssl_clientAuthentication
  $ssl_enableCryptoHardwareSupport
  $ssl_keyFileFormat $ssl_keyFileName $ssl_keyFilePassword
  $ssl_securityLevel
  $ssl_trustFileFormat $ssl_trustFileName
  $ssl_trustFilePassword]
# defining the whole SSL object
set ssl_entry [list [list alias $ssl_alias] [list setting $ssl_def] ]
# remove existing dummy SSL entry
set sslList [$AdminConfig list SSLConfig]
$AdminConfig remove [lindex $sslList 0]
# creating the new entry
$AdminConfig create SSLConfig $security_root $ssl_entry repertoire
# setting variables using the new entry
set sslList [$AdminConfig list SSLConfig]
set default_ssl [list [list defaultSSLSettings [lindex $sslList 0]]]
```

```
# modifying the security object to use new entry
$AdminConfig modify $security_root $default_ssl
# saving the configuration
$AdminConfig save
```

b. Update the keyFileName, keyFilePassword, trustFileName, and trustFilePassword fields to specify the files and passwords created in Step 3a.

Note: If your password contains a special character such as $, you have to enter a preceding backslash. For example, pa$$w0rd should be entered as "pa\$\$w0rd"

c. Make certain that the following lines in the sample script are concatenated into one comment line (starting with '# this') and one command line (starting with 'set'):

```
# this long line puts the attributes for the object together
   from the variables and values
set ssl_def [list $ssl_clientAuthentication
$ssl_enableCryptoHardwareSupport
$ssl_keyFileFormat $ssl_keyFileName
$ssl_keyFilePassword $ssl_securityLevel
$ssl_trustFileFormat $ssl_trustFileName
$ssl_trustFilePassword]
```

4. Ensure wsadmin is configured to connect to the ISC_Portal by doing the following:

Note: If an existing WAS install was used instead of the ISC, that WAS node should be specified instead of DefaultNode in all the applicable paths below.

a. Check the serverindex.xml file in <isc_root>/AppServer/profiles/default/config/cells/DefaultNode/nodes/DefaultNode for the serverEntry with serverName=.ISC_Portal.

b. Under that serverEntry, find the port number specified under the endPointName="SOAP_CONNECTOR_ADDRESS" (it should be port 9425).

c. Edit the wsadmin.properties file in <isc_root>/AppServer/profiles/default/properties and change the com.ibm.ws.scripting.port setting to the port found in Step 4b.

5. Open a command prompt, change to <isc_root>/AppServer/bin, and run the command: wsadmin -f addSSLentry.jacl.

6. Edit the "ConfigService.properties" file. For AIX, Linux, Solaris, and Windows systems, the path is <isc_root>\PortalServer\shared\app\config\services\. Change the following parameters:

```
redirect.login.ssl = true
redirect.logout.ssl = true
```

7. Set the security constraints for the console URL. Perform the following steps:

a. Use an editor to open the file "web.xml." For AIX, Linux, Solaris, and Windows systems, the file is in the path <isc_root>\AppServer\profiles\default\config\cells\DefaultNode\applications\wps.ear\deployments\wps\wps.war\WEB-INF\web.xml.

b. In the web.xml file, change the <security-constraint> element for the console URL to use HTTPS as shown in the following example:

```
      <security-constraint id="SecurityConstraint_1">
         <web-resource-collection
    id="WebResourceCollection_1">
            <web-resource-name></web-resource-name>
            <url-pattern>/console/*</url-pattern>
```

```
                <http-method>DELETE</http-method>
                <http-method>GET</http-method>
                <http-method>POST</http-method>
                <http-method>PUT</http-method>
            </web-resource-collection>
            <auth-constraint id="AuthConstraint_1">
                <description></description>
                <role-name>All Role</role-name>
            </auth-constraint>
            <user-data-constraint id="UserDataConstraint_4">
                <transport-guarantee>CONFIDENTIAL
        </transport-guarantee> //
    replace NONE with CONFIDENTIAL
            </user-data-constraint>
        </security-constraint>
```

8. Restart the Integrated Solutions Console for these changes to take effect.
9. Test your changes by launching the regular HTTP portal home page in a Web browser. The session will be redirected to a secure connection for login (a browser security prompt appears during the redirection to accept the SSL certificate).

   Note: You will not need to click anything to verify SSL is being used, but a login is recommended to confirm changes.

Setting up the Lightweight Directory Access Protocol (LDAP) over SSL

To secure the data flowing between the LDAP server, the underlying portal, and the embedded application server, you must set up the LDAP server to use the Secure Sockets Layer (SSL) protocol. Be sure to set up the LDAP server and install the Integrated Solutions Console before you perform the procedure in this section.

Note: The following instructions apply to IBM Directory Server Version 4.1 and Version 5.1, which are the LDAP servers supported for this release of Integrated Solutions Console.

To set up LDAP over SSL, perform the following procedure:

1. To set up SSL in the IBM Directory Server, perform the following steps:
   a. Generate the key certificate for the IBM Directory Server using gsk6ikm. You can create a self-signed certificate if you are in a development environment or buy a certificate from a certificate authority. Create a CMS key database file for a self-signed certificate from IBM Directory Server. Remember the password.
   b. Within the CMS key database file, create a new self-signed certificate with X.509 Version 3 format and a 1024-bit key. Give the certificate a label. Remember the label.
   c. Extract the new self-signed certificate as a certificate file using **Base64-encoded ASCII data** as the data type. This will save the certificate with the file name that you specify and the file extension .arm (for example, LDAPcert.arm).
   d. If the LDAP server is not already configured, set up IBM Directory Server for LDAP over SSL. Use the CMS key database file that contains the self-signed certificate. See the IBM Directory Server documentation for instructions.
   e. Use your Web browser to open the administration interface for the IBM Directory Server. The URL is http://*ldap_host* /ldap, where *ldap_host* is the fully-qualified host name for the LDAP server.

f. For Directory Server Version 4.1 installations, perform the following steps:
   1) In the left navigation tree, click **Security > SSL > Settings**. The SSL settings window is displayed.
   2) Click **SSL On**.
   3) Click **Server Authentication**.
   4) Type an SSL port (such as 636).
   5) Click **Update**.
   6) Restart the LDAP server.
g. For Directory Server Version 5.1 installations, perform the following steps:
   1) Click **SSL Properties**. The SSL settings window is displayed.
   2) Click **SSL ON > Server Authentication**. Then type an SSL port (such as 636).
   3) Click **Apply**.
   4) Restart the LDAP server.

2. To set up the Web server, perform the following steps:
   a. Use the ikeyman utility to import LDAPcert.arm (or the certificate from the certificate authority) into the Web server key file (for example, DummyServerTrustfile.jks).
   b. Click the **Personal Certificates** tab.
   c. Click **Import**. The Import Key window displays.
   d. In the **File Name** field, specify LDAPcert.arm.
   e. Save the key store file.
3. To set up Integrated Solutions Console, perform the following steps:
   a. Create a self-signed certificate by performing the following steps:
      1) Use ikeyman to import ibmserver.arm (or the certificate from the certificate authority) into Integrated Solutions Console. Open the key file in one of the specified directories. Open <isc_root>/AppServer/java/jre/lib/security/cacerts. The default password for the file is changeit.
      2) Select **Signer Certificates** in the drop-down menu. Click **Add**.
      3) Select **Base64-encoded ASCII** data as the data type and then browse to the certificate file that you exported from IBM Directory Server.
      4) When prompted for a label for the new certificate, enter the same value that you specified for the label when you created the certificate.
      5) Save the cacerts file.
   b. Configure ISC to use LDAP over SSL by performing the following steps:
      1) Open the file wmm.xml. The file is in one of the following paths. The path is <isc_root>/PortalServer/shared/app/wmm/wmm.xml.
      2) Make the following changes to the wmm.xml file:
         a) Change the LDAP port from 389 to the port on which your LDAP server is listening for LDAP over SSL traffic. The default value is 636. In the **<ldapRepository>** element of the wmm.xml file, change the port number to the correct value for your installation:
            `ldapPort="636"`
         b) In the **<ldapRepository>** element of the wmm.xml file, add the following parameter setting:
            `java.naming.security.protocol="ssl"`
      3) Restart the Integrated Solutions Console.

4. To set up the Web server and LDAP, perform the following steps:
   a. Open the file security.xml. The file is in the following path. The path is <isc_root>/AppServer/config/cells/*node* /security.xml.
   b. Check the following line in the file to verify that SSL is enabled and that the port is set:

```
<userRegistries xmi:type="security:LDAPUserRegistry"
xmi:id="LDAPUserRegistry_1"
   serverId="iscadmin" serverPassword="{xor}KC8sPjsyNjE="
realm="domain.ibm.com:636"
   ignoreCase="true" type="IBM_DIRECTORY_SERVER" sslEnabled="true"
   sslConfig="DefaultNode/DefaultSSLSettings" baseDN="o=ibm"
   searchTimeout="120" reuseConnection="true">
```

   c. Check the single signon settings to ensure that SSL is enabled and that the domain name is set correctly:

```
<singleSignon xmi:id="SingleSignon_1" requiresSSL="true"
  domainName="domain.ibm.com" enabled="true" />
```

   d. Verify that global security is enabled and is using an LTPA token. These settings should have been enabled when you set up LDAP authentication.

```
<security:Security xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
  xmlns:orb.securityprotocol="http://www.ibm.com/websphere/appserver/schemas
/5.0/orb.securityprotocol.xmi"
  xmlns:security="http://www.ibm.com/websphere/appserver/schemas/5.0
/security.xmi"
  xmi:id="Security_1" useLocalSecurityServer="true"
useDomainQualifiedUserNames="false"
  enabled="true" cacheTimeout="600" issuePermissionWarning="true"
  activeProtocol="BOTH" enforceJava2Security="false"
activeAuthMechanism="LTPA_1"
  activeUserRegistry="LDAPUserRegistry" defaultSSLSettings="SSLConfig_1">
```

   e. To verify that the security level for the LDAP server is set to HIGH, perform the following steps:
      1) Open the file *your_ldap_root* \etc\slapd32.conf.
      2) Check the parameter ibm-slapdSSLCipherSpecs = 15360
5. Restart the Integrated Solutions Console.
6. Open https://*hostname* :9444/ibm/console, where *hostname* is the host name of the Integrated Solutions Console machine. A login prompt should be displayed.

Appendix D. 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 give 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 (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corp. 1993, 2005

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

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Programming Interface

This publication is intended to help the customer plan for and manage the IBM Tivoli Storage Manager server.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM Tivoli Storage Manager. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming interface information

End of Programming interface information

Trademarks

IBM, the IBM logo and the following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX
AS/400
DB2
DFS
DFSMS/MVS
DFSMShsm
DFSMSrmm
DPI
Enterprise Storage Server
ESCON
eServer
FICON
FlashCopy
HACMP
Informix
iSeries
Lotus
Lotus 1–2–3
Lotus Approach
Lotus Domino
Lotus Notes
Magstar
MVS
NetView
OpenEdition
OS/2
OS/390
OS/400
Passport Advantage
pSeries
RACF
Rational
Redbooks
RS/6000
S/390
SANergy
SecureWay
StorageSmart
SystemView
Tivoli
Tivoli Enterprise Console
Tivoli Management Environment
TotalStorage
TME
VTAM
WebSphere
z/OS
zSeries

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Glossary

The terms in this glossary are defined as they pertain to the IBM Tivoli Storage Manager library. If you do not find the term you need, refer to the IBM Software Glossary on the Web at this address: www.ibm.com/ibm/terminology/.

This glossary may include terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York 10036.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC2/SC1).

A

absolute mode. 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*. Contrast with *modified mode*.

access mode. An attribute of a storage pool or a storage volume that specifies whether the server can write to or read from the storage pool or storage volume. The access mode can be read/write, read-only, or unavailable. Volumes in primary storage pools can also have an access mode of destroyed. Volumes in copy storage pools can also have an access mode of offsite.

activate. To validate the contents of a policy set and then make it the active policy set.

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* and *policy set*.

active version. The most recent backup copy of a file stored by IBM Tivoli Storage Manager. 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 workstation. Contrast with *inactive version*.

activity log. A log that records normal activity messages generated by the server. These messages include information about server and client operations, such as the start time of sessions or device I/O errors. Each message includes a message ID, date and time stamp, and a text description. The number of days to retain messages in the activity log can be specified.

administrative client. A program that runs on a file server, workstation, or mainframe that administrators use to control and monitor the IBM Tivoli Storage Manager server. Contrast with *backup-archive client.*

administrative command schedule. A database record that describes the planned processing of an administrative command during a specific time period. See also *client schedule*.

administrative privilege class. See *privilege class*.

administrative session. A period of time in which an administrator user ID communicates with a server to perform administrative tasks. Contrast with *client node session*.

administrator. A user who has been registered to the server. Administrators can be authorized to one or more of the following administrative privilege classes: system, policy, storage, operator, or analyst. Administrators can use the administrative commands and queries allowed by their privileges.

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 packaged together. See *logical file* and *physical file*.

analyst privilege class. A privilege class that allows an administrator to reset statistics. See also *privilege class*.

application client. One of the IBM Tivoli Storage Manager data protection programs installed on a system to protect an application. The IBM Tivoli Storage Manager server provides backup services to an application client.

archive. To copy one or more files to a storage pool for long-term storage. Archived files can include descriptive information and can be retrieved by archive date, by file name, or by description. Contrast with *retrieve*.

archive copy. A file that was archived to server storage.

archive copy group. A policy object containing attributes that control the generation, destination, and

© Copyright IBM Corp. 1993, 2005

expiration of archived files. An archive copy group belongs to a management class.

archive retention grace period. The number of days that IBM Tivoli Storage Manager retains an archived file when the server is unable to rebind the file to an appropriate management class.

assigned capacity. The portion of available space that can be used to store database or recovery log information. See also *available space*.

association. (1) The defined relationship between a client node and a client schedule. An association identifies the name of a schedule, the name of the policy domain to which the schedule belongs, and the name of a client node that performs scheduled operations. (2) On a configuration manager, the defined relationship between a profile and an object such as a policy domain. Profile associations define the configuration information that will be distributed to a managed server when it subscribes to the profile.

audit. To check for logical inconsistencies between information that the server has and the actual condition of the system. IBM Tivoli Storage Manager can audit volumes, the database, libraries, and licenses. For example, when IBM Tivoli Storage Manager audits a volume, the server checks for inconsistencies between information about backed-up or archived files stored in the database and the actual data associated with each backup version or archive copy in server storage.

authentication. The process of checking a user's password before allowing that user access to the server. Authentication can be turned on or off by an administrator with system privilege.

authority. The right granted to a user to perform tasks with IBM Tivoli Storage Manager servers and clients. See also *privilege class*.

autochanger. A small, multislot tape device that automatically puts tape cartridges into tape drives. See also *library*.

available space. The amount of space, in megabytes, that is available to the database or the recovery log. This space can be used to extend the capacity of the database or the recovery log, or to provide sufficient free space before a volume is deleted from the database or the recovery log.

B

back up. To copy information to another location to ensure against loss of data. In IBM Tivoli Storage Manager, you can back up user files, the IBM Tivoli Storage Manager database, and storage pools. Contrast with *restore*. See also *database backup series* and *incremental backup*.

backup-archive client. A program that runs on a workstation or file server and provides a means for users to back up, archive, restore, and retrieve files. Contrast with *administrative client*.

backup copy group. A policy object containing attributes that control the generation, destination, and expiration of backup versions of files. A backup copy group belongs to a management class.

backup retention grace period. The number of days that IBM Tivoli 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 backup versions of files, generated for a backup-archive client.

backup version. A file that a user backed up to server storage. More than one backup version can exist in server storage, but only one backup version is the active version. See also *active version* and *inactive version*.

binding. The process of associating a file with a management class name. See *rebinding*.

buffer pool. Temporary space used by the server to hold database or recovery log pages. See *database buffer pool* and *recovery log buffer pool*.

C

cache. The process of leaving a duplicate copy on random access media when the server migrates a file to another storage pool in the hierarchy.

central scheduler. A function that allows an administrator to schedule client operations and administrative commands. The operations can be scheduled to occur periodically or on a specific date. See *client schedule* and *administrative command schedule*.

client. A program running on a PC, workstation, file server, LAN server, or mainframe that requests services of another program, called the server. The following types of clients can obtain services from an IBM Tivoli Storage Manager server: administrative client, application client, API client, backup-archive client, and HSM client (also known as Tivoli Storage Manager for Space Management).

client domain. The set of drives, file systems, or volumes that the user selects to back up or archive using the backup-archive client.

client migration. The process of copying a file from a client node to server storage and replacing the file with a stub file on the client node. The space management attributes in the management class control this migration. See also *space management*.

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 period of time in which a client node communicates with a server to perform backup, restore, archive, retrieve, migrate, or recall requests. Contrast with *administrative session*.

client options file. A file that a client can change, containing a set of processing options that identify the server, communication method, and options for backup, archive, hierarchical storage management, and scheduling. Also called the *dsm.opt* file.

client-polling scheduling mode. A client/server communication technique where the client queries the server for work. Contrast with *server-prompted scheduling mode*.

client schedule. A database record that describes the planned processing of a client operation during a specific time period. The client operation can be a backup, archive, restore, or retrieve operation, a client operating system command, or a macro. See also *administrative command schedule*.

client system options file. A file, used on UNIX clients, containing a set of processing options that identify the IBM Tivoli Storage Manager servers to be contacted for services. This file also specifies communication methods and options for backup, archive, hierarchical storage management, and scheduling. Also called the *dsm.sys* file. See also *client user options file*.

client user options file. A user-created file, used on UNIX clients, containing a set of processing options that identify the server, communication method, backup and archive options, space management options, and scheduling options. Also called the *dsm.opt* file. See also *client system options file*.

closed registration. A registration process in which only an administrator can register workstations as client nodes with the server. Contrast with *open registration*.

collocation. The process of keeping all data belonging to a single client file space, a single client node, or a group of client nodes on a minimal number of sequential-access volumes within a storage pool. Collocation can reduce the number of volumes that must be accessed when a large amount of data must be restored.

collocation group. A user-defined group of client nodes whose data is stored on a minimal number of volumes through the process of collocation.

compression. The process of saving storage space by eliminating empty fields or unnecessary data in a file. In IBM Tivoli Storage Manager, compression can occur at a workstation before files are backed up or archived to server storage. On some types of tape drives, hardware compression can be used.

configuration manager. One IBM Tivoli Storage Manager server that distributes configuration information to other IBM Tivoli Storage Manager servers (called managed servers) via profiles. Configuration information can include policy and schedules. See *managed server* and *profile*.

copy group. A policy object whose attributes control how backup versions or archive copies are generated, where backup versions or archive copies are initially located, and when backup versions or archive copies expire. A copy group belongs to a management class. See also *archive copy group*, *backup copy group*, *backup version*, and *management class*.

copy storage pool. A named set of volumes that contains copies of files that reside in primary storage pools. Copy storage pools are used only to back up the data stored in primary storage pools. A copy storage pool cannot be a destination for a backup copy group, an archive copy group, or a management class (for space-managed files). See *primary storage pool* and *destination*.

D

damaged file. A physical file for which IBM Tivoli Storage Manager has detected read errors.

database. A collection of information about all objects managed by the server, including policy management objects, users and administrators, and client nodes.

database backup series. One full backup of the database, plus up to 32 incremental backups made since that full backup. Each full backup that is run starts a new database backup series. A number identifies each backup series.

database backup trigger. A set of criteria that defines when and how database backups are run automatically. The criteria determine how often the backup is run, whether the backup is a full or incremental backup, and where the backup is stored.

database buffer pool. Storage that is used as a cache to allow database pages to remain in memory for long periods of time, so that the server can make continuous updates to pages without requiring input or output (I/O) operations from external storage.

database snapshot. A complete backup of the entire IBM Tivoli Storage Manager database to media that can be taken off-site. When a database snapshot is created, the current database backup series is not interrupted. A database snapshot cannot have incremental database backups associated with it. See also *database backup series*. Contrast with *full backup*.

data mover. A device, defined to IBM Tivoli Storage Manager, that moves data on behalf of the server. A NAS file server can be a data mover.

default management class. A management class assigned to a policy set that the server uses to manage backed-up or archived files when a user does not specify a management class for a file.

desktop client. The group of backup-archive clients supported by IBM Tivoli Storage Manager that includes clients on Windows, Apple, and Novell NetWare operating systems.

destination. A copy group or management class attribute that specifies the primary storage pool to which a client file will be backed up, archived, or migrated.

device class. A named set of characteristics 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. A file that contains information about defined device classes, and, on some IBM Tivoli Storage Manager servers, defined libraries and drives. The file can be created by using an IBM Tivoli Storage Manager administrative command or by using an option in the server options file. The information is a copy of the device configuration information in the IBM Tivoli Storage Manager database.

disaster recovery manager (DRM). A function in IBM Tivoli Storage Manager that assists in preparing and later using a disaster recovery plan file for the IBM Tivoli Storage Manager server.

disaster recovery plan. A file created by disaster recovery manager (DRM) that contains information about how to recover computer systems if a disaster occurs and scripts that can be run to perform some recovery tasks. The file includes information about the software and hardware used by the IBM Tivoli Storage Manager server, and the location of recovery media.

domain. See *policy domain* or *client domain*.

DRM. A short name for disaster recovery manager.

dsm.opt file. See *client options file* and *client user options file*.

dsmserv.opt. See *server options file*.

dsm.sys file. See *client system options file*.

dynamic. A value for serialization that specifies that IBM Tivoli Storage Manager accepts the first attempt to back up or archive a file regardless of whether the file is modified during the backup or archive process. See also *serialization*. Contrast with *shared dynamic*, *shared static*, and *static*.

E

enterprise configuration. A method of setting up IBM Tivoli Storage Manager servers so that the administrator can distribute the configuration of one of the servers to the other servers, using server-to-server communication. See *configuration manager*, *managed server*, *profile*, and *subscription*.

enterprise logging. The sending of events from IBM Tivoli Storage Manager servers to a designated event server. The event server routes the events to designated receivers, such as to a user exit. See also *event*.

estimated capacity. The available space, in megabytes, of a storage pool.

event. (1) An administrative command or a client operation that is scheduled to be run using IBM Tivoli Storage Manager scheduling. (2) A message that an IBM Tivoli Storage Manager server or client issues. Messages can be logged using IBM Tivoli Storage Manager event logging.

event record. A database record that describes actual status and results for events.

event server. A server to which other servers can send events for logging. The event server routes the events to any receivers that are enabled for the sending server's events.

exclude. To identify files that you do not want to include in a specific client operation, such as backup or archive. You identify the files in an include-exclude list.

exclude-include list. See *include-exclude list*.

expiration. The process by which files are identified for deletion because their expiration date or retention period has passed. Backed-up or archived files are marked expired by IBM Tivoli Storage Manager based on the criteria defined in the backup or archive copy group.

expiration date. On some IBM Tivoli Storage Manager servers, a device class attribute used to notify tape management systems of the date when IBM Tivoli Storage Manager no longer needs a tape volume. The date is placed in the tape label so that the tape management system does not overwrite the information on the tape volume before the expiration date.

export. To copy administrator definitions, client node definitions, policy definitions, server control

information, or file data to external media, or directly to another server. Used to move or copy information between servers.

extend. To increase the portion of available space that can be used to store database or recovery log information. Contrast with *reduce*.

F

file space. A logical space in IBM Tivoli Storage Manager server storage that contains a group of files backed up or archived by a client. For clients on Windows systems, a file space contains files from a logical partition that is identified by a volume label. For clients on UNIX systems, a file space contains files from the same file system, or the part of a file system that stems from a virtual mount point. Clients can restore, retrieve, or delete their file spaces from IBM Tivoli Storage Manager server storage. IBM Tivoli Storage Manager does not necessarily store all the files belonging to a single file space together, but can identify all the files in server storage that belong to a single file space.

file space ID (FSID). A unique numeric identifier that the server assigns to a file space when it is stored in server storage.

frequency. A copy group attribute that specifies the minimum interval, in days, between incremental backups.

FSID. See *file space ID*.

full backup. The process of backing up the entire server database. A full backup begins a new database backup series. See also *database backup series* and *incremental backup*. Contrast with *database snapshot*.

fuzzy copy. A backup version or archive copy of a file that might not accurately reflect the original content of the file because IBM Tivoli Storage Manager backed up or archived the file while the file was being modified. See also *backup version* and *archive copy*.

H

hierarchical storage management (HSM) client. The Tivoli Storage Manager for Space Management program that runs on workstations to allow users to maintain free space on their workstations by migrating and recalling files to and from IBM Tivoli Storage Manager storage. Synonymous with *space manager client*.

high migration threshold. A percentage of the storage pool capacity that identifies when the server can start migrating files to the next available storage pool in the hierarchy. Contrast with *low migration threshold*. See *server migration*.

HSM client. Hierarchical storage management client. Also known as the space manager client.

I

IBM Tivoli Storage Manager command script. A sequence of IBM Tivoli Storage Manager administrative commands that are stored in the database of the IBM Tivoli Storage Manager server. You can run the script from any interface to the server. The script can include substitution for command parameters and conditional logic.

image backup. A backup of a full file system or raw logical volume as a single object.

import. The process of copying exported administrator definitions, client node definitions, policy definitions, server control information or file data from external media to a target server. A subset of information can be imported to a target server from the external media. Used to move or copy information between servers. See *export*.

inactive version. A backup version of a file that is either not the most recent backup version, or that is a backup version of a file that no longer exists on the client system. Inactive backup versions are eligible for expiration processing according to the management class assigned to the file. Contrast with *active version*.

include-exclude file. A file containing statements that IBM Tivoli Storage Manager uses to determine whether to include certain files in specific client operations, and to determine the associated management classes to use for backup, archive, and space management. See *include-exclude list*.

include-exclude list. A group of include and exclude option statements that IBM Tivoli Storage Manager uses. The exclude options identify files that are not to be included in specific client operations such as backup or space management. The include options identify files that are exempt from the exclusion rules. The include options can also assign a management class to a file or group of files for backup, archive, or space management services. The include-exclude list for a client may include option statements from the client options file, from separate include-exclude files, and from a client option set on the server.

incremental backup. (1) The process of backing up files or directories that are new or have changed since the last incremental backup. See also *selective backup*. (2) The process of copying only the pages in the database that are new or changed since the last full or incremental backup of the database. Contrast with *full backup*. See also *database backup series*.

L

LAN-free data movement. The direct movement of client data between a client machine and a storage device on a SAN, rather than on the LAN.

library. (1) A repository for demountable recorded media, such as magnetic tapes. (2) For IBM Tivoli Storage Manager, a collection of one or more drives, and possibly robotic devices (depending on the library type), which can be used to access storage volumes. (3) In the AS/400® system, a system object that serves as a directory to other objects. A library groups related objects, and allows the user to find objects by name.

library client. An IBM Tivoli Storage Manager server that uses server-to-server communication to access a library that is managed by another IBM Tivoli Storage Manager server. See also *library manager*.

library manager. An IBM Tivoli Storage Manager server that controls device operations when multiple IBM Tivoli Storage Manager servers share a storage device. The device operations include mount, dismount, volume ownership, and library inventory. See also *library client*.

logical file. A file stored in one or more server storage pools, either by itself or as part of an aggregate. See also *aggregate* and *physical file*.

logical occupancy. The space used by logical files in a storage pool. Because logical occupancy does not include the unused space created when logical files are deleted from aggregates, it may be less than *physical occupancy*. See also *physical file* and *logical file*.

logical volume. (1) A portion of a physical volume that contains a file system. (2) For the IBM Tivoli Storage Manager server, the combined space on all volumes for either the database or the recovery log. The database is one logical volume, and the recovery log is one logical volume.

low migration threshold. A percentage of the storage pool capacity that specifies when the server can stop the migration of files to the next storage pool. Contrast with *high migration threshold*. See *server migration*.

M

macro file. A file that contains one or more IBM Tivoli Storage Manager administrative commands, which can be run only from an administrative client by using the MACRO command. Contrast with *IBM Tivoli Storage Manager command script*.

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 associated with that profile become managed objects in the database of the managed server. In general, a managed object cannot be modified locally on the managed server. Objects can include policy, schedules, client options sets, server scripts, administrator registrations, and server and server group definitions.

managed server. An IBM Tivoli Storage Manager server that receives configuration information from a configuration manager via subscription to one or more profiles. Configuration information can include definitions of objects such as policy and schedules. See *configuration manager*, *subscription*, and *profile*.

managed system. A client or server that requests services from the IBM Tivoli Storage Manager server.

management class. A policy object that users can bind to each file to specify how the server manages the file. The management class can contain a backup copy group, an archive copy group, and space management attributes. The copy groups determine how the server manages backup versions or archive copies of the file. The space management attributes determine whether the file is eligible to be migrated by the space manager client to server storage and under what conditions the file is migrated. See also *copy group*, *space manager client*, *binding*, and *rebinding*.

maximum extension. Specifies the maximum amount of storage space, in megabytes, that you can extend the database or the recovery log.

maximum reduction. Specifies the maximum amount of storage space, in megabytes, that you can reduce the database or the recovery log.

maximum utilization. The highest percentage of assigned capacity used by the database or the recovery log.

migrate. To move data from one storage location to another. See also *client migration* and *server migration*.

mirroring. The process of writing the same data to multiple disks at the same time. The mirroring of data protects 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 *modified* and *absolute*.

modified mode. 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 changed if the date, size, owner, or permissions of the file has changed. See also *mode*. Contrast with *absolute mode*.

mount. To place a data medium (such as a tape cartridge) on a drive in a position to operate.

mount limit. A device class attribute that specifies 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 *mount point*.

mount point. A logical drive through which volumes in a sequential access device class are accessed. For removable media device types, such as tape, a mount point is a logical drive associated with a physical drive. For the file device type, a mount point is a logical drive associated with an I/O stream. The number of mount points for a device class is defined by the value of the mount limit attribute for that device class. See *mount limit*.

mount retention period. A device class attribute that specifies 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. A device class attribute that specifies the maximum number of minutes that the server waits for a sequential access volume mount request to be satisfied before canceling the request.

N

NAS. Network-attached storage.

NAS node. A client node that is a network-attached storage (NAS) file server. Data for the NAS node is transferred by the NAS file server, which is controlled by the IBM Tivoli Storage Manager server. The Tivoli Storage Manager server uses the network data management protocol (NDMP) to direct the NAS file server in its backup and restore operations. A NAS node is also called a NAS file server node.

native format. A format of data that is written to a storage pool directly by the IBM Tivoli Storage Manager server. Contrast with *non-native data format*.

NDMP. Network Data Management Protocol.

network-attached storage (NAS) file server. A dedicated storage device with an operating system that is optimized for file-serving functions. In IBM Tivoli Storage Manager, a NAS file server can have the characteristics of both a node and a data mover. See also *data mover* and *NAS node*.

Network Data Management Protocol (NDMP). An industry-standard protocol that allows a network storage-management application (such as IBM Tivoli Storage Manager) to control the backup and recovery of an NDMP-compliant file server, without installing third-party software on that file server.

node. (1) A workstation or file server that is registered with an IBM Tivoli Storage Manager server to receive its services. See also *client node* and *NAS node*. (2) In a Microsoft cluster configuration, one of the computer systems that make up the cluster.

node privilege class. A privilege class that allows an administrator 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 written to a storage pool that is different from the format that the server uses for basic LAN-based operations. The data is written by a data mover instead of the server. Storage pools with data written in a non-native format may not support some server operations, such as audit of a volume. The NETAPPDUMP data format for NAS node backups is an example of a non-native data format.

O

open registration. A registration process in which any users can register their own workstations as client nodes with the server. Contrast with *closed registration*.

operator privilege class. A privilege class that allows an administrator to issue commands that disable or halt the server, enable the server, cancel server processes, and manage removable media. See also *privilege class*.

P

page. A unit of space allocation within IBM Tivoli Storage Manager database volumes.

path. An IBM Tivoli Storage Manager object that defines a one-to-one relationship between a source and a destination. Using the path, the source accesses the destination. Data may flow from the source to the destination, and back. An example of a source is a data mover (such as a NAS file server), and an example of a destination is a tape drive.

physical file. A file stored in one or more storage pools, consisting of either a single logical file, or a group of logical files packaged together (an aggregate). See also *aggregate* and *logical file*.

physical occupancy. The amount of space used by physical files in a storage pool. This space includes the unused space created when logical files are deleted from aggregates. See also *physical file*, *logical file*, and *logical occupancy*.

policy domain. A grouping of policy users with one or more policy sets, which manage data or storage resources for the users. In IBM Tivoli Storage Manager, the users are client nodes. See *policy set*, *management class*, and *copy group*.

policy privilege class. A privilege class that allows an administrator 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 policy object that contains a group of management classes that exist for a policy domain. Several policy sets can exist within a policy domain but only one policy set is active at one time. See *management class* and *active policy set*.

premigration. For a space manager client, the process of copying files that are eligible for migration to server storage, while leaving the original file intact on the local system.

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 HSM client nodes. You can back up a primary storage pool to a copy storage pool. See *destination* and *copy storage pool*.

privilege class. A level of authority granted to an administrator. The privilege class determines which administrative tasks the administrator can perform. For example, an administrator with system privilege class can perform any administrative task. Also called administrative privilege class. See also *system privilege class*, *policy privilege class*, *storage privilege class*, *operator privilege class*, *analyst privilege class*, and *node 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 administrators, policy, client schedules, client option sets, administrative schedules, IBM Tivoli Storage Manager command scripts, server definitions, and server group definitions. See *configuration manager* and *managed server*.

R

randomization. The process of distributing schedule start times for different clients within a specified percentage of the schedule's startup window.

rebinding. The process of associating a backed-up file with a new management class name. For example, rebinding occurs when the management class associated with a file is deleted. See *binding*.

recall. To access files that were migrated from workstations to server storage by using the space manager client. Contrast with *migrate*.

receiver. A server repository that contains a log of server messages and client messages as events. For example, a receiver can be a file exit, a user exit, or the IBM Tivoli Storage Manager server console and activity log. See also *event*.

reclamation. A process of consolidating the remaining data from many sequential access volumes onto fewer new sequential access volumes.

reclamation threshold. The percentage of reclaimable 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. The percentage is set for a storage pool.

recovery log. A log of updates that are about to be written to the database. The log can be used to recover from system and media failures.

recovery log buffer pool. Storage that the server uses to hold new transaction records until they can be written to the recovery log.

reduce. To free up enough space from the database or the recovery log, such that you can delete a volume. Contrast with *extend*.

register. (1) To define a client node or administrator who can access the server. See *registration*. (2) To specify licenses that have been purchased for the server.

registration. The process of identifying a client node or administrator to the server.

restore. To copy information from its backup location to the active storage location for use. In IBM Tivoli Storage Manager, you can restore the server database, storage pools, storage pool volumes, and users' backed-up files. The backup version of a file in the storage pool is not affected by the restore operation. Contrast with *backup*.

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.

retention period. On an MVS™ server, a device class attribute that specifies how long files are retained on sequential access media. When used, IBM Tivoli Storage Manager passes this information to the MVS operating system to ensure that other tape management systems do not overwrite tape volumes that contain retained data.

retrieve. To copy archived information from the storage pool to the client node for use. The retrieve operation does not affect the archive copy in the storage pool. Contrast with *archive*.

rollback. To remove changes that were made to database files since the last commit point.

S

schedule. A database record that describes scheduled client operations or administrative commands. See *administrative command schedule* and *client schedule*.

scheduling mode. The method of interaction between a server and a client for running scheduled operations on the client. IBM Tivoli Storage Manager supports two scheduling modes for client operations: *client-polling* and *server-prompted*.

scratch volume. A labeled volume that is either blank or contains no valid data, that is not currently defined to IBM Tivoli Storage Manager, and that is available for use.

script. See *IBM Tivoli Storage Manager command script*.

selective backup. The process of backing up selected files or directories from a client domain. See also *incremental backup*.

serialization. The process of handling files that are modified during backup or archive processing. See *static*, *dynamic*, *shared static*, and *shared dynamic*.

server migration. The process of moving data from one storage pool to the next storage pool defined in the hierarchy, based on the migration thresholds defined for the storage pools. See also *high migration threshold* and *low migration threshold*.

server options file. A file that contains settings that control various server operations. These settings, or options, affect such things as communications, devices, and performance.

server-prompted scheduling mode. A client/server communication technique where the server contacts the client when a scheduled operation needs to be done. Contrast with *client-polling scheduling mode*.

server storage. The primary and copy storage pools used by the server to store users' files: backup versions, archive copies, and files migrated from space manager client nodes (space-managed files). See *primary storage pool*, *copy storage pool*, *storage pool volume*, and *volume*.

session resource usage. The amount of wait time, CPU time, and space used or retrieved during a client session.

shared dynamic. A value for serialization that specifies that a file must not be backed up or archived if it is being modified during the operation. IBM Tivoli Storage Manager retries the backup or archive operation a number of times; if the file is being modified during each attempt, IBM Tivoli Storage Manager will back up or archive the file on its last try. See also *serialization*. Contrast with *dynamic*, *shared static*, and *static*.

shared library. A library device that is shared among multiple IBM Tivoli Storage Manager servers.

shared static. A value for serialization that specifies that a file must not be backed up or archived if it is being modified during the operation. IBM Tivoli Storage Manager retries the backup or archive operation a number of times; if the file is being modified during each attempt, IBM Tivoli Storage Manager will not back up or archive the file. See also *serialization*. Contrast with *dynamic*, *shared dynamic*, and *static*.

snapshot. See *database snapshot*.

source server. A server that can send data, in the form of *virtual volumes*, to another server. Contrast with *target server*.

space-managed file. A file that is migrated from a client node by the space manager client (HSM client). The space manager client recalls the file to the client node on demand.

space management. The process of keeping sufficient free storage space available on a client node by migrating files to server storage. The files are migrated based on criteria defined in management classes to which the files are bound, and the include-exclude list. Synonymous with *hierarchical storage management*. See also *migration*.

space manager client. The Tivoli Storage Manager for Space Management program that enables users to maintain free space on their workstations by migrating and recalling files to and from server storage. Also called *hierarchical storage management (HSM) client*.

startup window. A time period during which a schedule must be initiated.

static. A value for serialization that specifies that a file must not be backed up or archived if it is being modified during the operation. IBM Tivoli Storage Manager does not retry the operation. See also *serialization*. Contrast with *dynamic*, *shared dynamic*, and *shared static*.

storage agent. A program that enables IBM Tivoli Storage Manager to back up and restore client data directly to and from SAN-attached storage.

storage hierarchy. A logical ordering of primary storage pools, as defined by an administrator. The ordering is usually based on the speed and capacity of the devices that the storage pools use. In IBM Tivoli Storage Manager, the storage hierarchy is defined by identifying the *next* storage pool in a storage pool definition. See *storage pool*.

storage pool. A named set of storage volumes that is the destination that the IBM Tivoli Storage Manager server uses to store client data. A storage pool stores

backup versions, archive copies, and files that are migrated from space manager client nodes. You back up a primary storage pool to a copy storage pool. See *primary storage pool* and *copy storage pool*.

storage pool volume. A volume that has been assigned to a storage pool. See *volume*, *copy storage pool*, and *primary storage pool*.

storage privilege class. A privilege class that allows an administrator to control how storage resources for the server are allocated and used, such as monitoring the database, the recovery log, and server storage. Authority can be restricted to certain storage pools. See also *privilege class*.

stub file. A file that replaces the original file on a client node when the file is migrated from the client node to server storage by Tivoli Storage Manager for Space Management.

subscription. In a Tivoli environment, the process of identifying the subscribers that the profiles are distributed to. For IBM Tivoli Storage Manager, this is the process by which a managed server requests that it receive configuration information associated with a particular profile on a configuration manager. See *managed server*, *configuration manager*, and *profile*.

system privilege class. A privilege class that allows an administrator to issue all server commands. See also *privilege class*.

T

tape library. A term used to refer to a collection of drives and tape cartridges. The tape library may be an automated device that performs tape cartridge mounts and demounts without operator intervention.

tape volume prefix. A device class attribute that is the high-level-qualifier of the file name or the data set name in the standard tape label.

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 being operated on.

target server. A server that can receive data sent from another server. Contrast with *source server*. See also *virtual volumes*.

U

UCS-2. An ISO/IEC 10646 encoding form, Universal Character Set coded in 2 octets. The IBM Tivoli Storage Manager client on Windows uses the UCS-2 code page when the client is enabled for Unicode.

Unicode Standard. 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. It can also support many classical and historical texts and is continually being expanded. The Unicode Standard is compatible with ISO/IEC 10646. For more information, see www.unicode.org.

UTF-8. Unicode transformation format - 8. A byte-oriented encoding form specified by the Unicode Standard.

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. A virtual file space is used to back up the directory as a file space in IBM Tivoli Storage Manager server storage.

virtual volume. An archive file on a *target server* that represents a sequential media volume to a *source server*.

volume. The basic unit of storage for the IBM Tivoli Storage Manager database, recovery log, and storage pools. A volume can be an LVM logical volume, a standard file system file, a tape cartridge, or an optical cartridge. Each volume is identified by a unique volume identifier. See *database volume*, *scratch volume*, and *storage pool volume*.

volume history file. A file that contains information about: volumes used for database backups and database dumps; volumes used for export of administrator, node, policy, or server data; and sequential access storage pool volumes that have been added, reused, or deleted. The information is a copy of the same types of volume information in the IBM Tivoli Storage Manager database.

Index

Special characters

$$CONFIG_MANAGER$$ 592

Numerics

3480 tape drive
  cleaner cartridge 191
  device support 91
3490 tape drive
  cleaner cartridge 191
  device support 91
3494 automated library device 64, 111
3494SHARED server option 102
3570 tape drive
  defining device class 80
  device support 91
3590 tape drive
  ASSISTVCRRECOVERY server option 102
  defining device class 80, 197, 198
  device support 112
3592 tape drive
  as element of storage hierarchy 233
  defining device class 80
  DEVICETYPE parameter 172
  enabling for WORM media 175
3592 WORM media 174

A

absolute mode, description of 383
ACCEPT DATE command 460
access authority, client 327
access mode, volume
  changing 230
  description 231
  determining for storage pool 221, 301
access, managing 348, 350
accounting record
  description of 541
  monitoring 541
accounting variable 541
ACSLS (Automated Cartridge System Library Software)
 StorageTek library
  configuring 123
  description 64
  Tivoli Storage Manager server options for 102
ACTIVATE POLICYSET command 389
ACTIVE policy set
  creating 379, 389
  replacing 360
activity log
  description of 523
  monitoring 523
  querying 524
  setting retention period 525
  setting size limit 525
  setting the retention period 525
administration center
  using 36
Administration Center
  commands not supported 55
  features not supported 55
administrative client
  description of 3
  viewing information after IMPORT or EXPORT 633
administrative commands
  ACCEPT DATE 464
  ASSIGN DEFMGMTCLASS 360, 389
  AUDIT LIBVOLUME 181
  AUDIT LICENSE 459
  AUDIT VOLUME 676
  BACKUP DB 660
  BACKUP DEVCONFIG 658
  BACKUP NODE 160, 161
  BACKUP STGPOOL 648
  BACKUP VOLHISTORY 657
  BEGIN EVENTLOGGING 527
  CANCEL PROCESS 466
  CANCEL RESTORE 347
  CANCEL SESSION 344
  CHECKIN LIBVOLUME 171
  CHECKOUT LIBVOLUME 180
  CLEAN DRIVE 189
  COMMIT 487
  CONVERT ARCHIVE 429, 431
  COPY CLOPTSET 342
  COPY DOMAIN 379
  COPY POLICYSET 380
  COPY SCHEDULE 442
  COPY SCRIPT 482
  COPY SERVERGROUP 600
  DEFINE ASSOCIATION 30, 435
  DEFINE BACKUPSET 415
  DEFINE CLIENTACTION 452
  DEFINE CLIENTOPT 452
  DEFINE CLOPTSET 340
  DEFINE COPYGROUP 381, 386, 387
  DEFINE DATAMOVER 139, 158
  DEFINE DBBACKUPTRIGGER 653, 655
  DEFINE DBVOLUME 499
  DEFINE DEVCLASS
    FILE device classes 204
    LTO device classes 207
    REMOVEABLEFILE device classes 203
    SERVER device classes 208
    tape device classes 198
    VOLSAFE device classes 210
  DEFINE DOMAIN 378
  DEFINE DRIVE 138
  DEFINE GRPMEMBER 599
  DEFINE LIBRARY 63, 64, 137
  DEFINE LOGCOPY 669
  DEFINE LOGVOLUME 499
  DEFINE MACHINE 693
  DEFINE MACHNODEASSOCIATION 694
  DEFINE PATH 140, 158
  DEFINE POLICYSET 379, 380
  DEFINE PROFASSOCIATION 580, 581
  DEFINE PROFILE 580
  DEFINE RECMEDMACHASSOCIATION 695

© Copyright IBM Corp. 1993, 2005

administrative commands *(continued)*
DEFINE RECOVERYMEDIA 695
DEFINE SCHEDULE 473
DEFINE SCRIPT 478
DEFINE SERVER 569, 597, 602
DEFINE SERVERGROUP 599
DEFINE SPACETRIGGER 497
DEFINE STGPOOL 224, 225, 234
DEFINE SUBSCRIPTION 590
DEFINE VIRTUALFSMAPPING 162
DEFINE VOLUME 68, 69, 229
DELETE ASSOCIATION 444
DELETE BACKUPSET 418
DELETE COPYGROUP 408
DELETE DBBACKUPTRIGGER 655
DELETE DBVOLUME 502
DELETE DEVCLASS 212
DELETE DOMAIN 409
DELETE DRIVE 193
DELETE EVENT 446
DELETE GRPMEMBER 601
DELETE LIBRARY 187
DELETE LOGVOLUME 502
DELETE MGMTCLASS 408
DELETE POLICYSET 409
DELETE PROFASSOCIATION 585
DELETE PROFILE 586
DELETE SCHEDULE 442
DELETE SCRIPT 484
DELETE SERVER 574
DELETE SERVERGROUP 600
DELETE STGPOOL 303
DELETE SUBSCRIBER 596
DELETE SUBSCRIPTION 586, 592
DELETE VOLHISTORY 656, 657
DELETE VOLUME 305
DISABLE EVENTS 527
DISABLE SESSIONS 346
DISMOUNT VOLUME 186
DSMFMT 226, 645
DSMSERV FORMAT 663
ENABLE EVENTS 527
ENABLE SESSIONS 346
END EVENTLOGGING 527
EXPIRE INVENTORY 32, 391
EXPORT ADMIN 609
EXPORT NODE 619
EXPORT POLICY 619
EXPORT SERVER 619
EXTEND DB 500
EXTEND LOG 653
GENERATE BACKUPSET 413
GRANT AUTHORITY 348
HALT 462
HELP 469
IMPORT 632, 633
IMPORT ADMIN 621
IMPORT NODE 621, 629
IMPORT POLICY 621
IMPORT SERVER 621, 629
LABEL LIBVOLUME 106, 134
LOCK ADMIN 352
LOCK NODE 320
LOCK PROFILE 584, 585
MOVE DATA 294
MOVE NODEDATA 298
NOTIFY SUBSCRIBERS 584, 585

administrative commands *(continued)*
PING SERVER 601
PREPARE 696
QUERY ACTLOG 524
QUERY BACKUPSET 416
QUERY BACKUPSETCONTENTS 417
QUERY CONTENT 284
QUERY COPYGROUP 406, 627
QUERY DB 501
QUERY DBBACKUPTRIGGER 655
QUERY DBVOLUME 501, 647
QUERY DEVCLASS 211
QUERY DOMAIN 407
QUERY DRIVE 188
QUERY DRMSTATUS 688
QUERY ENABLED 540
QUERY EVENT 32, 436
QUERY FILESPACE 338
QUERY LIBRARY 186
QUERY LICENSE 459
QUERY LOG 506
QUERY LOGVOLUME 501, 647
QUERY MGMTCLASS 406
QUERY MOUNT 186
QUERY NODE 323
QUERY NODEDATA 292
QUERY OCCUPANCY
backed-up, archived, and space-managed files 293
client file spaces 291
client nodes 291
device classes 292
storage pools 292
QUERY OPTION 516
QUERY POLICYSET 407
QUERY PROCESS 296
QUERY REQUEST 184
QUERY RESTORE 347
QUERY RPFCONTENT 697
QUERY RPFILE 697
QUERY SCHEDULE 436
QUERY SCRIPT 482
QUERY SERVERGROUP 600
QUERY STGPOOL 279, 287, 625
QUERY SUBSCRIPTION 591
QUERY SYSTEM 517
QUERY VOLHISTORY 656
QUERY VOLUME 281, 282, 297
RECONCILE VOLUMES 606
REDUCE DB 502
REDUCE LOG 502
REGISTER ADMIN 351
REGISTER LICENSE 459
REMOVE ADMIN 352
REMOVE NODE 321
RENAME ADMIN 351
RENAME FILESPACE 631
RENAME NODE 320
RENAME SCRIPT 483
RENAME SERVERGROUP 600
RENAME STGPOOL 300
RESET BUFPOOL 503
RESET DBMAXUTILIZATION 493, 495
RESET LOGCONSUMPTION 653
RESET LOGMAXUTILIZATION 493, 495
RESTORE NODE 160, 161
RESTORE STGPOOL 649, 681
RESTORE VOLUME 683

administrative commands *(continued)*
ROLLBACK 488
RUN 484
SELECT 518
SET ACCOUNTING 541
SET AUTHENTICATION 355
SET CLIENTACTDURATION 452
SET CONFIGMANAGER 577, 579
SET CONFIGREFRESH 591
SET CONTEXTMESSAGING 527
SET CROSSDEFINE 570, 573
SET DRMCHECKLABEL 691
SET DRMCOPYSTGPOOL 689
SET DRMCOURIERNAME 691
SET DRMDBBACKUPRXPIREDAYS 691
SET DRMFILEPROCESS 692
SET DRMINSTPREFIX 689
SET DRMNOTMOUNTABLE 691
SET DRMPLANPOSTFIX 689
SET DRMPLANPREFIX 690
SET DRMPRIMSTGPOOL 688
SET DRMRPFEXPIREDAYS 697
SET DRMVAULTNAME 692
SET EVENTRETENTION 446, 477
SET INVALIDPWLIMIT 355
SET LICENSEAUDITPERIOD 459
SET LOGMODE 656
SET MAXCMDRETRIES 452
SET MAXSCHEDSESSIONS 449
SET MINPWLENGTH 355
SET PASSEXP 354
SET QUERYSCHEDPERIOD 451
SET RANDOMIZE 450
SET REGISTRATION 311
SET RETRYPERIOD 452
SET SCHEDMODES 447
SET SERVERHLADDRESS 570, 573
SET SERVERLLADDRESS 570, 573
SET SERVERNAME 516, 569, 573
SET SERVERPASSWORD 569, 570, 573
SET SUBFILE 418
SET SUMMARYRETENTION 522
SETOPT 469
UNDO ARCHCONVERSION 429, 431
UNLOCK PROFILE 584, 585
UPDATE ADMIN 351
UPDATE ARCHIVE 431
UPDATE BACKUPSET 416
UPDATE CLIENTOPT 342
UPDATE CLOPTSET 343
UPDATE COPYGROUP 381, 387
UPDATE DBBACKUPTRIGGER 656
UPDATE DOMAIN 379
UPDATE DRIVE 188
UPDATE LIBRARY 186
UPDATE LIBVOLUME 69, 179
UPDATE MGMTCLASS 380
UPDATE NODE 320
UPDATE POLICYSET 379
UPDATE RECOVERYMEDIA 695
UPDATE SCHEDULE 473
UPDATE SCRIPT 481
UPDATE SERVER 574
UPDATE SERVERGROUP 600
UPDATE VOLUME 229
VALIDATE LANFREE 136
administrative privilege class
description 348
granting authority 348
reducing 354
revoking all 354
administrative user ID
creating automatically 327
description of 310
preventing automatic creation of 328
administrative Web interface
description 19
limitation of browser for script definitions 478
administrator
authorizing to manage a policy domain 348
locking 352
managing registration 457
querying 352
registering 351
removing 352
renaming 351
restrictions when registering 351
unlocking 352
updating 351
viewing information about 352
administrators
adding 39
modifying 39
viewing 39
aggregates
controlling the size of 234
estimating size 284
RECONSTRUCT parameter 297, 422
reconstructing 267, 274, 297
viewing information about 285, 286, 291
analyst privilege class
revoking 354
ANR8914I message 192
ANR9999D message 527
application client
adding node for 310
description 4
policy for 400
application program interface (API)
client, registering 313
compression option 313
deletion option 313
description of 3
registering to server 313
simultaneous write, version support for 250
archive
allowing while file changes 387
backup set, uses for 9, 13
conversion of a node 429
determining storage usage 293
directory 426, 427
instant 9, 13
package 426
performance considerations 427
policy, defining 378
policy, introduction 26
process description 375
storage usage, minimizing 429, 430
storage usage, reducing 429, 431
uses for 8, 13
archive copy group
defining 387, 389
deleting 408

archive copy group *(continued)*
  description of 364
archiving
  file 362, 375
  file management 362
ASCII restriction for browser script definition 478
ASSIGN DEFMGMTCLASS command 360, 389
assigned capacity 493, 500
association, client with schedule
  defining 435
  deleting 444
association, file with management class 370, 371
association, object with profile
  administrative command schedule 583
  administrator 581, 594
  client option set 581
  deleting 585
  policy domain 582
  script 581
AUDIT LIBRARY command 55
AUDIT LIBVOLUME command 181
AUDIT LICENSE command 459
AUDIT VOLUME command 671, 676
auditing
  library's volume inventory 181
  license, automatic by server 459
  multiple volumes in sequential access storage pool 677
  single volume in sequential access storage pool 677
  volume in disk storage pool 676
  volume, reasons for 671
  volumes by date 677
  volumes by storage pool 678
authentication, client/server 354
authority
  client access 327
  granting to administrators 348
  privilege classes 348
  server options 348
AUTOFSRENAME parameter 332
AUTOLABEL parameter for tape volumes 169
Automated Cartridge System Library Software (ACSLS)
 StorageTek library
  configuring 123
  description 64
  Tivoli Storage Manager server options for 102
automated library device
  auditing 181
  changing volume status 179
  checking in volumes 171
  defining 63
  informing server of new volumes 171
  labeling volumes 169
  overflow location 223
  removing volumes 180
  returning volumes 180
  scratch and private volumes 69
  updating 187
  volume inventory 69
automatically renaming file spaces 332
automating
  administrative commands 32
  client operations 30, 434
  server operations 471
  server startup 460
awk script 694, 717

B

background processes 465
backup
  amount of space used by client 293
  comparison of types 11, 14
  database 655, 660
  default policy 359
  defining criteria for client files 378
  differential, for NAS node 10, 75
  file 362, 373, 374
  file management 362
  file while open 381
  frequency for file 382
  full, for NAS node 75
  group 12
  incremental 362, 373
  logical volume 374
  NAS file server 75
  policy 26
  selective 362, 374
  snapshot, using hardware 9, 12
  storage pool 648
  subfiles, server set-up 25, 418
  types available 10, 14
  when to perform for database 654
backup copy group
  defining 381, 386
  deleting 408
  description of 364
  frequency 373
  mode 373
  serialization 373
BACKUP DB command 660
BACKUP DEVCONFIG command 658
backup period, specifying for incremental 449
backup set
  adding subfiles to 420
  deleting 418
  description of 413
  displaying contents of 417
  example of generating 414
  generating 413
  how the server manages and tracks 415
  media, selecting 413
  moving to other servers 415
  OST extension on 413
  selecting a name for 414
  selecting retention period for 415
  suggested usage 9
  updating 416
  use as archive 13, 15
  viewing information about 416
backup sets, managing 415
BACKUP STGPOOL command 648
BACKUP VOLHISTORY command 656
backup-archive client
  description of 3
  operations summary 10
  performing operations for 411, 441, 446
  policy for 367
  registering node 310
  scheduling operations for 434
  using to back up NAS file server 145, 161
bar-code reader
  auditing volumes in a library 181
  checking in volumes for a library 174
  labeling volumes in a library 170

base file 418
batch file, scheduling on client 437
binding a file to a management class 370
browser, limited to ASCII entry for script definition 478
buffer pool 503
BUFPOOLSIZE option 469, 505

C

cache
  deleting files from 249, 294
  description of 23
  disabling for disk storage pools 248
  effect on performance 248
  effect on statistics 249
  enabling for disk storage pools 222, 248
  monitoring utilization on disk 289
CANCEL PROCESS command 288, 466
CANCEL RESTORE command 347
CANCEL SESSION command 344
capacity, assigned 493, 500
capacity, tape 213
cartridge
  cleaner cartridge 191
  device support 91
categories, 349X library 111
center, administration
  using 36
central scheduling
  client operations 411, 433, 441, 446
  controlling the workload 449
  coordinating 447
  description of 31, 433
  server operations 472
changing date and time on server 464
characteristics, machine 694
check in
  cleaner cartridge 191
  library volume 171
  setting a time interval for volume 201
  VolSafe-enabled volumes 210
CHECKIN LIBVOLUME command 171
checking the log file generated by processed schedules 442
checklist for DRM project plan 714
CHECKOUT LIBVOLUME command 180
class, administrator privilege
  description 348
  granting authority 348
  reducing 354
  revoking all 354
class, device
  3570 197
  3590 197, 198
  3592 197
  4MM 197, 198
  8MM 197, 198
  amount of space used 292
  defining for database backup 653
  deleting 212
  DISK 197
  DLT 197, 198
  ECARTRIDGE 197
  FILE 197
  FORMAT parameter 199
  LTO 207
  NAS 156
  QIC 197, 198
class, device *(continued)*
  REMOVABLEFILE 203
  requesting information about 211
  selecting for import and export 618
  SERVER 197, 198, 603
  StorageTek devices 197, 210
  tape 198
  Ultrium, LTO 197
  VOLSAFE 210
class, policy privilege
  description 348, 353
  granting 353
  revoking 354
class, storage privilege
  description 348
  granting 353
  reducing 354
  revoking 354
classes, device
  adding 39
  modifying 39
  viewing 39
CLEAN DRIVE command 55, 189
cleaner cartridge
  checking in 191
  how often to use 190
  operations with 191
  restrictions on cleaning 190
CLEANFREQUENCY parameter 190
client
  access user ID 327
  administrative 3
  API (application program interface) 4, 313
  application client 4, 400
  backup-archive 3
  how to protect 8
  operations summary 10
  options file 313
  restore without primary volumes available 651
  Tivoli Storage Manager for Space Management (HSM client) 4, 367
  using to back up NAS file server 75, 161
client file
  allowing archive while changing 360
  allowing backup while changing 359, 381
  archive package 426
  associating with management class 370, 371
  damaged 651
  delaying migration of 243
  deleting 304
  deleting from a storage pool 303
  deleting from cache 249
  deleting when deleting a volume 304
  duplication when restoring 651
  eligible for archive 360, 372
  eligible for backup 359, 372
  eligible for expiration 361
  eligible for space management 375
  how IBM Tivoli Storage Manager stores 234
  on a volume, querying 284
  server migration of 238
client migration 375, 376
client node
  adding 309
  agent 322
  amount of space used 291
  archive performance considerations 427

client node *(continued)*
converting 429
creating backup sets for 413
file spaces, QUERY OCCUPANCY command 291
finding tapes used by 286
immediate processing 452
importing 630
locking 320
managing registration 310, 318, 457
options file 314
performing operations for 411, 441, 446
privilege class for scheduling operations for 434
proxy node relationships 321
querying 323
reducing archive packages for 431
registering 313
removing 321
renaming 320
reversing conversion 429
scheduling operations for 434
setting password authentication 355
setting scheduling mode 449
setting up subfile backups 419
target 322
unlocking 320
updating 320
viewing information about 323
client nodes
adding 39
modifying 39
viewing 39
client option
TXNBYTELIMIT 235
VIRTUALMOUNTPOINT 329
client option set
adding client options to 341
assigning clients to 342
copying 342
creating 341
deleting 343
deleting an option from 342
for NAS node 158
requesting information about 342
updating description for 343
client point-in-time restore, enabling 404
client queries to the server, setting the frequency 451
client restartable restore session
canceling 347
interrupting, active 347
requesting information about 347
client session
canceling 344
DSMC loop 343
held volume 343
managing 343
querying 343, 514
viewing information about 343, 514
client system options file 313
client-polling scheduling 447, 451
client, application
adding node for 310
description 4
policy for 400
client/server, description of 3
closed registration
description 311
process 311
closed registration *(continued)*
setting 311
collocation
changing, effect of 264
definition 222, 258, 301
description of 23
determining whether to use collocation 222, 258, 301
effects on operations 260
effects on volume reclamation 276
enabling for sequential storage pool 222, 258, 301
how the server selects volumes when disabled 263
how the server selects volumes when enabled 261
migration thresholds 244
command file, scheduling on client 437
command retry attempts
setting the amount of time between 452
setting the number of 452
command routing 597
command script 478
commands, administrative
ACCEPT DATE 464
ASSIGN DEFMGMTCLASS 360, 389
AUDIT LIBVOLUME 181
AUDIT LICENSE 459
AUDIT VOLUME 676
BACKUP DB 660
BACKUP DEVCONFIG 658
BACKUP NODE 160, 161
BACKUP STGPOOL 648
BACKUP VOLHISTORY 657
BEGIN EVENTLOGGING 527
CANCEL PROCESS 466
CANCEL RESTORE 347
CANCEL SESSION 344
CHECKIN LIBVOLUME 171
CHECKOUT LIBVOLUME 180
CLEAN DRIVE 189
COMMIT 487
CONVERT ARCHIVE 429, 431
COPY CLOPTSET 342
COPY DOMAIN 379
COPY POLICYSET 380
COPY SCHEDULE 442
COPY SCRIPT 482
COPY SERVERGROUP 600
DEFINE ASSOCIATION 30, 435
DEFINE BACKUPSET 415
DEFINE CLIENTACTION 452
DEFINE CLIENTOPT 452
DEFINE CLOPTSET 340
DEFINE COPYGROUP 381, 386, 387
DEFINE DATAMOVER 139, 158
DEFINE DBBACKUPTRIGGER 653, 655
DEFINE DBVOLUME 499
DEFINE DEVCLASS
FILE device classes 204
LTO device classes 207
REMOVEABLEFILE device classes 203
SERVER device classes 208
tape device classes 198
VOLSAFE device classes 210
DEFINE DOMAIN 378
DEFINE DRIVE 138
DEFINE GRPMEMBER 599
DEFINE LIBRARY 63, 64, 137
DEFINE LOGCOPY 669
DEFINE LOGVOLUME 499

commands, administrative *(continued)*
DEFINE MACHINE 693
DEFINE MACHNODEASSOCIATION 694
DEFINE PATH 140, 158
DEFINE POLICYSET 379, 380
DEFINE PROFASSOCIATION 580, 581
DEFINE PROFILE 580
DEFINE RECMEDMACHASSOCIATION 695
DEFINE RECOVERYMEDIA 695
DEFINE SCHEDULE 473
DEFINE SCRIPT 478
DEFINE SERVER 569, 597, 602
DEFINE SERVERGROUP 599
DEFINE SPACETRIGGER 497
DEFINE STGPOOL 224, 225, 234
DEFINE SUBSCRIPTION 590
DEFINE VIRTUALFSMAPPING 162
DEFINE VOLUME 68, 69, 229
DELETE ASSOCIATION 444
DELETE BACKUPSET 418
DELETE COPYGROUP 408
DELETE DBBACKUPTRIGGER 655
DELETE DBVOLUME 502
DELETE DEVCLASS 212
DELETE DOMAIN 409
DELETE DRIVE 193
DELETE EVENT 446
DELETE GRPMEMBER 601
DELETE LIBRARY 187
DELETE LOGVOLUME 502
DELETE MGMTCLASS 408
DELETE POLICYSET 409
DELETE PROFASSOCIATION 585
DELETE PROFILE 586
DELETE SCHEDULE 442
DELETE SCRIPT 484
DELETE SERVER 574
DELETE SERVERGROUP 600
DELETE STGPOOL 303
DELETE SUBSCRIBER 596
DELETE SUBSCRIPTION 586, 592
DELETE VOLHISTORY 656, 657
DELETE VOLUME 305
DISABLE EVENTS 527
DISABLE SESSIONS 346
DISMOUNT VOLUME 186
DSMFMT 226, 645
DSMSERV FORMAT 663
ENABLE EVENTS 527
ENABLE SESSIONS 346
END EVENTLOGGING 527
EXPIRE INVENTORY 32, 391
EXPORT ADMIN 609
EXPORT NODE 619
EXPORT POLICY 619
EXPORT SERVER 619
EXTEND DB 500
EXTEND LOG 653
GENERATE BACKUPSET 413
GRANT AUTHORITY 348
HALT 462
HELP 469
IMPORT 632, 633
IMPORT ADMIN 621
IMPORT NODE 621, 629
IMPORT POLICY 621
IMPORT SERVER 621, 629

commands, administrative *(continued)*
LABEL LIBVOLUME 106, 134
LOCK ADMIN 352
LOCK NODE 320
LOCK PROFILE 584, 585
MOVE DATA 294
MOVE NODEDATA 298
NOTIFY SUBSCRIBERS 584, 585
PING SERVER 601
PREPARE 696
QUERY ACTLOG 524
QUERY BACKUPSET 416
QUERY BACKUPSETCONTENTS 417
QUERY CONTENT 284
QUERY COPYGROUP 406, 627
QUERY DB 501
QUERY DBBACKUPTRIGGER 655
QUERY DBVOLUME 501, 647
QUERY DEVCLASS 211
QUERY DOMAIN 407
QUERY DRIVE 188
QUERY DRMSTATUS 688
QUERY ENABLED 540
QUERY EVENT 32, 436
QUERY FILESPACE 338
QUERY LIBRARY 186
QUERY LICENSE 459
QUERY LOG 506
QUERY LOGVOLUME 501, 647
QUERY MGMTCLASS 406
QUERY MOUNT 186
QUERY NODE 323
QUERY NODEDATA 292
QUERY OCCUPANCY
backed-up, archived, and space-managed files 293
client file spaces 291
client nodes 291
device classes 292
storage pools 292
QUERY OPTION 516
QUERY POLICYSET 407
QUERY PROCESS 296
QUERY REQUEST 184
QUERY RESTORE 347
QUERY RPFCONTENT 697
QUERY RPFILE 697
QUERY SCHEDULE 436
QUERY SCRIPT 482
QUERY SERVERGROUP 600
QUERY STGPOOL 279, 287, 625
QUERY SUBSCRIPTION 591
QUERY SYSTEM 517
QUERY VOLHISTORY 656
QUERY VOLUME 281, 282, 297
RECONCILE VOLUMES 606
REDUCE DB 502
REDUCE LOG 502
REGISTER ADMIN 351
REGISTER LICENSE 459
REMOVE ADMIN 352
REMOVE NODE 321
RENAME ADMIN 351
RENAME FILESPACE 631
RENAME NODE 320
RENAME SCRIPT 483
RENAME SERVERGROUP 600
RENAME STGPOOL 300

commands, administrative *(continued)*
RESET BUFPOOL 503
RESET DBMAXUTILIZATION 493, 495
RESET LOGCONSUMPTION 653
RESET LOGMAXUTILIZATION 493, 495
RESTORE NODE 160, 161
RESTORE STGPOOL 649, 681
RESTORE VOLUME 683
ROLLBACK 488
RUN 484
SELECT 518
SET ACCOUNTING 541
SET AUTHENTICATION 355
SET CLIENTACTDURATION 452
SET CONFIGMANAGER 577, 579
SET CONFIGREFRESH 591
SET CONTEXTMESSAGING 527
SET CROSSDEFINE 570, 573
SET DRMCHECKLABEL 691
SET DRMCOPYSTGPOOL 689
SET DRMCOURIERNAME 691
SET DRMDBBACKUPRXPIREDAYS 691
SET DRMFILEPROCESS 692
SET DRMINSTPREFIX 689
SET DRMNOTMOUNTABLE 691
SET DRMPLANPOSTFIX 689
SET DRMPLANPREFIX 690
SET DRMPRIMSTGPOOL 688
SET DRMRPFEXPIREDAYS 697
SET DRMVAULTNAME 692
SET EVENTRETENTION 446, 477
SET INVALIDPWLIMIT 355
SET LICENSEAUDITPERIOD 459
SET LOGMODE 656
SET MAXCMDRETRIES 452
SET MAXSCHEDSESSIONS 449
SET MINPWLENGTH 355
SET PASSEXP 354
SET QUERYSCHEDPERIOD 451
SET RANDOMIZE 450
SET REGISTRATION 311
SET RETRYPERIOD 452
SET SCHEDMODES 447
SET SERVERHLADDRESS 570, 573
SET SERVERLLADDRESS 570, 573
SET SERVERNAME 516, 569, 573
SET SERVERPASSWORD 569, 570, 573
SET SUBFILE 418
SET SUMMARYRETENTION 522
SETOPT 469
UNDO ARCHCONVERSION 429, 431
UNLOCK PROFILE 584, 585
UPDATE ADMIN 351
UPDATE ARCHIVE 431
UPDATE BACKUPSET 416
UPDATE CLIENTOPT 342
UPDATE CLOPTSET 343
UPDATE COPYGROUP 381, 387
UPDATE DBBACKUPTRIGGER 656
UPDATE DOMAIN 379
UPDATE DRIVE 188
UPDATE LIBRARY 186
UPDATE LIBVOLUME 69, 179
UPDATE MGMTCLASS 380
UPDATE NODE 320
UPDATE POLICYSET 379
UPDATE RECOVERYMEDIA 695
commands, administrative *(continued)*
UPDATE SCHEDULE 473
UPDATE SCRIPT 481
UPDATE SERVER 574
UPDATE SERVERGROUP 600
UPDATE VOLUME 229
VALIDATE LANFREE 136
COMMIT command 487
COMMTIMEOUT server option 344, 345
communication set up
among servers 568
command routing, for 571
cross definition 569, 570, 573
enterprise configuration, for 568
enterprise event logging, for 538, 568
security 570
server-to-server virtual volumes 602
compressing the server database 508
compression
choosing client or drive 213
option for API 313
options for clients 312
setting 312
tape volume capacity, effect on 214
configuration file, device
backing up 658
example 660
information 658
recreating 659
using when compressing the database 509
configuration information, enterprise management
administrative command schedule 577, 579, 583
administrator 581, 594
changing 584
client option set 578, 581
client schedule 578, 579, 582
copy group 578, 582
deleting 585, 586
distributing 575, 581, 584
management class 582
policy domain 578, 579, 582
refreshing 584, 591, 593
script 577, 581
server 582
server group 582
configuration manager
communication setup 568
default profile 577, 582
scenario 577
setting up 577, 579, 580
configuring
devices, automated library example 102, 112
devices, manual library example 133
drives with more than one device type in a single library 198
NDMP operations for NAS file servers 154
planning your storage environment 70
shared library 107, 117
console mode 633
contents of a volume 284
context messaging for ANR9999D 527
continuation characters, using 479
conversion of client nodes 429
CONVERT ARCHIVE command 429, 431
COPY CLOPTSET command 342
COPY DOMAIN command 379

copy group
archive, description of 364
backup, description of 364
defining archive 387
defining backup 381
deleting 408
COPY MGMTCLASS command 381
COPY POLICYSET command 379
COPY SCHEDULE command 442, 476
COPY SCRIPT command 482
COPY SERVERGROUP command 600
copy storage pool
compared with primary 302
defining a 301
restore from multiple 651
role in storage pool migration 248
simultaneous write 249
storage hierarchy, effect on 237
creating
backup sets 25
client schedules 30
new policy 26
server scripts 478
creating backup sets
benefits of 413
example for 414
cross definition 569, 570, 573
customer support, contacting xvi
cyclic redundancy check
during a client session 411
for storage pool volumes 673
for virtual volumes 602
performance considerations for nodes 412
perfornance considerations for storage pools 676

D

damaged files 640, 678, 679
data
considering user needs for recovering 81
exporting 609
importing 609
data compression 312
data format for storage pool 145, 165
definition 223
operation restrictions 224
data movement
LAN-free 49
data movement, querying 296
data mover
defining 139, 158
description 68
managing 164
NAS file server 68
data protection with WORM media 174
data retention protection 55, 393
data storage
client files, process for storing 5
concepts overview 15
considering user needs for recovering 81
deleting files from 304
evaluating 70
example 219
managing 20
monitoring 671
planning 70
protection, methods 174, 640
data storage *(continued)*
server options affecting 102
tailoring definitions 627
using another IBM Tivoli Storage Manager server 601
using disk devices 83
using the storage hierarchy 237
data validation
during a client session 411
for storage pool volumes 673
for virtual volumes 602
performance considerations for nodes 412
perfornance considerations for storage pools 676
database backup and recovery
database backup trigger 655, 666
defining device classes for 653
example recovery procedures 680
full backup 654, 660
general strategy 601, 639, 640
incremental backup 654
methods 601, 639
point-in-time 662
providing 601, 639
roll-forward 642, 666
to most current state 666
unloading 509
when to back up 654
database volumes
adding 39
modifying 39
viewing 39
database, IBM Tivoli Storage Manager
adding space to 499
automating increase of 497
available space 492, 495
backup 175, 660
backup trigger 655
buffer pool 503, 505
committing data to 505
compressing 508
defining a volume 499
defining mirrored volumes 645
deleting a volume 502
deleting space 501
description of 32, 489
determining how much space is allocated 492, 495
DEVCONFIG option 509
ensuring integrity of 35
estimating the amount of space needed 494
estimating the benefit of reorganization 506
formatting 509
fragmented 508
loading 509
logical volume 492, 495
managing 489
mirroring 645
monitoring space 493, 495
monitoring the buffer 505
optimizing performance 503
querying the buffer pool 504
querying using SQL 518
querying volumes 496
recovering 661
reducing capacity 502
reloading 509
reorganize 508
resetting buffer pool statistics 504
restoring 653

database, IBM Tivoli Storage Manager *(continued)*
shadowing 646
space trigger 496, 497
storage pool size effect 489
transactions 489, 490
unloading 508
viewing information about 504
date and time, changing on the server 464
day of the week parameter 473
DBPAGESHADOW server option 646
DBPAGESHADOWFILE server option 646
deactivating policy 360
Decision Support Loader 541
default management class
assigning for a policy set 389
binding files to 371
description of 365
purpose 368
recommendation for using 370
default policy 359
default profile 577, 582, 589
DEFINE ASSOCIATION command 30, 435
DEFINE BACKUPSET command 415
DEFINE CLIENTACTION command 452
DEFINE CLIENTOPT command 341
DEFINE CLOPTSET command 340
DEFINE COPYGROUP command 381, 386, 387, 389
DEFINE DBBACKUPTRIGGER command 653, 655
DEFINE DBVOLUME command 499
DEFINE DEVCLASS command 198
DEFINE DOMAIN command 378
DEFINE DRIVE command 138
DEFINE GRPMEMBER command 599
DEFINE LIBRARY command 137
DEFINE LOGCOPY command 669
DEFINE LOGVOLUME command 499
DEFINE MACHINE command 693
DEFINE MACHNODEASSOCIATION command 694
DEFINE MGMTCLASS command 380
DEFINE POLICYSET command 379
DEFINE PROFASSOCIATION command 581
DEFINE PROXYNODE command 322
DEFINE RECMEDMACHASSOCIATION command 695
DEFINE RECOVERYMEDIA command 695
DEFINE SCHEDULE command 30, 32, 473
DEFINE SCRIPT command 478
DEFINE SERVER command 569, 597, 602
DEFINE SPACETRIGGER command 497
DEFINE STGPOOL command 224, 225, 234
DEFINE SUBSCRIPTION command 590
DEFINE VIRTUALFSMAPPING command 162
DEFINE VOLUME command 22, 229
delaying migration for files 243
delaying reuse of volumes 276
DELETE ASSOCIATION command 444
DELETE BACKUPSET command 418
DELETE CLIENTOPT command 342
DELETE COPYGROUP command 408
DELETE DBBACKUPTRIGGER 655
DELETE DBVOLUME command 502
DELETE DEVCLASS command 212
DELETE DOMAIN command 409
DELETE DRIVE command 193
DELETE EVENT command 446, 477
DELETE FILESPACE command 339
DELETE GRPMEMBER command 601
DELETE LIBRARY command 187
DELETE LOGVOLUME command 502
DELETE MGMTCLASS command 408
DELETE POLICYSET command 409
DELETE PROFASSOCIATION command 585
DELETE PROFILE command 586
DELETE SCHEDULE command 442, 476
DELETE SCRIPT command 484
DELETE SERVER command 575
DELETE SERVERGROUP command 600
DELETE STGPOOL command 303
DELETE SUBSCRIBER command 596
DELETE SUBSCRIPTION command 592
DELETE VOLHISTORY command 656, 657
DELETE VOLUME command 305
deleting
cached files on disk 294
empty volume 305, 657
file spaces 339
files 304, 391
media-managed storage pools 133
scratch volume 228, 657
storage volume 305
subfile 420
volume history information 657
volume with residual data 305
deletion hold 394
descriptions, for archive packages 426, 430
DESTINATION parameter (storage pool) 359, 360, 381
destroyed volume access mode 232, 641
determining
cause of ANR9999D messages 527
the time interval for volume check in 201
DEVCONFIG option 508, 658
device
attaching to server 146
element number 160
multiple types in a library 100
name 94
supported by IBM Tivoli Storage Manager 91
zfcp device driver 97
device class
3570 197
3590 197, 198
3592 197
4MM 197, 198
8MM 197, 198
amount of space used 292
defining for database backup 653
deleting 212
DISK 197
DLT 197, 198
ECARTRIDGE 197
FILE 197
FORMAT parameter 199
LTO 207
NAS 156
QIC 197, 198
REMOVABLEFILE 203
requesting information about 211
selecting for import and export 618
SERVER 197, 198, 603
StorageTek devices 197, 210
tape 198
Ultrium, LTO 197
VOLSAFE 210
device classes
adding 39

device classes *(continued)*
modifying 39
viewing 39
device configuration file
backing up 658
example 660
information 658
recreating 659
using when compressing the database 509
device driver
configuring 96
for IBM 3490, 3570, and 3590 tape drives 95
for manual tape devices 92
IBM Tivoli Storage Manager, installing 92, 100
installing 91, 93
requirements 91, 93
device sharing 70
device type
3590 198
4MM 198
8MM 198
DISK 197
DLT 198
ECARTRIDGE 197
LTO 198, 207
multiple in a single library 100, 198
NAS 156
QIC 198
SERVER 198, 602, 603
VOLSAFE 210
device, storage
automated library device 102, 112
disk 83
manual library device 133
optical device 129, 133
overview for removable media 99
removable media device 129, 203
required IBM Tivoli Storage Manager definitions 80
supported devices 91
diagnosing ANR9999D messages 527
differential backup
compared to incremental 14
of image, description 10, 75
direct-to-tape, policy for 399
directories
deleting from archive packages 431
directory-level backup 154
preventing archive of 431
storage usage for archive packages 426
DISABLE EVENTS command 527
DISABLE SESSIONS command 346
disaster recovery
auditing storage pool volumes 679
example recovery procedures 680
general strategy 601, 639
methods 35, 601, 639
providing 601, 639
when to backup 640, 654
disaster recovery manager
awk script 717
client recovery information 688
creating a disaster recovery plan 696
customizing 688
displaying a disaster recovery plan 697
enabling 687
expiring a disaster recovery plan 697
features 688
disaster recovery manager *(continued)*
moving volumes back onsite 702
not available in Administration Center 55
project plan, checklist 714
querying a disaster recovery plan 697
recovery media 695
saving machine characteristics 693
stanzas, recovery instructions 692
storing a disaster recovery plan 696
disaster, protecting against 35
disk device class, defined 197
disk storage
and file-system requirements 83
random access (DISK) 65
sequential access (FILE) 65
disk storage pool
cache, use of 248
deleting cached files from 294
estimating space 277
estimating space for archived files 278
estimating space for backed up files 277
migration threshold 239
setting up 83
DISMOUNT VOLUME command 186
DLT WORM media 174
domain, policy
assigning client node 391
changing 360
creating 379
deleting 409
description of 364
distributing via profile 405, 579
for NAS file server node 157
querying 407
updating 376, 378
drive
cleaning 189
configuring multiple for single library 198
defining 138
defining path for 140
deleting 193
detecting changes on a SAN 142
element address 139, 140
multiple device types in a library 105
querying 188
serial number 138
server retries for acquiring 102
simultaneous write, requirements for 256
updating 188
updating to use for NDMP operations 165
DRIVEACQUIRERETRY server option 102
driver, device
configuring 96
for IBM 3490, 3570, and 3590 tape drives 95
for manual tape devices 92
IBM Tivoli Storage Manager, installing 92, 100
installing 91, 93
requirements 91, 93
drives
adding 39
modifying 39
viewing 39
dsm.opt file 313, 340, 433
dsmaccnt.log 541
DSMADMC command 615, 627, 633
DSMC loop session 343
DSMFMT utility 226, 645

DSMLABEL utility 169, 170
dsmsched.log file 442
DSMSERV DISPLAY DBBACKUPVOLUME command 659
DSMSERV FORMAT utility 461
DSMSERV LOADDB utility 509, 510
DSMSERV LOADFORMAT utility 509
DSMSERV RESTORE DB command 659
DSMSERV UNLOADDB utility 509
DSMSERV_ACCOUNTING_DIR 541
duplication of restored data 651
dynamic serialization, description of 382, 387

E

ECARTRIDGE device class 197
education xv
element address 138, 139
ENABLE EVENTS command 527
ENABLE SESSIONS command 346
encryption key 642
END EVENTLOGGING command 527
Enterprise Administration
  description 563
enterprise configuration
  communication setup 568
  description 564, 575
  procedure for setup 576
  profile for 578
  scenario 566, 576
  subscription to 579
enterprise event logging 538, 568
enterprise management
  setting up 49
environment variable, accounting 541
environment variables 461
error analysis 517
error checking for drive cleaning 192
error reporting for ANR9999D messages 527
error reports for volumes 283
establishing server-to-server communications
  enterprise configuration 568
  enterprise event logging 568
  virtual volumes 574
ESTIMATE DBREORGSTATS command 506
estimated benefit of reorganization 506
estimated capacity for storage pools 279
estimated capacity for tape volumes 283
estimated database space savings 506
event logging 55, 526
event record (for a schedule)
  deleting 446, 477
  description of 436, 445
  managing 476
  querying 476
  removing from the database 446, 477
  setting retention period 446, 477
event server 538
events, querying 32
EXPINTERVAL option 391
expiration date, setting 474
expiration processing 32
  description 391, 652
  files eligible 361, 391
  of subfiles 361, 384, 391, 420
  starting 391
  using disaster recovery manager 392
EXPIRE INVENTORY command 32
EXPIRE INVENTORY command *(continued)*
  deleting expired files 391
  duration of process 392
export
  labeling tapes 613
  monitoring 631
  planning for sequential media 618
  PREVIEW parameter 617
  querying about a process 632
  querying the activity log 633
  using scratch media 618
  viewing information about a process 632
EXPORT ADMIN command 619
EXPORT commands 632, 633
EXPORT NODE command 619
EXPORT POLICY command 620
EXPORT SERVER command 617, 621
exporting
  administrator data 619
  client node data 619
  data to OS/400 PASE 617
  data to tape 619
  description of 609
  policy data 620
  server data 621
  subfiles 419
EXPQUIET server option 392
EXTEND DB command 500
EXTEND LOG command 500
EXTERNAL library type 745
external media management
  IBM Tivoli Storage Manager setup 131
  initialization requests 745
  interface description 741
  overview 131
  processing during server initialization 742
  using with IBM Tivoli Storage Manager 132
  volume dismount requests 750
  volume mount requests 747
  volume release requests 747

F

file data, importing 609
FILE device type
  benefits 65
  defining device class 197
  deleting scratch volumes 657
  setting up storage pool 88
file exit 526
file name for a device 94, 159
file retrieval date 249
file server, network-attached storage (NAS)
  backup methods 75
  registering a NAS node for 158
  using NDMP operations 74, 143
file size, determining maximum for storage pool 222
file space
  deleting, effect on reclamation 267
  deleting, overview 339
  description of 329
  merging on import 612, 622
  names that do not display correctly 339
  QUERY OCCUPANCY command 291
  querying 329
  renaming 631
  Unicode enabled 338

file space *(continued)*
viewing information about 329
file space identifier (FSID) 338
file-level restore
managing 162
planning 152
file-system requirements for disk storage 83
file, client
allowing archive while changing 360
allowing backup while changing 359, 381
archive package 426
associating with management class 370, 371
damaged 651
delaying migration of 243
deleting 304
deleting from a storage pool 303
deleting from cache 249
deleting when deleting a volume 304
duplication when restoring 651
eligible for archive 360, 372
eligible for backup 359, 372
eligible for expiration 361
eligible for space management 375
how IBM Tivoli Storage Manager stores 234
on a volume, querying 284
server migration of 238
files, damaged 640, 651, 678, 679
files, unreadable 640, 678, 679
firewall, client nodes
client-initiated sessions 318
server-initiated sessions 319
fixes, obtaining xvi
format for storage pool 145, 165
definition 223
operation restrictions 224
formatting
database volume 500
event classes 530
recovery log volume 500
storage pool volume 87, 228
frequency of backup 382
FSID 338
full image backup, NAS node 75
full library 180

G

GENERATE BACKUPSET command 413
GRANT AUTHORITY command 348
group backup, on the client 12
group, server
copying 600
defining 599
deleting 600
member, deleting 601
moving a member 601
querying 600
renaming 600
updating description 600

H

HALT command 462
halting the server 462
held volume in a client session 343
HELP command 469
hierarchical storage management
See Tivoli Storage Manager for Space Management
hierarchy, storage 22
defining in reverse order 224, 234
establishing 232
example 219
how the server stores files in 234
next storage pool
definition 233
deleting 304
migration to 238, 287
staging data on disk for tape storage 237
HL ADDRESS 319
how to cause the server to accept date and time 464
HSM
See Tivoli Storage Manager for Space Management

I

IBM error analysis 517
IBM Tivoli Storage Manager (Tivoli Storage Manager)
introduction 3
server network 33, 563
IDLETIMEOUT server option 344, 345
image backup
policy for 400, 401
suggested use 8, 12
import
monitoring 631
PREVIEW parameter 617, 624
querying about a process 632
querying the activity log 633
recovering from an error 631
viewing information about a process 632
IMPORT ADMIN command 621
IMPORT commands 632, 633
IMPORT NODE command 621, 629
IMPORT POLICY command 621
IMPORT SERVER command 621, 629
importing
data 621
data storage definitions 626, 627
date of creation 623, 629
description of 609
directing messages to an output file 615, 627
duplicate file spaces 628
file data 628
policy definitions 626
server control data 627
subfiles 419
subsets of information 630
include-exclude file
creating 29
description of 26, 368
for policy environment 364, 368
incomplete copy storage pool, using to restore 651
incremental backup, client
file eligibility for 373
frequency, specifying 449
full 373
partial 373
progressive 14
information centers, searching to find software problem resolution xv
inheritance model for simultaneous write 252
initial start date for schedule 473
initial start time for schedule 473

installing IBM Tivoli Storage Manager 310
instant archive
creating on the server 412
description of 9, 13
interface, application program
client, registering 313
compression option 313
deletion option 313
description of 3
registering to server 313
simultaneous write, version support for 250
interfaces to IBM Tivoli Storage Manager 19
Internet, searching to find software problem resolution xvi
introduction to IBM Tivoli Storage Manager 3

K

knowledge bases, searching to find software problem resolution xv

L

label
automatic labeling in SCSI libraries 169
checking media 174
overwriting existing labels 168, 169
sequential storage pools 167, 229
volume examples 169
volumes using a library device 169
LABEL LIBVOLUME command
identifying drives 169
insert category 170
labeling sequential storage pool volumes 168
manually mounted devices 133
overwriting existing volume labels 168
removable media volumes 168
restrictions for VolSafe-enabled drives 210
SCSI libraries used by one server 102
using a library device 169
using a manual library 134
using an automated library 106, 116, 128
volume labeling examples 169
LAN-free data movement
configuration 135
description 15, 73
enabling 49
suggested usage 10
libraries
adding 39
modifying 39
viewing 39
library
ACSLS (Automated Cartridge System Library Software) 64, 123
adding volumes 171
attaching for NAS file server backup 146, 155
auditing volume inventory 181
automated 179
categories for volumes in IBM 3494 111
configuration example 102, 112, 133
configure for more than one device type 100, 198
defining 137, 188
defining path for 140, 158
deleting 187
detecting changes to, on a SAN 138, 142
external 64
library *(continued)*
full 180
IBM 3494 64, 111
managing 186
manual 63, 133, 184
mixing device types 100
mode, random or sequential 93
overflow location 223
querying 186
SCSI 63
serial number 138
sharing among servers 107, 117
type 71
updating 186
volume inventory 69
library client, shared library 72, 110, 118, 128
library manager, shared library 72, 108, 118, 127
license
compliance 459
features 457
monitoring 459
registering 458
using 457
limitation for script definition on administrative Web interface 478
limitations, Administration Center 55
LL ADDRESS 319
location, volume
changing 230
overflow for storage pool 223
querying volume 284
LOCK ADMIN command 352
LOCK NODE command 320
LOCK PROFILE command 584, 585
log mode
normal 653, 655
roll-forward 653, 655
setting 653
logical devices 88
logical volume on client
backup 362
management class for 370
policy for 372, 400
process for backup 374
restore 362
LOGPOOLSIZE option 505
loop session, DSMC 343
LTO Ultrium device type
device class, defining and updating 207
WORM 174, 202
LUN
using in paths 140

M

machine characteristics 694
machine recovery information 694
macro
commit individual commands 487
continuation characters 486
controlling command processing 487
running 487
scheduling on client 437
substitution variables 486
testing 488
using 484
writing commands 485

macro *(continued)*
writing comments 485
MACRO administrative command, using 317
magnetic disk devices 65, 83
managed server
changing the configuration manager 590, 595
communication setup 568
deleting a subscription 592
description 564
managed objects 564, 589
refreshing configuration information 593
renaming 596
returning managed objects to local control 594
setting up 578
subscribing to a profile 579, 589, 590
management class
assigning a default 389
associating a file with 370
binding a file to 370
configuration 367
controlling user access 367
copying 376, 380
default 29, 368
defining 380
deleting 408
description of 364, 367
querying 406
rebinding a file 371
updating 371, 376, 381
management classes
adding 39
modifying 39
viewing 39
manual library device 133
maximum extension 499
MAXSCRATCH parameter 221, 230, 301
media
See also device, tape, and volume
loss, recovery from 683
tape rotation 77, 177
media label
checking 174
for tape 168
recording 168
merging file spaces 612, 622
messages
determining cause of ANR9999D message 527
directing import messages to an output file 615, 627
for automated libraries 184
for drive cleaning 192
getting help on 469
mount, using the administrative client 184
severe 527
MIGRATE STGPOOL command 246
migrating a file 363, 375
migration, client
automatic, for HSM client
demand 363
files, eligible 375
threshold 363
using management class 376
premigration for HSM client 363
reconciliation 363
selective, for HSM client 363
stub file on HSM client 363
migration, server
canceling the server process 288
migration, server *(continued)*
controlling by file age 243
controlling duration 246
controlling start of, server 242
copy storage pool, role of 248
defining threshold for disk storage pool 242
defining threshold for tape storage pool 244
delaying by file age 243
description, server process 240
minimizing access time to migrated files 244
monitoring thresholds for storage pools 287
multiple concurrent processes
random access storage pool 222, 240
sequential access storage pool 222, 247
problems, diagnosing and fixing 239
providing additional space for server process 289
starting manually 246
starting server process 238, 242
threshold for a storage pool 240
mirrored volume
description of 647
querying 647
viewing information about 647
mirroring
advantages 645
database 645
defining volumes 647
description of 35
recovery log 642, 645
recovery procedure 669
mixed device types in a library 100
mobile client support 418
mode
client backup 382
library (random or sequential) 93
scheduling 447
modified mode, description of 382
monitoring the IBM Tivoli Storage Manager server 513
mount
count of number of times per volume 283
library 200
limit 200
mode 184
operations 183
query 186
retention period 201
wait period 201
mount point
preemption 467
queue, server option 102
relationship to mount limit in a device class 200, 206
requirements for simultaneous write 255
settings for a client session 311
MOVE BATCHSIZE option 469
MOVE DATA command 294
MOVE DRMEDIA command 703
MOVE NODEDATA 298
MOVE SIZETHRESH option 469
moving a backup set
benefits of 415
to another server 415
moving data
from offsite volume in a copy storage pool 295
monitoring the movement of 297
procedure 295
requesting processing information 296
to another storage pool 295

moving data *(continued)*
to other volumes in same storage pool 294
multiple
copy storage pools, restoring from 651
device types in a single library 198
managing IBM Tivoli Storage Manager servers 33, 563
servers, running 461

N

name of device 94
NAS file server
protecting 49
NAS file server, NDMP operations
backing up a NAS file server 161
configuration checklist 154
data format 145
data mover, description 68, 139
defining a data mover 139, 158
defining a device class 156
defining a path for data mover and a library 158
defining a storage pool 156
defining a tape drive 159
differential image backup, description 75
full image backup, description 75
interfaces used with 145
managing NAS nodes 164
path, description 68, 140
planning 146
policy configuration 157, 401
registering a NAS node 158, 312
requirements for set up 143
restoring a NAS file server 161
scheduling a backup 160
storage pools for NDMP operations 156
NAS node
defining 158
deleting 164
registering 158
renaming 164
NATIVE data format 145
NDMP operations for NAS file servers
backing up a NAS file server 161
configuration checklist 154
data format 145
data mover, description 68, 139
defining a data mover 139, 158
defining a device class 156
defining a path for data mover and a library 158
defining a storage pool 156
defining a tape drive 159
differential image backup, description 75
full image backup, description 75
interfaces used with 145
managing NAS nodes 164
path, description 68, 140
planning 146
policy configuration 157, 401
registering a NAS node 158, 312
requirements for set up 143
restoring a NAS file server 161
scheduling a backup 160
storage pools for NDMP operations 156
NETAPPDUMP data format 145, 156
Network Appliance file server
backup methods 75
data format for backup 145
Network Appliance file server *(continued)*
international characters 153
requirements 143
storage pool for backup 224
tape device for backup 144
using NDMP operations 74, 143
network of IBM Tivoli Storage Manager servers 33, 563
network-attached storage (NAS) file server
backup methods 75
registering a NAS node for 158
using NDMP operations 74, 143
next storage pool
definition 233
deleting 304
migration to 238, 287
node privilege class
description of 326
granting 327
node, client
adding 309
agent 322
amount of space used 291
archive performance considerations 427
converting 429
creating backup sets for 413
file spaces, QUERY OCCUPANCY command 291
finding tapes used by 286
immediate processing 452
importing 630
locking 320
managing registration 310, 318, 457
options file 314
performing operations for 411, 441, 446
privilege class for scheduling operations for 434
proxy node relationships 321
querying 323
reducing archive packages for 431
registering 313
removing 321
renaming 320
reversing conversion 429
scheduling operations for 434
setting password authentication 355
setting scheduling mode 449
setting up subfile backups 419
target 322
unlocking 320
updating 320
viewing information about 323
nodes, client
adding 39
modifying 39
viewing 39
NOPREEMPT server option 467
NORETRIEVEDATE server option 249
NOTIFY SUBSCRIBERS command 584, 585
number of times mounted, definition 283

O

occupancy, querying 291
ODBC driver 518
offsite recovery media (for DRM)
volumes
moving back onsite 702
sending offsite 701
states 700

offsite volume access mode 232
offsite volumes
  limiting the number to be reclaimed 223, 275
  moving data in a copy storage pool 295
one-drive library, volume reclamation 223, 271
open registration
  description 311
  process 311
  setting 311
operational reporting
  command line, options 559
  command line, using 558
  description 543
  management console, installing as part of 546
  monitors, generating 551
  reports, generating 547
  stand-alone package, installing as 547
operations available to client 10
operator privilege class
  reducing 354
  revoking 354
option set, client
  adding client options to 341
  assigning clients to 342
  copying 342
  creating 341
  deleting 343
  deleting an option from 342
  for NAS node 158
  requesting information about 342
  updating description for 343
option, server
  3494SHARED 102
  ACSLS options 102
  ASSISTVCRRECOVERY 102
  AUDITSTORAGE 459, 469
  BUFPOOLSIZE 504
  changing with SETOPT command 469
  COMMTIMEOUT 344, 345
  DEVCONFIG 658
  DRIVEACQUIRERETRY 102
  EXPINTERVAL 392
  EXPQUIET 392
  IDLETIMEOUT 344, 345, 515
  LOGPOOLSIZE 505
  MOVEBATCHSIZE 469
  MOVESIZETHRESH 469
  NOPREEMPT 102, 467
  NORETRIEVEDATE 249
  overview 20
  QUERYAUTH 348
  REQSYSAUTHOUTFILE 348
  RESOURCETIMEOUT 102
  RESTOREINTERVAL 347, 361, 392
  SEARCHMPQUEUE 102
  SELFTUNEBUFPOOLSIZE 469
  THROUGHPUTDATATHRESHOLD 345
  THROUGHPUTTIMETHRESHOLD 345
  TXNGROUPMAX 234
  VOLUMEHISTORY 657
options file, client 314
options, querying
  BUFPOOLSIZE 505
  LOGPOOLSIZE 505
  VIRTUALMOUNTPOINT client option 330
ordering publications xv
OS/400 PASE, exporting data to 617
overflow location 223
owner authority, client 326, 327

P

page shadowing, database server options 646
page, description of 503
PARALLEL command 479
password
  resetting an administrative 351
  setting authentication for a client 355
  setting expiration 354
  setting invalid limit 355
  setting minimum length 355
path
  defining 140
  description 68, 152
  to library 158
paths
  adding 39
  modifying 39
  viewing 39
pending, volume state 284
performance
  archive operations 427
  cache, considerations for using 89, 248
  clients, optimizing restore 420
  concurrent client/server operation considerations 449
  data protection, increasing with simultaneous write 249
  data validation for nodes 412
  data validation for storage pools 676
  database or recovery log, optimizing 503
  database read, increase with mirroring 645
  estimating the benefit of reorganization 506
  file system effects on 87
    random-access disk (DISK) 83, 228
  fragmentation, private FILE volumes for reducing disk 65
  migration, multiple concurrent processes 222, 247
  mobile client 418
  reclamation, multiple concurrent processes
    copy storage pools 274
    primary sequential access storage pools 222, 270
  server options, automatic tuning of 469
  storage pool backup, reducing time required for 249
  storage pool volume 244, 668
  volume frequently used, improve with longer mount retention 201
period, specifying for an incremental backup 449
point-in-time restore
  enable for clients 9, 404
  for database 662
  for storage pools 665
policy
  default 5, 359
  deleting 408
  description of 364
  distributing with enterprise management 405
  effect of changing 390
  for application clients 400
  for clients using SAN devices 402
  for direct-to-tape backup 399
  for logical volume backups 400
  for NAS file server node 157
  for point-in-time restore 404
  for server as client 404
  for space management 360, 375, 380
  importing 626

policy *(continued)*
managing 357
operations controlled by 362
planning 358
querying 405
policy domain
assigning client node 391
changing 360
creating 379
deleting 409
description of 364
distributing via profile 405, 579
for NAS file server node 157
querying 407
updating 376, 378
policy domains
adding 39
modifying 39
viewing 39
policy privilege class
description 348, 353
granting 353
revoking 354
policy set
activating 390
changing, via the active policy set 360
copying 360, 376, 379
defining 379
deleting 409
description of 364
querying 407
updating 379
validating 389, 391
pool, storage
amount of space used 292
auditing a volume 671
backup
full 648
incremental 648
backup and recovery 648
comparing primary and copy types 302
copy 219
creating a hierarchy 232
data format 145, 223, 224
defining 220
defining a copy storage pool 301
defining for disk, example 224, 234
defining for NDMP operations 156
defining for tape, example 224, 234
deleting 303
description of 218
destination in copy group 381, 387
determining access mode 221, 301
determining maximum file size 222
determining whether to use collocation 222, 258, 301
disk 23
duplicate, using to restore 651
enabling cache for disk 222, 248
estimating space for archived files on disk 278
estimating space for backed up files on disk 277
estimating space for disk 277
estimating space for sequential 278
estimating space in multiple 232
incomplete, using to restore 651
increasing sizes 23
LTO Ultrium, special considerations for 207
managing 217
pool, storage *(continued)*
monitoring 279
moving files 294
moving files between 295
multiple, using to restore 651
next storage pool
definition 233
deleting 304
migration to 238, 287
overview 66
policy use 381, 387
primary 218
querying 279
random access 218
recovery log, effect on 489
renaming 300
restoring 649, 681
sequential access 218
simultaneous write 249
updating 220
updating for disk, example 225, 234
using cache on disk 222, 248
validation of data 673
viewing information about 279
WORM media composition 175
portable media
description of 6, 8, 412
restoring from 414
preemption
mount point 467
volume access 467
prefix, for recovery instructions 689
prefix, for recovery plan file 690
prefix, server 597
premigration 363
PREPARE command 696
PREVIEW parameter 617, 624
primary volumes unavailable for restore 651
private category, 349X library 111
private volumes 68
privilege class, administrator
description 348
granting authority 348
reducing 354
revoking all 354
privilege class, policy
description 348, 353
granting 353
revoking 354
privilege class, storage
description 348
granting 353
reducing 354
revoking 354
problem determination
describing problem for IBM Software Support xvii
determining business impact for IBM Software Support xvii
migration 239
reclamation 269
submitting problem to IBM Software Support xviii
process
background 465
canceling 466
drive clean error checking 192
expiration 652
number for migration 222, 240

process *(continued)*
number for storage pool backup 649
number for storage pool restore 667
reclamation 267, 275
profile
associating configuration information with 580
changing 580, 584, 585
default 582, 589
defining 580, 581
deleting 585, 586
description 580
getting information about 587
locking 584
problems with synchronization 595
unlocking 584
programming interface notice 766
progressive incremental backup 14
protecting your data 35, 640
simultaneous write 249
WORM (write once, read many media) 174
protection options
client 8
server 14, 35
proxy node relationships 323
publications, ordering xv

Q

query
authority 348
database volumes 496
for general information 281
policy objects 405
recovery log volumes 496
storage volumes 282
QUERY ACTLOG command 524, 633
QUERY ADMIN command 352
QUERY BACKUPSET command 416
QUERY BACKUPSETCONTENTS command 417
QUERY CONTENT command 284
QUERY COPYGROUP command 406, 628
QUERY DB command 501, 504
QUERY DBBACKUPTRIGGER command 655
QUERY DBVOLUME command 501, 647
QUERY DEVCLASS command 618
QUERY DOMAIN command 407
QUERY DRIVE command 188
QUERY DRMSTATUS command 688
QUERY ENABLED command 540
QUERY EVENT command 32, 445, 476
QUERY FILESPACE command 329
QUERY LIBRARY command 186
QUERY LICENSE command 459
QUERY LOG command 506
QUERY LOGVOLUME command 501, 647
QUERY MGMTCLASS command 406
QUERY MOUNT command 186
QUERY NODE command 323
QUERY OCCUPANCY command 291, 292, 293
QUERY OPTION command 516
QUERY POLICYSET command 407
QUERY PROCESS command 466, 515, 632
identification numbers of migration processes 288
information about data movement process 296
QUERY REQUEST command 184
QUERY RESTORE command 347
QUERY RPFCONTENT command 697
QUERY RPFILE command 697
QUERY SCHEDULE command 436
QUERY SCRIPT command 482
QUERY SERVERGROUP command 600
QUERY SESSION command 343, 514
QUERY STATUS command 516
QUERY STGPOOL command 279, 287, 289
QUERY SUBSCRIPTION command 591
QUERY SYSTEM command 517
QUERY VOLHISTORY command 657
QUERY VOLUME command 281, 282, 297
QUERYAUTH server option 348

R

random mode for libraries 93
randomize, description of 450
read-only access mode 231
read/write access mode 231
rebinding
description of 371
file to a management class 371
recalling a file
selective 363
transparent 363
receiver 526
RECLAIM STGPOOL command 270
reclamation
aggregate reconstruction 267
controlling duration 270
delayed start of process 267
delaying reuse of volumes 276, 651
description of 23
effects of collocation 276
effects of DELETE FILESPACE 267
multiple concurrent processes
copy storage pools 274
primary sequential access storage pools 222, 270
offsite volume
controlling when reclamation occurs 274
limiting the number to be reclaimed 275
problems, diagnosing and fixing 269
setting a threshold for sequential storage pool 222, 267, 302
starting reclamation manually 270
storage pool for 223
virtual volumes 272
with single drive 271
RECONCILE VOLUMES command 606
reconstructing aggregates 267, 274, 297
recovering storage pools 648
recovering the database 661
recovery from disaster
See disaster recovery
recovery instructions file 722
recovery log
adding space to 498, 499
automating increase of 497
available space 492, 495
buffer pool 506
consistent database image 489
defining a volume 499
defining mirrored volumes 645
deleting a volume 502
deleting space 501
description of 32, 489
determining how much space is allocated 492, 495

recovery log *(continued)*
estimating the amount of space needed 494
logical volume 492, 495
managing 489
mirroring 642, 645
monitoring space 492, 495
monitoring the buffer pool 506
optimizing performance 503
querying the buffer pool 506
querying volumes 496
reducing capacity 502
size of 653
space trigger 496, 497
storage pool size effect 489
viewing information about 506
when to backup 640, 645, 654
recovery log mode
normal 653, 655
roll-forward 653, 655
setting 653
recovery log volumes
adding 39
modifying 39
viewing 39
recovery plan file
break out stanzas 717
creating 696
example 720
prefix 690
stanzas 717
recovery, disaster
auditing storage pool volumes 679
example recovery procedures 680
general strategy 601, 639, 640
media 695
methods 601, 639
providing 601, 639
when to backup 640, 654
REDUCE DB command 502
REDUCE LOG command 502
REGISTER ADMIN command 351
REGISTER LICENSE command 459
REGISTER NODE command 327
registering
client option sets 312
workstation 313
registration
description of 311
licensing for a client node 457
licensing for an administrator 457
managing client node 310, 318
setting for a client node 311
source server 312
remote access to clients 325
removable file system device
labeling requirements 131
REMOVABLEFILE device type, defining and updating 203
support for 129, 203
REMOVE ADMIN command 352
REMOVE NODE command 321
RENAME ADMIN command 351
RENAME FILESPACE command 631
RENAME NODE command 320
RENAME SCRIPT command 483
RENAME SERVERGROUP command 600
RENAME STGPOOL command 300
renaming
administrator ID 351
NAS node 164
storage pool 300
reorganizing the database 509
reporting ANR9999D messages 527
REQSYSAUTHOUTFILE server option 348
RESET BUFPOOL command 504
RESET DBMAXUTILIZATION command 493, 495
RESET LOGCONSUMPTION command 653
RESET LOGMAXUTILIZATION command 493, 495
resetting
administrative password 351
buffer pool statistic 504
database and recovery log volume utilization counter 495
user password expiration 354
RESOURCETIMEOUT server option 102
restartable restore session, client
canceling 347
interrupting, active 347
requesting information about 347
restarting the server 463
RESTORE STGPOOL command 649, 681
restore to point-in-time, enabling for clients 404
RESTORE VOLUME command 683
RESTOREINTERVAL server option 347, 361, 392
restoring
clients, optimizing restore 420
database
point-in-time 661
to its most current state 666
file 362
storage pools with incomplete volumes 651
restriction
ASCII characters in administrative Web interface 478
drive cleaning 190
serial number detection 142
retain extra versions, description of 359, 383, 386
retain only version, description of 359, 383, 386
retention grace period
description of archive 379
description of backup 378
for backup sets 414
using archive 379
using backup 378
RETEXTRA parameter 359, 383
RETONLY parameter 359, 383
retrieval date for files 249
retrieval from archive
archive package 426, 427
file 362
performance 427
reuse of sequential volumes
delaying 276, 651
storage pool volumes 176
volume pending state 284
roll-forward recovery
database backup trigger 666
mirroring recovery log 666
recovery log 666
ROLLBACK command 488
routing commands to servers 597
RUN command 484

S

SAN (storage area network)
  client access to devices 73
  device changes, detecting 142
  LAN-free data movement 73
  NDMP operations 74, 143
  policy for clients using LAN-free data movement 402
  sharing a library among servers 71, 107, 117
  storage agent role 73
scale capacity 202
schedule
  administrative command 471
  associating client node with 435
  checking the log file 442
  coordinating 447
  copying 442, 476
  day of the week 473
  defining 434, 473
  deleting 442, 476
  description of 433
  expiration date 474
  failed, querying 437, 445
  for NAS file server backup 160
  frequency of service 474
  initial start date 473
  initial time 473
  mode, setting 447
  priority 474
  querying 436
  results of 445, 476
  server administrative command 471
  startup window 449, 473
  type of action 474
  uncertain status 446, 477
  updating 473
  viewing information about 436
schedule event
  managing 444, 476
  querying 445, 476
  viewing information about 445, 476
scheduled operations, setting the maximum 449
scheduler workload, controlling 449
schedules for client nodes
  adding 39
  modifying 39
  viewing 39
scheduling
  administrative commands 32
  client 30
scheduling mode
  client-polling 447
  overview of 447
  selecting 447
  server-prompted 447
  setting on a client node 449
  setting on the server 447
scratch category, 349X library 111
scratch volume
  deleting 228, 657
  description 68
  FILE volumes 90
  number allowed in a storage pool 221, 301
  using in storage pools 230
  WORM volumes 175
script, scheduling on client 437
script, server
  continuation characters 479
script, server *(continued)*
  copying 482
  defining 478
  deleting 484
  EXIT statement 481
  GOTO statement 481
  IF clause 480
  querying 482
  renaming 483
  routing commands in 598
  running 484
  running commands in parallel 479
  running commands serially 479
  substitution variables 480
  updating 481
  used with SNMP 533
  Web browser, restricted to ASCII entry 478
scripts, server
  adding 39
  modifying 39
  viewing 39
SCSI
  See also library
  automatic labeling of volumes 169
  library with different tape technologies 198
SEARCHMPQUEUE server option 102
security
  client access, controlling 327
  features, overview 25
  for the server 348
  locking and unlocking administrators 352
  locking and unlocking nodes 320
  managing access 348, 350
  password expiration for nodes 354
  privilege class authority for administrators 348
  server options 348
SELECT command 518
selective backup 362, 374
selective recall 363
SELFTUNEBUFPOOLSIZE option 469
sending commands to servers 597
sequence number 341, 342
sequential mode for libraries 93
sequential storage pool
  auditing a single volume in 677
  auditing multiple volumes in 677
  collocation 264
  estimating space 278
  migration threshold 244
  reclamation 267
SERIAL command 479
serial number
  automatic detection by the server 138, 140, 142
  for a drive 139
  for a library 138, 140
serialization parameter 359, 360, 381, 387
server
  backing up subfiles on 418
  canceling process 466
  changing the date and time 464
  deleting 574
  description of 3
  disabling access 346
  disaster recovery 35
  enabling access 346
  halting 462
  importing subfiles from 419

server *(continued)*
maintaining, overview 19
managing multiple 33
managing operations 457
managing processes 465
messages 527
multiple instances 461
network of IBM Tivoli Storage Manager 33, 563
options
adding or updating 469
prefix 597
protecting 35
querying about processes 466, 515
querying options 516
querying status 516
restarting 463
running multiple servers 461
setting the server name 468
starting 460
stopping 462
updating 574
viewing information about 516
viewing information about processes 466, 515
server console, description of 348
SERVER device type 601
server group
copying 600
defining 599
deleting 600
member, deleting 601
moving a member 601
querying 600
renaming 600
updating description 600
server option
3494SHARED 102
ACSLS options 102
ASSISTVCRRECOVERY 102
AUDITSTORAGE 459, 469
BUFPOOLSIZE 504
changing with SETOPT command 469
COMMTIMEOUT 344, 345
DEVCONFIG 658
DRIVEACQUIRERETRY 102
EXPINTERVAL 392
EXPQUIET 392
IDLETIMEOUT 344, 345, 515
LOGPOOLSIZE 505
MOVEBATCHSIZE 469
MOVESIZETHRESH 469
NOPREEMPT 102, 467
NORETRIEVEDATE 249
overview 20
QUERYAUTH 348
REQSYSAUTHOUTFILE 348
RESOURCETIMEOUT 102
RESTOREINTERVAL 347, 361, 392
SEARCHMPQUEUE 102
SELFTUNEBUFPOOLSIZE 469
THROUGHPUTDATATHRESHOLD 345
THROUGHPUTTIMETHRESHOLD 345
TXNGROUPMAX 234
VOLUMEHISTORY 657
server script
continuation characters 479
copying 482
defining 478
server script *(continued)*
deleting 484
EXIT statement 481
GOTO statement 481
IF clause 480
querying 482
renaming 483
routing commands in 598
running 484
running commands in parallel 479
running commands serially 479
substitution variables 480
updating 481
used with SNMP 533
Web browser, restricted to ASCII entry 478
server scripts
adding 39
modifying 39
viewing 39
server storage
client files, process for storing 5
concepts overview 15
considering user needs for recovering 81
deleting files from 304
evaluating 70
example 219
managing 20
monitoring 671
planning 70
protection, methods 174, 640
server options affecting 102
tailoring definitions 627
using another IBM Tivoli Storage Manager server 601
using disk devices 83
using the storage hierarchy 237
server-free data movement
not available in Administration Center 55
server-prompted scheduling 447
server-to-server communication
setting up 49
server-to-server communications, establishing
enterprise configuration 568
enterprise event logging 568
virtual volumes 574
server-to-server virtual volumes
reclaiming 272
using to store data 601
servers
adding 39
modifying 39
viewing 39
session
canceling 344
negative number 347
server-initiated 319
setting the maximum percentage for scheduled operations 449
session, client
canceling 344
DSMC loop 343
held volume 343
managing 343
querying 343, 514
viewing information about 343, 514
SET ACCOUNTING command 541
SET ACTLOGRETENTION command 525
SET AUTHENTICATION command 355

SET CLIENTACTDURATION command 453
SET CONFIGMANAGER command 577, 579
SET CONFIGREFRESH command 591
SET CONTEXTMESSAGING command 527
SET CROSSDEFINE command 570, 573
SET DRMCHECKLABEL command 691
SET DRMCOPYSTGPOOL command 689
SET DRMCOURIERNAME command 691
SET DRMDBBACKUPEXPIREDAYS command 691
SET DRMFILEPROCESS command 691, 692
SET DRMINSTPREFIX command 689
SET DRMNOTMOUNTABLE command 691
SET DRMPLANPOSTFIX command 689
SET DRMPLANPREFIX command 690
SET DRMPRIMSTGPOOL command 688
SET DRMRPFEXPIREDAYS 697
SET DRMVAULTNAME command 692
SET EVENTRETENTION command 446, 477
SET INVALIDPWLIMIT command 355
SET LICENSEAUDITPERIOD command 459
SET LOGMODE command 656
SET MAXCMDRETRIES command 452
SET MAXSCHEDSESSIONS command 449
SET PASSEXP command 354
SET QUERYSCHEDPERIOD command 451
SET RANDOMIZE command 450
SET REGISTRATION command 311
SET RETRYPERIOD command 452
SET SCHEDMODES command 447
SET SERVERHLADDRESS command 570, 573
SET SERVERLLADDRESS command 570, 573
SET SERVERNAME command 468, 569, 570, 573
SET SERVERPASSWORD 569, 570, 573
SET SUBFILE 418
SET SUMMARYRETENTION 522
SETOPT command 469
setting
  clients to use subfile backup 419
  compression 312
  library mode 93
  password 354
  time interval for checking in volumes 201
shared access, nodes 323
shared dynamic serialization, description of 382, 387
shared file system 88
shared library 107, 117
shared static serialization, description of 381, 387
simultaneous write to primary and copy storage pools
  drives 256
  inheritance model 251
  mount points 255
  storage pools 256
  use during client storage operations 249
SnapLock
  data protection, ensuring 398
  reclamation 395
  retention periods 395
  WORM FILE volumes, setting up 398
snapshot, using in backup 9, 12
  using in directory-level backups 163
SNMP
  agent 535
  communications 536
  configuring 537
  enabled as a receiver 526, 533
  heartbeat monitor 526, 533
  manager 535
SNMP *(continued)*
  subagent 535
Software Support
  contacting xvi
  describing problem for IBM Software Support xvii
  determining business impact for IBM Software Support xvii
  submitting problem to IBM Software Support xviii
Sony WORM media (AIT50 and AIT100) 174
source server 602
space
  adding to the database or recovery log 499
  deleting from the database or recovery log 501
  planning how much to allocate 494
space management
  See Tivoli Storage Manager for Space Management
space trigger
  database space 497
  recovery log space 497
space-managed file 363
special file names 94
SQL 518
SQL activity summary table 522
standard management class, copying 381
standard storage management policies, using 359
start time, randomizing for a schedule 450
starting the server 460
startup window, description of 450
static serialization, description of 381, 387
status of a volume in an automated library 69
stopping the server 462
storage agent 73
storage area network (SAN)
  client access to devices 73
  device changes, detecting 142
  LAN-free data movement 73
  NDMP operations 74, 143
  policy for clients using LAN-free data movement 402
  sharing a library among servers 71, 107, 117
  storage agent role 73
storage devices 197
storage hierarchy 22
  defining in reverse order 224, 234
  establishing 232
  example 219
  how the server stores files in 234
  next storage pool
    definition 233
    deleting 304
    migration to 238, 287
  staging data on disk for tape storage 237
storage management policies
  description of 26, 364
  managing 357
  tailoring 376
  using standard 359
storage occupancy, querying 291
storage pool
  amount of space used 292
  auditing a volume 671
  backup
    full 648
    incremental 648
  backup and recovery 648
  comparing primary and copy types 302
  copy 219
  creating a hierarchy 232

storage pool *(continued)*
data format 145, 223, 224
defining 220
defining a copy storage pool 301
defining for disk, example 224, 234
defining for NDMP operations 156
defining for tape, example 224, 234
deleting 303
description of 218
destination in copy group 381, 387
determining access mode 221, 301
determining maximum file size 222
determining whether to use collocation 222, 258, 301
disk 23
duplicate, using to restore 651
enabling cache for disk 222, 248
estimating space for archived files on disk 278
estimating space for backed up files on disk 277
estimating space for disk 277
estimating space for sequential 278
estimating space in multiple 232
incomplete, using to restore 651
increasing sizes 23
LTO Ultrium, special considerations for 207
managing 217
monitoring 279
moving files 294
moving files between 295
multiple, using to restore 651
next storage pool
definition 233
deleting 304
migration to 238, 287
overview 66
policy use 381, 387
primary 218
querying 279
random access 218
recovery log, effect on 489
renaming 300
restoring 649, 681
sequential access 218
simultaneous write 249
updating 220
updating for disk, example 225, 234
using cache on disk 222, 248
validation of data 673
viewing information about 279
WORM media composition 175
storage pools
adding 39
modifying 39
viewing 39
storage privilege class
description 348
granting 353
reducing 354
revoking 354
storage volume
auditing 671
contents 284
formatting random access 87, 228
information about 281
labeling sequential access 167, 229
monitoring use 281
overview 67
preparing sequential access 167, 229
StorageTek devices 210
stub file 363
subfile backups
deleting 420
description of 418
example of 418
expiring 420
restoring 419
subordinate storage pool 233
subscriber, deleting 596
subscription
defining 589, 590
deleting 592
scenario 590
substitution variables, using 480
supported devices 91
swapping volumes in automated library 174
system privilege class
revoking 354

T

table of contents 152
managing 162
tape
See also volumes
capacity 213
exporting data 619
finding for client node 286
label prefix 202
monitoring life 283
number of times mounted 283
planning for exporting data 618
recording format 199
reuse in storage pools 176
rotation 77, 177
scratch, determining use 221, 230, 301
setting mount retention period 201
volumes
initializing 22
labeling 22
target server 602
threshold
migration, for storage pool 239, 244
reclamation 222, 267, 302
THROUGHPUTDATATHRESHOLD server option 345
THROUGHPUTTIMETHRESHOLD server option 345
time interval, setting for checking in volumes 201
timeout
client session 345
Tivoli Data Protection for NDMP
See NDMP operations for NAS file servers
Tivoli event console 526, 529
Tivoli Storage Manager for Space Management
archive policy, relationship to 376
backup policy, relationship to 376
description 363
files, destination for 380
migration of client files
description 363
eligibility 375
policy for, setting 375, 380
premigration 363
recall of migrated files 363
reconciliation between client and server 363
selective migration 363
setting policy for 376, 380

Tivoli Storage Manager for Space Management *(continued)*
simultaneous write, version support for 250
space-managed file, definition 363
stub file 363
Tivoli technical training xv
training, Tivoli technical xv
transactions, database 489, 490
transparent recall 363
trigger
database space 497
recovery log space 497
troubleshooting
errors in database with external media manager 133
tuning, server automatically 469
TXNBYTELIMIT client option 235
TXNGROUPMAX server option 234
type, device
3590 198
4MM 198
8MM 198
DISK 197
DLT 198
ECARTRIDGE 197
LTO 198, 207
multiple in a single library 100, 198
NAS 156
QIC 198
SERVER 198, 602, 603
VOLSAFE 210

U

Ultrium, LTO device type
device class, defining and updating 207
WORM 174, 202
unavailable access mode
description 232
marked by server 185
marked with PERMANENT parameter 185
uncertain, schedule status 446, 477
UNDO ARCHCONVERSION command 429, 431
Unicode
automatically renaming file space 333
client platforms supported 330
deciding which clients need enabled file spaces 331
description of 330
displaying Unicode-enabled file spaces 338
example of migration process 337
file space identifier (FSID) 338
how clients are affected by migration 336
how file spaces are automatically renamed 334
migrating client file spaces 332
options for automatically renaming file spaces 332
unloading the database 508
UNLOCK ADMIN command 352
UNLOCK NODE command 320
UNLOCK PROFILE command 584, 585
unplanned shutdown 462
unreadable files 640, 678, 679
unusable space for database and recovery log 492
UPDATE ADMIN command 351
UPDATE ARCHIVE command 431
UPDATE BACKUPSET command 416
UPDATE CLIENTOPT command 342
UPDATE CLOPTSET command 343
UPDATE COPYGROUP command 381, 387
UPDATE DBBACKUPTRIGGER command 656
UPDATE DOMAIN command 379
UPDATE DRIVE command 188
UPDATE LIBRARY command 186
UPDATE LIBVOLUME command 68, 179
UPDATE MGMTCLASS command 381
UPDATE NODE command 320, 337, 340
UPDATE POLICYSET command 379
UPDATE RECOVERYMEDIA command 695
UPDATE SCHEDULE command 473
UPDATE SCRIPT command 481
UPDATE SERVER command 574, 575
UPDATE VOLUME command 229
URL for client node 310
usable space 492
user exit 526
user ID, administrative
creating automatically 327
description of 310
preventing automatic creation of 328
utilization, database and recovery log
description of 493
monitoring 493, 495

V

VALIDATE LANFREE command 136
VALIDATE POLICYSET command 389
validating data
during a client session 411
for storage pool volumes 673
for virtual volumes 602
performance considerations for nodes 412
perfornance considerations for storage pools 676
variable, accounting log 541
VARY command 89
varying volumes on or off line 89
VERDELETED parameter 359, 383
VEREXISTS parameter 359, 383
versions data deleted, description of 359, 383, 386
versions data exists, description of 359, 383, 385
virtual file space mapping, command 154
virtual volumes, server-to-server
reclaiming 272
using to store data 601
VIRTUALMOUNTPOINT client option 330
Vital Cartridge Records (VCR), corrupted condition 102
VOLSAFE device class 210
volume capacity 199
volume copy
allocating to separate disks 645
description of 645
volume history
deleting information from 657
file, establishing 510, 656
using backup to restore database 656, 662
VOLUMEHISTORY option 656
volumes
access preemption 467
access, controlling 176
adding 39
allocating space for disk 87, 228
assigning to storage pool 228
auditing 181, 671
auditing considerations 671
automated library inventory 69
capacity, compression effect 214
checking in new volumes to library 171

volumes *(continued)*
checking out 180
contents, querying 284
defining for database 499
defining for recovery log 499
defining to storage pools 229
delaying reuse 276, 651
deleting 305, 657
detailed report 285
determining which are mounted 186
disk storage 229
disk storage pool, auditing 676
dismounting 186
errors, read and write 283
estimated capacity 283
finding for client node 286
help in dsmc loop session 343
inventory maintenance 176
location 284
managing 179
modifying 39
monitoring life 283
monitoring movement of data 297
monitoring use 281
mount retention time 201
moving files between 294
number of times mounted 283
offsite, limiting number to be reclaimed 223, 275
overview 68
pending status 284
private 69
querying contents 284
querying db and log volumes 496
querying for general information 281
random access storage pools 218, 230
reclamation 271
recovery using mirroring 669
removing from a library 180
restoring random-access 55
returning to a library 180
reuse delay 276, 651
scratch 68
scratch, using 230
sequential 229
sequential storage pools 167, 229
setting access mode 231
standard report 285
status, in automated library 69
status, information on 282
swapping 174
updating 179, 229
using private 68, 69
varying on and off 89
viewing 39
WORM scratch category 111

W

Web administrative interface
description 19
limitation of browser for script definitions 478
Web backup-archive client
granting authority to 327
remote access overview 325
URL 310, 326
wizard
client configuration 315
wizard *(continued)*
setup 315
workstation, registering 313
WORM devices and media
DLT WORM 174
IBM 3592 174
LTO WORM 174
Sony AIT50 and AIT100 174
special considerations for WORM media 174
VolSafe
considerations for media 174
defining VOLSAFE device classes 202
WORM FILE and SnapLock 394
WORM parameter 202
WORM scratch category for volumes in 349X library 111

Program Number: 5608-HSM
5608-ISM
5608-ISX
5608-SAN
5608-ARM

Printed in USA

GC23-4690-04

Spine information:

IBM Tivoli Storage Manager for Linux **Version 5.3** **Administrator's Guide**