Microsoft®

Microsoft® Windows® 2000 Server Operations Guide



Samsung v. Four Batons
IPR2025-00494, Exhibit 1012

Microsoft®

Microsoft® Windows® 2000 Server Operations Guide

Exhibit 1012
Page 002 of 776

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2000 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging-in-Publication Data
Microsoft Windows 2000 Server Resource Kit / Microsoft Corporation.
p. cm.
ISBN 1-57231-805-8
1. Microsoft Windows 2000 Server. 2. Operating systems (Computers). I. Microsoft Corporation.
QA76.76.O63 M5241328 2000
005.4'4769--dc21 99-045616

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 WCWC 4 3 2 1 0 9

Distributed in Canada by Penguin Books Canada Limited.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at mspress.microsoft.com.

Active Accessibility, Active Channel, Active Client, Active Desktop, Active Directory, ActiveMovie, ActiveX, Authenticode, BackOffice, DirectAnimation, DirectPlay, DirectShow, DirectSound, DirectX, DoubleSpace, DriveSpace, FrontPage, Georgia, Hotmail, IntelliMirror, IntelliSense, JScript, Links, Microsoft, Microsoft Press, MSDN, MS-DOS, MSN, Natural, NetMeeting, NetShow, OpenType, Outlook, PowerPoint, Sidewalk, Slate, Starts Here, TrueImage, Verdana, Visual Basic, Visual C++, Visual InterDev, Visual J++, Visual Studio, WebBot, Win32, Windows, Windows Media, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. NT is a trademark of Northern Telecom Limited. Other product and company names mentioned herein may be the trademarks of their respective owners.

Any RFC excerpts are subject to the following statement:
Copyright © The Internet Society (1999). All Rights Reserved. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. The example companies, organizations, products, people, and events depicted herein are fictitious. No association with any real company, organization, product, person, or event is intended or should be inferred.

Acquisitions Editor: Juliana Aldous
Project Editor: Maureen Williams Zimmerman Part No. 097-0002722

Thank you to those who contributed to this book:

Department Managers: Paul Goode, Ken Western
Documentation Managers: Laura Burris, Martin DelRe, Peggy Etchevers
Resource Kit Program Managers: Chris Hallum, Martin Holladay, Louis Kahn, Ryan Marshall, Paul Sutton

Server Operations Guide

Technical Writing Lead: Sybil Wood
Writers: Rick DeJarnette, James Klima, Brian Roberts, Audrey Wehba

Editing Leads: Deborah Annan, Jennifer Hendrix, Kate O'Leary
Book Editing Leads: Kristen Gill, Paulette McKay
Copy Editors: Kate McLaughlin, Mary Rose Sliwoski
Scott Somohano, Thelma Warren
Glossary: Daniel Bell

Resource Kit Tools Software Developers: Dan Grube, Michael Hawkins, Darryl Wood, Zeyong Xu
Documentation Tools Software Developers: Amy Buck, Tom Carey, Ryan Farber, Mark Pengra, Fred Taub

Production Leads: Sandy Dean, Jane Dow, Keri Grassl, Jason Hershey
Production Specialists: Michael Faber, Dani McIntyre, Lori Robinson

Indexing Leads: Jane Dow, Veronica Maier
Indexer: Kumud Dwivedi, Diana Rain

Lead Graphic Designer: Flora Goldthwaite
Designers: Chris Blanton, Siamack Sahafi

Art Production: Blaine Dollard, Jenna Kiter, Amy Shear, Gabriel Varela

Test Lead: Jonathan Fricke
Testers: Brian Klauber, Jeremy Sullivan

Windows 2000 Lab Manager: Edward Lafferty
Administrators: Deborah Jay, Grant Mericle, Dave Meyer, Dean Prince, Robert Thingwold, Luke Walker, Joel Wingert, Frank Zamarron
Lab Partners: Cisco Systems, Inc., Compaq, Inc., Hewlett-Packard Corporation, Intel Corporation

A special thanks to the following technical experts who contributed to and supported this effort:

Brian Andrew, Reza Baghai, Eugene Baucom, Maurizio Bellassai, June Blender, Bill Blomgren, Roger Bruist, Felipe Cabrera, Phillip Carver, Ben Christenbury, Marion Cole, Arren Conner, Buddy Cox, Joseph Dadzie, Steve DeVos, Brian Dodd, Bo Downey, David Everett, David Fields, Patrick Franklin, Jee Fung Pang, David Golds, Darrell Gorter, Scott Graff, Glenn Grant, Karen Greaves, Tim Green, Rob Greenwell, Rich Hagemeyer, Renee Hall, Scott Hallock, Jeff Heffner, Martin Holladay, Michael Jacquet, Babak Jahromi, Romano Jerez, Peter Johnston, Nikhil Joshi, Keith Kaplan, Steven Kiraly, Rama Koneru , Tom Kufeldt, Norbert Kusters, Bruce Langworthy, Patrick Lewis, Daniel Lovinger, Mark Lucovksy, Tim Lytle, Ryan Marshall, Aaron Massey, Michael McCartney, Randy McLaughlin, Ed McLees, Chris McKitterick, Michael Miele, Brett Miller, Loren Moore, Paul Moore, Alan Morris, Jackie Mounts, Steven Munk, Erik Odenborg, Steve Olsson, Michael Peterson, Ravisankar Pudipeddi, Kartik Raghavan, Aaron Reynolds, Lynn Roe, Robert Rosa, Maher Saba, Derick Schaefer, Stuart Sechrest, Joseph Seifert, Kathy Sestrap, Art Shelest, Dale Sinnott, Bob Snead, Kent Somerville, Christy Sutton, Albert Ting, Rob Trace, Tom Van Baak, Catharine Van Ingen, Landy Wang, Brad Waters, Bob Watson, Dan Weston, Steve Wilson, David Winkler, Rick Winter, Wesley Witt, Jon Wojan, Bruce Worthington, Wade Yamauchi

Contents

Introduction xxvii
Document Conventions xxviii
Resource Kit Compact Disc xxviii
Resource Kit Support Policy xxix

Part 1 Storage, File Systems, and Printing

Chapter 1 Disk Concepts and Troubleshooting 3
Basic and Dynamic Disks 5
Terms 5
Features of Basic Disk 8
Features of Dynamic Disk 8
Features Common to Both Basic and Dynamic Disks 10
Disk Sectors Critical to Startup 10
Master Boot Record 11
Master Boot Code 11
Partition Table 12
Extended Boot Record 19
Boot Sector 21
Boot Sector Startup Processes 22
Components of a Boot Sector 23
FAT16 Boot Sector 24
FAT32 Boot Sector 27
NTFS Boot Sector 31
Protecting the Boot Sector 33
Troubleshooting Disk Problems 34
Viruses 35
MBR Viruses 35
Boot Sector Viruses 35
How MBR and Boot Sector Viruses Affect Windows 2000 36
Treating an MBR or Boot Sector Virus Infection 37
Damaged MBRs and Boot Sectors 38
Using an Emergency Repair Disk 39
Using the Recovery Console 39

Other Disk Problems 41
Stop 0x0000007B — Inaccessible Boot Device 41
Volume Displays as Unknown 42
CMOS Problems 42
Cables and Connectors 43
Additional Resources 43

Chapter 2 Data Storage and Management 45
Overview of Data Management 47
Current Trends 47
Storage Features 47
Removable Storage 47
Remote Storage 49
Disk Management 49
Disk Quotas 49
Removable Storage 50
Removable Storage Service 50
Removable Storage Database 51
Removable Storage Snap-in 52
Basic Concepts 52
Media 52
Physical Locations 53
Media Pools 55
Work Queue 56
Operator Requests 57
Media Identification and Naming 58
On-media Identifiers 58
Duplicate Copies of Media 59
Bar Codes 59
Media Names 60
Media Change Detection 60
Media Handling and Usage 60
Media States 61
Side States 61
State Transitions 62
Free Media 63
Administering Removable Storage 65
Device Support 65
Installing and Configuring Removable Storage-Aware Applications 65
Preparing Media 66

Using Operator Requests in Removable Storage 66
Security 67
Controlling the Service 68
Service Startup Tasks 68
Device Configuration 68
Database Backup and Recovery 71
Using the Removable Storage Snap-in 73
Scripting 77
Troubleshooting 77
Remote Storage 78
Basic Concepts 79
Benefits 81
Local Storage Management 82
Disk Management Attribute 82
Manage Files 83
Remove Volume From Management 83
Automatic File Truncation 84
Scheduled File Truncation 84
Validate Managed Files 84
Local Storage 85
First Read, Write, and Memory Map 85
NTFS "File Open" 85
Runaway Recall Detection and Prevention 86
Deletion of a Premigrated File or Placeholder 86
Renaming, Moving, or Copying Placeholders and Premigrated Files 86
Volume Decommission 87
Restoring Placeholders from Backup and Disconnected Placeholders 87
Interaction with other Reparse Point Types 88
Placeholders Binding to a Specific Remote Storage 88
Windows 2000 Security Integration 88
File Collocation 88
Replication 88
Remote Storage Engine 88
Flow Meter Capacity Metric 89
Storage Pool 89
Handling of Physically Full Remote Storage 90
Shelf Media 90
Media Status and Location 90
Upgrading Remote Storage Libraries 90

Remote Storage Device Support 91
Kernel-Level Drivers 91
Sharing Libraries with External Applications 91
User Interfaces 91
Remote Storage Snap-in 92
Recall Notification 92
Windows Explorer Integration 93
Disk Management Integration 94
Miscellaneous Windows 2000 Shell Integration 94
Local Storage Configuration and Management 94
Local Storage Management Monitoring 95
Remote Storage Configuration and Management 95
Remote Storage Management Monitoring 95
Job Management and Monitoring 95
Setup, Installation, and License Management 95
Remote Storage and Windows 2000 96
Multiple Remote Storage Interactions 97
Moving Data Between Instances of Remote Storage to Another Remote Storage 97
Recalling Across Remote Storage Instances 97
Version Compatibility 98
Remote Storage–Aware Products 98
Backup Programs 98
Antivirus Programs 99
Document Management Programs 99
Remote Storage Protection and Recovery 100
Remote Storage Media Copies 100
Remote Storage Media Copy Sets and Set Rotation 100
Media for Media Copies 100
Remote Storage Metadata Protection 101
Disk Management 101
Fault-Tolerant Disk Management 102
Dynamic Disks 102
Disk Quotas 102
Disk Quotas and Free Space 103
Disk Quota Limits 104
Disk Quotas States 105

Administering Disk Quotas 105
Enabling Disk Quotas 106
Local and Remote Implementations 106
Auditing Disk Space Use 107
Exceeding Disk Quota Limits 107
Additional Resources 107

Chapter 3 File Systems 109
About Windows 2000 File Systems 111
FAT File System 111
FAT16 File System 111
FAT32 File System 114
File Names on FAT Volumes 116
Using FAT with Windows 2000 117
The NTFS File System 117
Multiple Data Streams 118
Reparse Points 119
Change Journal 119
Encryption 119
Sparse File Support 119
Disk Quotas 120
Distributed Link-Tracking 121
Converting to Windows 2000 File Systems 121
Multiple Booting of Windows NT and Windows 2000 122
Windows NT Compatibility with the Version of NTFS Included with Windows 2000 122
Cleanup Operations on Windows NT Volumes 122
Structure of an NTFS Volume 123
Boot Sector 124
Master File Table and Metadata 124
NTFS File Attributes 126
MS-DOS-Readable File Names on NTFS Volumes 127
Using Long File Names 127
Compact Disc File System 130
Universal Disk Format 130
Comparing FAT16, FAT32, and NTFS 130
Comparing FAT File Systems 131

FAT16 vs. FAT32 131
Advantages of FAT16 131
Disadvantages of FAT16 132
Advantages of FAT32 132
Disadvantages of FAT32 133
Advantages of NTFS 133
Disadvantages of NTFS 134
Formatting the System Partition in Multiple-Boot Configurations 134
Which Is Faster – FAT16, FAT32, or NTFS? 135
Maximum Volume Sizes 136
Maximum Sizes on FAT16 Volumes 136
Maximum Sizes on FAT32 Volumes 136
Maximum Sizes on NTFS Volumes 137
Controlling Access to Files and Folders 138
POSIX Compliance 140
File and Folder Compression 141
Compressing and Decompressing Folders and Files 141
Using the Compact Program 143
Effects of Compression on Moving and Copying Files 143
Moving Files or Folders on NTFS Volumes 144
Copying Files or Folders on NTFS Volumes 145
Moving and Copying Files Between FAT16, FAT32, and NTFS Volumes 145
Adding Files to an Almost Full NTFS Volume 146
NTFS Compression Algorithm 146
Compression Performance 147
Other Compression Methods 147
NTFS Recoverability 148
Recovering Data with NTFS 148
Caching and Data Recovery 149
Cluster Remapping 149
Features Built on Reparse Points 150
Remote Storage 150
Volume Mount Points 150
File System Tools 151
Cacls: Displays and Modifies NTFS Access Control Lists 152
Compact: Compresses and Decompresses NTFS Files and Folders 153
Volume Compression Requirements 154
Compress: Compresses Files or Folders 154

Convert: Converts a Volume from FAT to NTFS 155
Free Space Required to Convert FAT to NTFS 156
Converting NTFS and FAT Volumes 156
DirUse: Scans a Directory and Reports On Disk Space Usage 157
Expand: Expands Compressed Files 157
Mountvol: Displays, Creates, and Deletes Volume Mount Points 158
Additional Resources 158

Chapter 4 Network Printing 159
Introduction to Network Printing 161
Print Devices, Print Clients, Printers, and Print Servers 161
Local and Remote Printing 162
Web Printing 166
Architecture 166
Graphics Device Interface 169
Printer Driver 169
Client Spooler 171
Remote Print Provider 171
Print Router 172
Print Server Service 172
Server Spooler 173
Local Print Provider 173
Print Processor 175
RAW Data Type 175
NT EMF 1.00x Data Type 175
TEXT Data Type 176
Print Monitors 176
Language Monitor 176
Local Port Monitor 177
Remote Port Monitors 178
Sharing Printers 178
Local Printing 178
Add Printer Wizard 179
Plug and Play 180
Forwarding Print Jobs 181

Remote Printing 181
Prerequisites 182
Print Server Protocols Used 182
Port Installation Procedure 183
Reconfiguration 184
Status Reporting 185
Internet Printing 185
Prerequisites 186
Installing a Printer from a Web Page 186
Information Displays in HTML 187
Security for Internet Printing 187
Earlier Network-Interface Printing 190
Prerequisites 190
Creating a Port 190
Working with Other Operating Systems 191
Printing from Windows 95 and Windows 98 Clients 191
Printing from Other Microsoft Print Clients 191
Required Redirectors 192
Output Destination Rules 192
Printing with UNIX Clients and Servers Using LPR/LPD 193
Printing Documents Received Through the LPD Utility 194
Sending Documents Through the LPR Utility 195
Printing with NetWare Clients and Servers 198
Printing Documents from NetWare Clients 198
Sending Documents to a NetWare Network 199
Printing with Macintosh Clients and AppleTalk Printers 200
Printing Documents from Macintosh Clients 201
Sending Documents over AppleTalk 203
Printing and Active Directory 204
Printing and Directory Service Overview 204
Publishing Windows 2000 Printers 205
Publishing Mechanism 206
Pruning Orphans 206
Sites Without Domain Controllers 207
Supporting Non-Windows 2000 Printers 208
Group Policy Settings 208
Printer Location Tracking 209

Printing and Clusters 210
Creating the Spooler 210
Prerequisites 210
Specifying the Resource 210
Installing Printing Components 212
Adding a Printer to a Cluster 212
What Happens at Failover 213
Monitoring Printer Performance 213
Using System Monitor 213
Auditing Printing Events 214
Logging Spooler Events 216
Utilities 217
rundll32 217
PrnAdmin 218

Part 2 Performance Monitoring

Chapter 5 Overview of Performance Monitoring 223
Performance Monitoring Concepts 225
Scope of Performance Data 225
Throughput Defined 226
Queue Defined 226
Response Time Defined 226
Data Collection Architecture 226
Performance Objects 227
Performance Counters and Instances 230
Data Collection and Reporting 231
Monitoring Tools 232
System Monitor 233
Starting System Monitor 234
Working with System Monitor 235
Getting the Most from System Monitor 241
Performance Logs and Alerts 243
Starting Performance Logs and Alerts 244
Working with Logs and Alerts 245
Getting the Most from Performance Logs and Alerts 249

Task Manager 250
Comparison with System Monitor 250
Starting Task Manager 250
Working with Task Manager 250
Monitoring Processes 251
Monitoring the System 255
Resource Kit Performance Tools 258
Starting Your Monitoring Routine 258
Your Minimum Monitoring Configuration 258
Establishing the Baseline 261
Analyzing Monitoring Results 261
Identifying Potential Bottlenecks 263
Investigating Bottlenecks 265
Troubleshooting Problems with Performance Tools 267
Investigating Zero Values 267
Investigating Other Issues with Performance Tools 267
Controlling Performance Monitoring Overhead 272
Specific Monitoring Scenarios 272
Monitoring Remote Computers 272
Methods of Monitoring 273
Security Issues 274
Monitoring Servers and Services 275
Characterizing Server Workload 275
Establishing a Baseline on Server Computers 277
Upgrading Servers and Capacity Planning 278
Monitoring Legacy Applications 278
Monitoring 16-bit Windows Applications 278
Running 16-bit Windows Applications in a Separate Process 281
Monitoring MS-DOS Applications 282
Integrating the System Monitor Control into Office and Other Applications 283
Placing the Control in an Office Document or on a Web Page 283
Formatting the Control in a Document 286

Chapter 6 Evaluating Memory and Cache Usage 289
Overview of Memory Monitoring 291
Determining the Amount of Installed Memory 292
Understanding Memory and the File System Cache 293
Optimizing Your Memory Configuration 296
Adjusting Paging File Size 297

Establishing a Baseline for Memory 300
Virtual Memory Usage 301
Default Services Memory Consumption 301
Investigating Memory Problems 302
Investigating Memory Shortages 304
Investigating Disk Paging 307
Confirming Hard Page Faults 307
Assessing the Effect of Page Faults on the Disk 309
Investigating User-Mode Memory Leaks 310
Identifying a Memory Leak 311
Memory Leaks and the Nonpaged Pool 312
Investigating Kernel-Mode Memory Leaks 316
Monitoring the Cache 316
Understanding the Cache Counters 318
Interpreting Changes in Cache Counter Values 320
Resolving Memory and Cache Bottlenecks 322
Optimizing Cache Usage 322
Resolving a Memory Bottleneck 322
Additional Resources 323

Chapter 7 Analyzing Processor Activity 325
Overview of Processor Monitoring and Analysis 327
Processor Counters 328
Resource Kit Tools for Processor Monitoring 330
Establishing a Baseline for Processor Performance 330
Selecting Counters for Baseline Monitoring 330
Selecting Times for Baseline Monitoring 331
Recognizing a Processor Bottleneck 332
Examining the Processor Time Counter 333
Observing Processor Queue Length 334
Monitoring Interrupts 337
Monitoring Context Switches 338
Processes in a Bottleneck 340
Identifying Active Processes 340
Isolating Processor-Intensive Workloads 341
Reducing Single-Process Bottlenecks 342
Observing Processor Consumption by Multiple Processes 343

Threads in a Bottleneck 345
Single vs. Multiple Threads in a Bottleneck 345
Charting Processor Usage Per Thread 347
Context Switches 348
User Mode and Privileged Mode 349
Advanced Topic: Changing Thread Priority to Improve Performance 351
Examining and Adjusting Thread Priority 354
Priority Class and Priority 354
Thread Scheduling 355
Determining and Tuning Priority 356
System Monitor 357
Task Manager 358
Process Viewer 359
Start Command 359
Windows 2000 Configuration and Process Priority 359
Testing Priority Changes 360
Eliminating a Processor Bottleneck 362
Additional Resources 363

Chapter 8 Examining and Tuning Disk Performance 365
Disk Monitoring Concepts 367
Configuring the Disk and File System for Performance 367
Configuring Your File System 368
Bypassing I/O Counts 369
Configuring Disk Alignment 370
Working with Disk Counters 370
Monitoring Disk Space 374
Monitoring Disk Efficiency 376
Preparing for Comparison Testing 377
Testing Maximum Throughput 378
Testing Reading vs. Writing 379
Establishing a Baseline for Disk Usage 381
Data Collection 382
Data Analysis 384
Investigating Disk Performance Problems 385
Monitoring Paging 385
Monitoring Usage 387
Monitoring Queue Length 388
Resolving Disk Bottlenecks 389

Evaluating Cache and Disk Usage by Applications 390
Random and Sequential Data Access 391
I/O Request Size 391
Investigating Disk Usage by Applications 392

Chapter 9 Monitoring Network Performance 393
Introduction to Network Performance Analysis 395
Tools for Monitoring Network Performance 396
System Monitor 396
Network Counters 399
Trace Logs 413
Network Monitor 413
Network Monitor Features 413
Installing Network Monitor 415
Capturing Frame Data 416
Displaying Captured Data 421
Reviewing Captured Data 423
Network Monitor Performance Issues 425
Resolving Network Bottlenecks 425
Adding Network Adapters 427

Chapter 10 Measuring Multiprocessor System Activity 429
Overview of SMP Performance and Monitoring 431
Benefits of Scaling 431
Analyzing Performance on SMP Systems 432
SMP Impact on System Resources 432
Monitoring Activity on Multiprocessor Systems 433
Examining Processor Time Data 434
Observing Processor Queue Length 435
Analyzing Interrupt and DPC Activity 437
Monitoring Context Switches and Threads 438
Optimizing and Tuning Multiprocessor Installations 438
Thread Partitioning 439
DPC (Software Interrupt) Partitioning 440
Hardware Interrupt Partitioning 442
Bypassing I/O Counts 443
Upgrading or Adding Processors 443
Application Design and Multiprocessor Performance 443
Network Load Balancing and Scaling 445

Part 3 System Recovery

Chapter 11 Planning a Reliable Configuration 449
Planning Considerations 451
Approaches to Planning 452
System Maintenance 452
Planning for Security 453
Hardware and Software Upgrades 453
Contingency Planning 454
Disaster Planning 454
Assessing the Probability of Failure 455
Estimating Replacement Costs 456
Planning Tasks 457
Maintaining Configuration and System Information 457
Developing a Recovery Plan 459
Testing Your System for Possible Problems 460
Testing Recovery Procedures 460
Documenting Recovery Procedures 461
Training Personnel for Recovery 462
Creating a Set of Windows 2000 Setup Floppy Disks 462
Creating a Windows 2000 Startup Floppy Disk 464
Planning a Fault-Tolerant Disk Configuration 465
Dynamic Disks and Disk Groups 467
Converting Basic Disks to Dynamic Disks 468
Moving Disks 468
Connecting New Disks to a Computer 469
Importing Foreign Disks 469
Simple Volumes 471
Spanned Volumes 471
Striped Volumes 472
Mirrored Volumes 474
Advantages of Mirrored Volumes 475
Disadvantages of Mirrored Volumes 476
RAID-5 Volumes 476
Advantages of RAID-5 Volumes 478
Disadvantages of RAID-5 Volumes 478

Configuring and Using Mirrored Volumes and RAID-5 Volumes 479
Creating a Mirrored Volume or RAID-5 Volume 479
Configuring the System Partition on a Mirrored Volume 480
Guidelines for Configuring Mass Storage 482
Using Mirrored and RAID-5 Volumes 483
Fault-Tolerant Hardware Versus Software 484
Testing a Fault-Tolerant System 485
Avoiding Single Points of Failure 485
Avoiding Computer Component Failure 486
Motherboard and Central Processing Unit 486
RAM 486
Video Cards 487
Disks and Disk Controllers 487
Network Cards 487
Using an Uninterruptible Power Supply 488
Avoiding Disruptions to Network Connections 490
Network Cabling 490
Intermediary Devices 490
External Network Connections 491
Additional Resources 491

Chapter 12 Backup 493
Overview of Backups 495
Backup Types 495
Storage Devices and Media 495
Storage Devices 496
Media Types 496
Backup Strategies 496
Security Considerations 498
Backup and Restore Rights 499
Granting Rights at Large Sites 499
File Permissions 499
Storing Backup Media 500
Establishing a Backup Plan 501
Backup Scenarios 501
Small and Medium LANs 501
Methods for 24-Hour Operations or Very Large Backups 503

Establishing a Backup Schedule 505
Twelve-Week Schedule 505
Twelve-Month Schedule 506
Developing Backup and Restore Procedures 507
Considerations for Developing Procedures 507
Testing Backup-and-Restore Procedures 510
Documenting Backup-and-Restore Procedures 511
Conducting Verify Operations 511
Backing Up System State Data 511
Boot and System Files 514
Active Directory 514
Certificate Services Information 514
Cluster Database 515
Registry 515
Performance Counter Configuration 515
Component Services Class Registration Database 515
Using Backup 515
Selecting Backup Type 516
Selecting Backup Media 517
Files Skipped During Backup 517
Files Skipped by Default 518
Locked Files 518
Encrypted Files 518
Backing Up Files on Your Local Computer 518
Logon Scripts 518
User Data 519
Application Programs 519
Archived Data 519
Backing Up Files on Remote Computers 520
Backing Up Windows 2000 Services and NTFS Features 520
Remote Storage Considerations 520
Backing Up Remote Storage Databases 521
Protecting Remote Storage Media 521
Remote Storage Self-Backup 522
Exchange Database 522
Backing up Exchange 523
Restoring Exchange 523
SQL Database Backup 523
Logical Volume Information 523

Chapter 13 Repair, Recovery, and Restore 525
Repairing a Windows 2000 Installation 527
Using the Recovery Console 527
Starting the Recovery Console 528
Supported Commands 531
Replacing the Registry Using the Recovery Console 540
Recreating Boot.ini Using the Recovery Console 540
Using the Emergency Repair Disk 540
Creating the Emergency Repair Disk 541
Emergency Recovery Process 542
If the Emergency Repair Process Does Not Fix Your System 543
Restoring System State Data 544
Restoring the Registry 545
Restoring Data 546
Restoring Files from Third-Party Backup Programs 546
Restoring File Security Settings 546
Recovering Your Disk Configuration 546
Recovering a Mirrored or RAID-5 Volume 547
Recovering a Mirrored Volume 547
Repairing a Basic Mirrored Volume 548
Replacing a Failed Mirror 548
Breaking a Mirrored Volume 549
Deleting Mirrored Volumes on a Basic Disk 549
Resynchronizing Mirrored Volumes 549
Removing Mirrored Volumes 549
Recovering a RAID-5 Volume 550
Replacing a Stripe in the RAID-5 Volume 550
Deleting a RAID-5 Volume on a Basic Disk 551
Repairing a RAID-5 Volume on a Basic Disk 551
Recovering Data in Remote Storage 551
Using Rstore.exe 552
Restoring the Remote Storage and Removable Storage Databases 552
Restoring the Removable Storage Databases 553
Restoring the Remote Storage Database After Completely Reinstalling Windows 2000 554
Restoring a Damaged Remote Storage Database 555
Recovering Managed Files 557
Recovering from Bad Media 558
Additional Resources 559

Part 4 Troubleshooting

Chapter 14 Troubleshooting Strategies 563
Technical Support and Services 565
Project Ascent 565
Other Resources 566
Troubleshooting Tools for Windows 2000 567
System Information 567
System Information Categories 568
Tools Menu 570
Device Manager 573
AVBoot 574
System File Checker 574
Windows Update 576
NetDiag 577
IPConfig 577
NBTStat 579
PathPing 579
IPSecMon 580
Troubleshooting Options 580
Stop Messages 580
Event Viewer 581
Event Logs 581
Event Descriptions 582
Viewing Events 582
Error and Event Messages Help 583
System Monitor 584
Troubleshooters 584
Knowledge Base 585
Windows 2000 Registry 585
Registry Editors 586
Updating Registry Information 587
Troubleshooting with the Registry 588
HKEY_LOCAL_MACHINE\SYSTEM 589
Using the LastKnownGood Configuration 590
Select Subkey 592
Finding Service and Device Dependencies 592

Troubleshooting Suggestions 595
Keep Records 595
Keep Notes 595
Refer to Documentation 596
Finding Information About the Problem 596
Setting up the System for Further Troubleshooting 596
Local Stop File Creation 597
Sending the Dump File to Product Support Services 597
Troubleshooting Techniques 597
Follow the System Checklist 598
Analyzing Symptoms and Factors 598
Checking for Common Issues 600
Isolating the Source of the Problem 601
Troubleshooting Ideas 602
Test the Affected Feature 604
Developing an Action Plan 604
Contacting Technical Support Services 604
Troubleshooting Hardware Problems 605
Hardware Compatibility List 605
Other Approaches to Troubleshooting Hardware Problems 606
Troubleshooting Specific Problems 607
Troubleshooting Setup 607
Troubleshooting Stop Messages 607
Troubleshooting the Startup Process 608
Troubleshooting Networking Problems 608
Troubleshooting Video Problems 609
Troubleshooting Services and Drivers 610
Troubleshooting Problems with Portable Computers 612
Additional Resources 612

Chapter 15 Startup Process 613
Windows 2000 Startup Requirements 615
Starting Windows 2000 615
Power-On Self Test Process 616
Initial Startup Process 616
Bootstrap Loader Process 617
Signature() Syntax 617
NTLDR 618

Selecting the Operating System 619
Boot.ini Switches 620
Editing Boot.ini 621
Detecting Hardware 622
Loading and Initializing the Kernel 623
Selecting the Control Set 623
Loading Device Drivers 624
Initializing Device Drivers 624
Error Processing 625
Logging On to Windows 2000 625
Troubleshooting Startup Problems 625
Problem Occurs Before the Bootstrap Loader Starts 626
Problem Occurs After the Bootstrap Loader Starts 627
Hardware Problems 628
Options to Use When a System Does Not Start 629
Safe Mode 630
Recovery Console 633
Emergency Repair Disk 634
Additional Resources 635

Chapter 16 Windows 2000 Stop Messages 637
System Messages 639
Stop Messages 639
Stop Message Screen Sections 640
Bugcheck Information 640
Recommended User Action 641
Debug Port Information 641
Types of Stop Messages 641
General Stop Messages 641
Stop Messages During Installation 642
Stop Messages That Occur Only at Executive Initialization 642
Stop Messages Caused by Software Traps 644
Troubleshooting Stop Messages 644
Generic Troubleshooting Procedures 645
Troubleshooting Common Stop Messages 646
Stop 0x0000000A or IRQL_NOT_LESS_OR_EQUAL 646
Interpreting the Message 646
Resolving the Problem 647

Stop 0x0000001E or KMODE_EXCEPTION_NOT_HANDLED 648
Interpreting the Message 648
Resolving the Problem 649
Stop 0x00000024 or NTFS_FILE_SYSTEM 650
Interpreting the Message 650
Resolving the Problem 650
Stop 0x0000002E or DATA_BUS_ERROR 652
Interpreting the Message 652
Resolving the Problem 652
Stop 0x00000050 or PAGE_FAULT_IN_NONPAGED_AREA 655
Interpreting the Message 655
Resolving the Problem 655
Stop 0x00000077 or KERNEL_STACK_INPAGE_ERROR 657
Interpreting the Message 657
Resolving the Problem 658
Stop 0x00000079 or MISMATCHED_HAL 659
Interpreting the Message 659
Resolving the Problem 659
Stop 0x0000007A or KERNEL_DATA_INPAGE_ERROR 660
Interpreting the Message 660
Resolving the Problem 661
Stop 0x0000007B or INACCESSIBLE_BOOT_DEVICE 662
Interpreting the Message 662
Resolving the Problem 663
Stop 0x0000007F or UNEXPECTED_KERNEL_MODE_TRAP 665
Interpreting the Message 665
Resolving the Problem 666
Stop 0xC000021A or STATUS_SYSTEM_PROCESS_TERMINATED 668
Interpreting the Message 668
Resolving the Problem 668
Stop 0xC0000221 or STATUS_IMAGE_CHECKSUM_MISMATCH 669
Resolving the Problem 670
Hardware Malfunction Messages 672
Additional Resources 673

Glossary 675

Index 701

Introduction

Welcome to the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide.*

The *Microsoft® Windows® 2000 Server Resource Kit* consists of seven volumes and a single compact disc (CD) containing tools, additional reference materials, and an online version of the books. Supplements to the *Windows 2000 Server Resource Kit* will be released as new information becomes available, and updates and information will be available on the Web on an ongoing basis.

The *Server Operations Guide* provides the information you need to understand and maintain your server installations. This guide includes descriptions of the disk, file system, storage, printing, troubleshooting, and system recovery features that are included with Microsoft® Windows® 2000. Particular attention is given to the new storage features, such as Removable Storage and Remote Storage, and characteristics new to NTFS, such as reparse points and disk quotas. This guide also covers troubleshooting and system recovery tools, strategies, and recommendations. With the exceptions of printing and performance monitoring, the information in this guide pertains to stand-alone computers. See the other volumes in the *Windows 2000 Server Resource Kit* for information about networked systems.

Document Conventions

The following style conventions and terminology are used throughout this guide.

Element	Meaning
bold font	Characters that you type exactly as shown, including commands and switches. User interface elements are also bold.
Italic font	Variables for which you supply a specific value. For example, *Filename.ext* could refer to any valid file name for the case in question.
`Monospace font`	Code samples.
`%SystemRoot%`	The folder in which Windows 2000 is installed.

Reader Alert	Meaning
Tip	Alerts you to supplementary information that is not essential to the completion of the task at hand.
Note	Alerts you to supplementary information.
Important	Alerts you to supplementary information that is essential to the completion of a task.
Caution	Alerts you to possible data loss, breaches of security, or other more serious problems.
Warning	Alerts you that failure to take or avoid a specific action might result in physical harm to you or to the hardware.

Resource Kit Compact Disc

The *Windows 2000 Server Resource Kit* companion CD includes a wide variety of tools and resources to help you work more efficiently with Windows 2000.

Note The tools on the CD are designed and tested for the U.S. version of Windows 2000. Use of these programs on other versions of Windows 2000 or on versions of Microsoft® Windows NT® can cause unpredictable results.

The *Resource Kit* companion CD contains the following:

Windows 2000 Server Resource Kit Online Books An HTML Help version of the print books. Use these books to find the same detailed information about Windows 2000 as is found in the print versions. Search across all of the books to find the most pertinent information to complete the task at hand.

Windows 2000 Server Resource Kit Tools and Tools Help Over 200 software tools, tools documentation, and other resources that harness the power of Windows 2000. Use these tools to manage Active Directory™, administer security features, work with the registry, automate recurring jobs, and many other important tasks. Use Tools Help documentation to discover and learn how to use these administrative tools.

Windows 2000 Resource Kit References A set of HTML Help references:

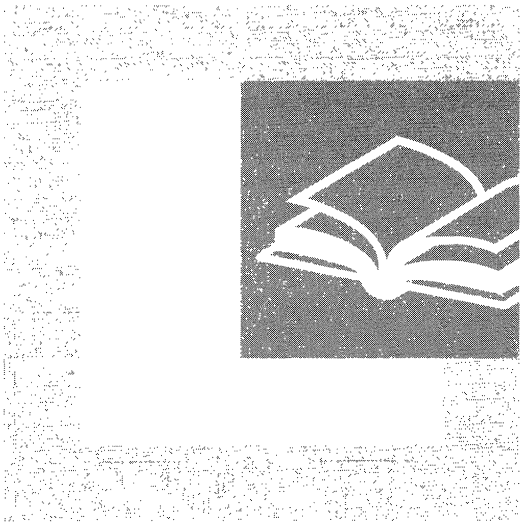
- **Error and Event Messages Help** contains most of the error and event messages generated by Windows 2000. With each message comes a detailed explanation and a suggested user action.
- **Technical Reference to the Registry** provides detailed descriptions of Windows 2000 registry content, such as the subtrees, keys, subkeys, and entries that advanced users want to know about, including many entries that cannot be changed by using Windows 2000 tools or programming interfaces.
- **Performance Counter Reference** describes all performance objects and counters provided for use with tools in the Performance snap-in of Windows 2000. Use this reference to learn how monitoring counter values can assist you in diagnosing problems or detecting bottlenecks in your system.
- **Group Policy Reference** provides detailed descriptions of the Group Policy settings in Windows 2000. These descriptions explain the effect of enabling, disabling, or not configuring each policy, as well as explanations of how related policies interact.

Resource Kit Support Policy

The software supplied in the *Windows 2000 Server Resource Kit* is not supported. Microsoft does not guarantee the performance of the *Windows 2000 Server Resource Kit* tools, response times for answering questions, or bug fixes to the tools. However, we do provide a way for customers who purchase the *Windows 2000 Server Resource Kit* to report bugs and receive possible fixes for their issues. You can do this by sending e-mail to rkinput@microsoft.com. This e-mail address is only for *Windows 2000 Server Resource Kit* related issues. For issues relating to the Windows 2000 operating system, please refer to the support information included with your product.

PART 1

Storage, File Systems, and Printing



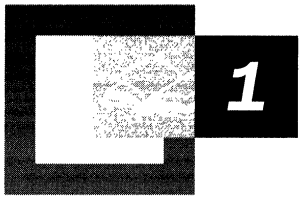
Microsoft® Windows® 2000 provides significant enhancements to data storage, NTFS and FAT32 file systems, and network printing. Part 1 describes the storage, file system, and printing features and tools in Windows 2000 and provides information that can be used during troubleshooting.

In This Part

Disk Concepts and Troubleshooting 3
Data Storage and Management 45
File Systems 109
Network Printing 159

CHAPTER 1

Disk Concepts and Troubleshooting



Understanding the organizational structure of information on hard disks, as well as being familiar with disk terminology, is key to diagnosing and troubleshooting disk problems. When troubleshooting system problems, you can refer to this chapter for detailed descriptions of the master boot record (MBR) and boot sector, two disk sectors that are critical to the startup process.

In This Chapter

Basic and Dynamic Disks 5
Disk Sectors Critical to Startup 10
Troubleshooting Disk Problems 34

Related Information in the Resource Kit

- For more information about startup, see "Startup Process" in this book.
- For more information about file systems, see "File Systems" in this book.
- For more information about preparing for and performing recovery, see "Repair, Recovery, and Restore" in this book.

Basic and Dynamic Disks

Microsoft® Windows® 2000 offers two types of disk storage configurations: basic disk and dynamic disk. Basic disk is similar to the disk structures used in Microsoft® Windows NT®. Dynamic disk is new to Windows 2000. By default, Windows 2000 initializes hard disks as basic disk.

The Disk Administrator tool found in Microsoft® Windows NT® version 4.0 and earlier has been replaced in Windows 2000 with the Disk Management snap-in for Microsoft Management Console (MMC). Disk Management supports both basic and dynamic disks. You can use the upgrade wizard in Disk Management to convert hard disks to dynamic disks.

You can use both basic and dynamic disks on the same computer system, and with any combination of file systems (file allocation table [FAT], including FAT16 and FAT32, and NTFS file system). However, all volumes on a physical disk must be either basic or dynamic.

You can upgrade from basic to dynamic storage at any time. Any changes made to your disk are immediately available in Windows 2000—you do not need to quit Disk Management to save them or restart your computer to implement them. However, if you upgrade the startup disk to dynamic, or if a volume or partition is in use on the disk that you are upgrading, the computer must be restarted for the upgrade to succeed.

Terms

To help you understand the differences between basic disk and dynamic disk, a set of definitions are provided.

Basic Disk

A basic disk is a physical disk that contains primary partitions and/or extended partitions with logical drives used by Windows 2000 and all versions of Windows NT. Basic disks can also contain volume, striped, mirror, or redundant array of independent disks (RAID) Level 5 (also known as striped with parity) sets that were created using Windows NT 4.0 or earlier. As long as a compatible file format is used, basic disks can be accessed by Microsoft® MS-DOS®, Microsoft® Windows® 95, Microsoft® Windows® 98, and all versions of Windows NT.

Since Windows 2000 automatically initializes disks as basic, you can troubleshoot partitions and volumes using the same methods as in Windows NT.

Note FAT32 is new in Windows 2000. Disk troubleshooting tools from Windows NT will likely not recognize FAT32 boot sectors and may cause problems with FAT32-formatted volumes. If FAT32 is used on your computer, be sure to use a disk troubleshooting tool designed for Windows 2000 that recognizes this file format.

New or empty disks can be initialized as either basic or dynamic after the hardware installation is complete. However, to set up a new fault-tolerant (FT) disk system, you must upgrade to dynamic disk.

Basic Volume

A basic volume is a volume on a basic disk. Basic volumes include primary partitions, logical drives within extended partitions, as well as volume, striped, mirror, or RAID-5 sets created using Windows NT 4.0 or earlier. You cannot create basic volumes on dynamic disks.

Note Creating new FT sets, such as mirrored and RAID-5 volumes, is only available on computers running Windows 2000 Server. The disk must be upgraded to dynamic disk before these volumes can be created. You can, however, use a computer running Windows 2000 Professional to create mirrored and RAID-5 volumes on a remote computer running Windows 2000 Server.

Dynamic Disk

A dynamic disk is a physical disk that has been upgraded by and is managed with Disk Management. Dynamic disks do not use partitions or logical drives. They can contain only dynamic volumes created by Disk Management. Only computers running Windows 2000 can access dynamic volumes.

Note Disks that have been upgraded from basic to dynamic disk still contain references to partitions in the partition table of the MBR. However, the MBR's reference to these partitions identifies the partition types as dynamic, indicating to Windows 2000 that the disk configuration data is now maintained in the disk management database at the end of the disk. Furthermore, any new changes made to the disk, such as deleting existing or creating additional volumes, are not recorded in the partition table.

Dynamic disks use dynamic volumes to subdivide physical disks into one or more drives enumerated by letters of the alphabet. Disk configuration data is contained in a disk management database stored in the last 1 megabyte (MB) of space at the end of the disk. Since dynamic disks do not use the traditional disk organization scheme of partitions and logical volumes, they cannot be directly accessed by MS-DOS, Windows 95, Windows 98, or any versions of Windows NT. Disk shares on dynamic disks, however, are available to computers running all of these operating systems.

Dynamic Volume

A dynamic volume is a logical volume that is created on a dynamic disk using Disk Management. Dynamic volume types include simple, spanned, striped, mirrored, and RAID-5, although only Windows 2000 Server supports the FT volume types (mirrored and RAID-5). You cannot create dynamic volumes on basic disks. Dynamic volumes are not supported on portable computers or removable media.

Note Dynamic volumes that were upgraded from basic disk partitions cannot be extended. This specifically includes the system volume, which contains hardware-specific files needed to start Windows 2000, and the boot volume, which contains the Windows 2000 system files required for startup. Only volumes created after the disk was upgraded to dynamic can be extended.

Partitions and Volumes

When you upgrade to dynamic disk, existing partitions and logical volumes are converted into dynamic volumes. Table 1.1 illustrates the translation of terms between basic and dynamic disk structures.

Table 1.1 Translation of Terms Between Basic and Dynamic Disk

Basic Disk Organization	Dynamic Disk Organization
Primary partition	Simple volume
System and boot partitions	System and boot volumes
Active partition	Active volume
Extended partition	Volume and unallocated space
Logical drive	Simple volume
Volume set	Spanned volume
Stripe set	Striped volume
Stripe set with parity	RAID-5 volume
Mirror set	Mirrored volume

Features of Basic Disk

You can use partitions on a basic disk just as you did with Microsoft® Windows NT® Server version 4.0, but you do not need to commit changes to save them or to restart your computer to make the changes effective. Changes made by Disk Management are implemented immediately. Unless you are making a change that affects existing files on the disk, the system executes your change without confirmation.

You can create up to four partitions in the free space on a physical hard disk; one of these can be an extended partition. You can use the free space in the extended partition to create one or more logical drives. You cannot use basic disk to create any kind of multiple volume sets or FT volumes.

You can perform the following tasks only on a basic disk:

- Create and delete primary and extended partitions.
- Create and delete logical drives within an extended partition.
- Format a partition and mark it as active.
- Delete volume, striped, mirror, or stripe sets with parity.
- Break a mirror from a mirror set.
- Repair failed legacy FT volumes such as mirror sets or stripe sets with parity.

Certain legacy functions are no longer available on basic disks because multiple-disk storage systems need to use dynamic disks. Disk Management supports legacy volumes that exceed a single partition on more than one hard disk, but it does not allow you to create new ones. For example, you cannot create volume, striped, mirror, or RAID-5 sets or extend volumes and volume sets on a basic disk.

While you cannot create new multiple disk sets on basic disks, you can delete them. Be sure to back up all the information on the set before you delete it.

To establish a new spanned, striped, mirror, or RAID-5 set, first upgrade the disk to dynamic disk. To convert an existing volume, striped, mirror, or RAID-5 set, upgrade the physical disks on which the set resides to dynamic disk.

Features of Dynamic Disk

Disk Management is very flexible. The number of volumes that you can create on a physical hard disk is limited only by the amount of available free space on the disk. You can also create volumes that span two or more disks and that, if you are running Windows 2000 Server, are fault tolerant.

You can perform the following tasks only on a dynamic disk:

- Create and delete simple, spanned, striped, mirrored, and RAID-5 volumes.
- Extend a simple or spanned volume.
- Remove a mirror from a mirrored volume or break the mirrored volume into two volumes.
- Repair mirrored or RAID-5 volumes.
- Reactivate a missing or offline disk.

Dynamic disks are not supported on portable computers. If you are using a portable computer and right-click a disk in the graphical or list view in Disk Management, you will not see the option to upgrade the disk to dynamic.

Note On some older and non-Advanced Configuration and Power Interface (ACPI)–compliant portable computers, you might be able to upgrade to dynamic disk, but it is neither recommended nor supported.

The limitations of dynamic volumes occur in the following situations:

When installing Windows 2000 If a dynamic volume is created from unallocated space on a dynamic disk, you cannot install Windows 2000 on that volume.

The setup limitation occurs because Windows 2000 Setup uses BIOS calls that only recognize volumes listed in the partition table. Only basic disk partitions, as well as simple and mirrored volumes of dynamic disks that were upgraded from basic disk partitions, appear in the partition table. Dynamic disk does not use the partition table to manage its volumes, so new dynamic volumes are not registered in the partition table as they are created. Windows 2000 must be installed on a volume that is correctly represented in the partition table.

When extending a volume You can install Windows 2000 on a dynamic volume that was upgraded from a basic disk partition, but you cannot extend either the system volume or the boot volume. Neither can be part of a spanned volume, since Windows 2000 considers extended volumes to be the same as spanned volumes.

Windows 2000 cannot extend a dynamic volume that was a basic volume before the dynamic disk upgrade, and the system and boot volumes (which might be one and the same) are likely the same volumes that existed under basic disk. The upgraded simple volume must match the listing found in the partition table. Extending the dynamic volume changes its size, but the partition table's registration of that volume is not updated to reflect the change. The only dynamic volumes on which you can install Windows 2000 are simple and mirrored volumes, and since these volumes must be registered in the partition table, they must be upgraded from basic to dynamic.

Features Common to Both Basic and Dynamic Disks

You can perform the following tasks on both basic and dynamic disks:

- Check disk properties such as capacity, available free space, and current status.
- View volume and partition properties such as size, drive-letter assignment, label, type, and file system.
- Establish drive-letter assignments for disk volumes or partitions, and for CD-ROM devices.
- Establish disk sharing and security arrangements for a volume or partition.
- Upgrade a basic disk to dynamic or revert a dynamic disk to basic.

Disk Sectors Critical to Startup

The two sectors critical to starting your computer are the master boot record (MBR), which is always located at sector 1 of cylinder 0, head 0, the first sector of a hard disk, and the boot sector, which resides at sector 1 of each volume. These sectors contain both executable code and the data required to run the code.

Note The use of basic or dynamic disk does not affect where the MBR is located on disk and only minor differences exist between the two for how the partition table is configured. However, as the Disk Management database contains the information where dynamic volumes begin and end, the method of walking, or navigating, partition tables to find the start and end of partitions and logical volumes, as well as finding volume boot sectors, does not work on dynamic disks. Disk editing tools, such as DiskProbe and third-party tools, can walk the partitions as expected with basic disks. Also, many disk editor tools that work with Windows NT and NTFS are not currently compatible with FAT32 boot sectors and volumes.

Master Boot Record

The MBR, the most important data structure on the disk, is created when the disk is partitioned. The MBR contains a small amount of executable code called the master boot code, the disk signature, and the partition table for the disk. At the end of the MBR is a 2-byte structure called a signature word or end of sector marker, which is always set to 0x55AA. A signature word also marks the end of an extended boot record (EBR) and the boot sector.

The disk signature, a unique number at offset 0x01B8, identifies the disk to the operating system. Windows 2000 uses the disk signature as an index to store and retrieve information about the disk in the registry subkey:

HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices

Master Boot Code

The master boot code performs the following activities:

1. Scans the partition table for the active partition.
2. Finds the starting sector of the active partition.
3. Loads a copy of the boot sector from the active partition into memory.
4. Transfers control to the executable code in the boot sector.

If the master boot code cannot complete these functions, the system displays one of the following error messages:

- Invalid partition table
- Error loading operating system
- Missing operating system

Note There is no MBR on a floppy disk. The first sector on a floppy disk is the boot sector. Although every hard disk contains an MBR, the master boot code is used only if the disk contains the active, primary partition.

For more information about troubleshooting MBR problems, see "Damaged MBRs and Boot Sectors" later in this chapter.

Partition Table

The partition table, a 64-byte data structure used to identify the type and location of partitions on a hard disk, conforms to a standard layout independent of the operating system. Each partition table entry is 16 bytes long, with a maximum of four entries. Each entry starts at a predetermined offset from the beginning of the sector, as follows:

- Partition 1 0x01BE (446)
- Partition 2 0x01CE (462)
- Partition 3 0x01DE (478)
- Partition 4 0x01EE (494)

Note Only basic disk makes use of the partition table in Windows 2000. Dynamic disk uses the Disk Management database located at the end of the disk for disk configuration information. The partition table is not updated when volumes are deleted or extended after the dynamic disk upgrade, or when new dynamic volumes are created.

The following example shows a partial printout of an MBR revealing the partition table from a computer with three partitions. When there are fewer than four partitions on a disk, the remaining partition table fields are set to the value 0.

```
000001B0:                                                 80 01                ..
000001C0: 01 00 07 FE BF 09 3F 00  - 00 00 4B F5 7F 00 00 00   ......?...K.□...
000001D0: 81 0A 07 FE FF FF 8A F5  - 7F 00 3D 26 9C 00 00 00   ........□.=&....
000001E0: C1 FF 05 FE FF FF C7 1B  - 1C 01 D6 96 92 00 00 00   ................
000001F0: 00 00 00 00 00 00 00 00  - 00 00 00 00 00 00         ..............
```

Table 1.2 describes the fields in each entry in the partition table. The sample values correspond to the first partition table entry shown in the preceding example. The Byte Offset values correspond to the addresses of the first partition table entry. There are three additional entries whose values can be calculated by adding 10h to the byte offset value specific for each additional partition table entry (for example, add 20h for partition table entry 3 and 30h for partition table entry 4).

Table 1.2 Partition Table Fields

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x01BE	BYTE	0x80	**Boot Indicator**. Indicates whether the volume is the active partition. Legal values include: 00. Do not use for booting. 80. Active partition.
0x01BF	BYTE	0x01	**Starting Head**.
0x01C0	6 bits	0x01 *	**Starting Sector**. Only bits 0-5 are used. The upper two bits, 6 and 7, are used by the Starting Cylinder field.
0x01C1	10 bits	0x00 *	**Starting Cylinder**. Uses 1 byte in addition to the upper 2 bits from the Starting Sector field to make up the cylinder value. The Starting Cylinder is a 10-bit number, with a maximum value of 1023.
0x01C2	BYTE	0x07	**System ID**. Defines the volume type. See Table 1.3 for sample values.
0x01C3	BYTE	0xFE	**Ending Head**.
0x01C4	6 bits	0xBF *	**Ending Sector**. Only bits 0-5 are used. The upper two bits, 6 and 7, are used by the Ending Cylinder field.
0x01C5	10 bits	0x09 *	**Ending Cylinder**. Uses 1 byte in addition to the upper 2 bits from the Ending Sector field to make up the cylinder value. The Ending Cylinder is a 10-bit number, with a maximum value of 1023.
0x01C6	DWORD	0x3F000000	**Relative Sectors**. The offset from the beginning of the disk to the beginning of the volume, counting by sectors.
0x01CA	DWORD	0x4BF57F00	**Total Sectors**. The total number of sectors in the volume.

A BYTE is 8 bits, a WORD is 16 bits, a DWORD is 32 bits, and a LONGLONG is 64 bits. Sample values marked with an asterisk (*) do not accurately represent the value of the fields, because the fields are either 6 bits or 10 bits and the data is recorded in bytes.

Numbers larger than one byte are stored in little endian format, or reverse-byte ordering. Little endian format is a method of storing a number so that the least significant byte appears first in the hexadecimal number notation. For example, the sample value for the **Relative Sectors** field in the previous table, 0x3F000000, is a little endian representation of 0x0000003F. The decimal equivalent of this little endian number is 63.

Boot Indicator Field

The first element of the partition table, the **Boot Indicator** field, indicates whether or not the volume is the active partition. Only one primary partition on the disk can have this field set.

It is possible to have different operating systems and different file systems on different volumes. By using disk configuration tools such as the Windows 2000-based Disk Management or the MS-DOS-based Fdisk to designate a primary partition as active, the **Boot Indicator** field for that partition is set in the partition table.

System ID Field

Another element of the partition table is the **System ID** field. It defines which file system, such as FAT16, FAT32, or NTFS, was used to format the volume and the FT characteristics of the volume. The **System ID** field also identifies an extended partition, if one is defined. Windows 2000 uses the **System ID** field to determine which file system device drivers to load during startup. Table 1.3 identifies the values for the **System ID** field.

Table 1.3 System ID Values

Partition Type	ID Value
0x01	FAT12 primary partition or logical drive (fewer than 32,680 sectors in the volume)
0x04	FAT16 partition or logical drive (32,680–65,535 sectors or 16 MB–33 MB)
0x05	Extended partition
0x06	BIGDOS FAT16 partition or logical drive (33 MB–4 GB)
0x07	Installable File System (NTFS partition or logical drive)
0x0B	FAT32 partition or logical drive
0x0C	FAT32 partition or logical drive using BIOS INT 13h extensions
0x0E	BIGDOS FAT16 partition or logical drive using BIOS INT 13h extensions
0x0F	Extended partition using BIOS INT 13h extensions
0x12	EISA partition
0x42	Dynamic disk volume
0x86	Legacy FT FAT16 disk *
0x87	Legacy FT NTFS disk *
0x8B	Legacy FT volume formatted with FAT32 *
0x8C	Legacy FT volume using BIOS INT 13h extensions formatted with FAT32 *

Partition types denoted with an asterisk (*) indicate that they are also used to designate non-FT configurations such as striped and spanned volumes.

When a mirrored or RAID-5 volume is created in Windows NT 4.0 or earlier, the high bit of the **System ID** field byte is set for each primary partition or logical drive that is a member of the volume. For example, a FAT16 primary partition or logical drive that is a member of a mirrored or RAID-5 volume, has a **System ID** value of 0x86. A FAT32 primary partition or logical drive has a **System ID** value of 0x8B, and an NTFS primary partition or logical drive has a **System ID** value of 0x87. Volumes that have the high bit set can only be directly accessed by Windows 2000 and Windows NT. Disk shares on FT disks, however, are also available to computers running MS-DOS, Windows 95, and Windows 98.

Note MS-DOS can only access volumes that have a **System ID** value of 0x01, 0x04, 0x05, or 0x06. However, you can delete volumes that have the other values listed in Table 1.3 using MS-DOS tools such as Fdisk. If you use a low-level disk editor, such as DiskProbe, you can read and write to any sector, including ones that are in NTFS volumes.

Starting and Ending Cylinder, Head, and Sector Fields

The **Starting** and **Ending Cylinder**, **Head**, and **Sector** fields (collectively known as the **CHS** fields) are additional elements of the partition table. These fields are essential for starting the computer. The master boot code uses these fields to find and load the boot sector of the active partition. The **Starting CHS** fields for non-active partitions point to the boot sectors of the remaining primary partitions and the EBR of the first logical drive in the extended partition as shown in Figure 1.1.

Knowing the starting sector of an extended partition is very important for low-level disk troubleshooting. If your disk fails, you need to work with the partition starting point (among other factors) to retrieve stored data.

Note To have a written record of the starting and ending sectors of the partitions on your hard disk, as well as other useful disk configuration data, use the DiskMap tool. For more information about DiskMap, see the documentation provided on the *Windows 2000 Resource Kit* companion CD.

Figure 1.1 shows the MBR, partition table, and boot sectors on a disk with four partitions. The definitions of the fields in the partition table and the extended partition tables are the same.

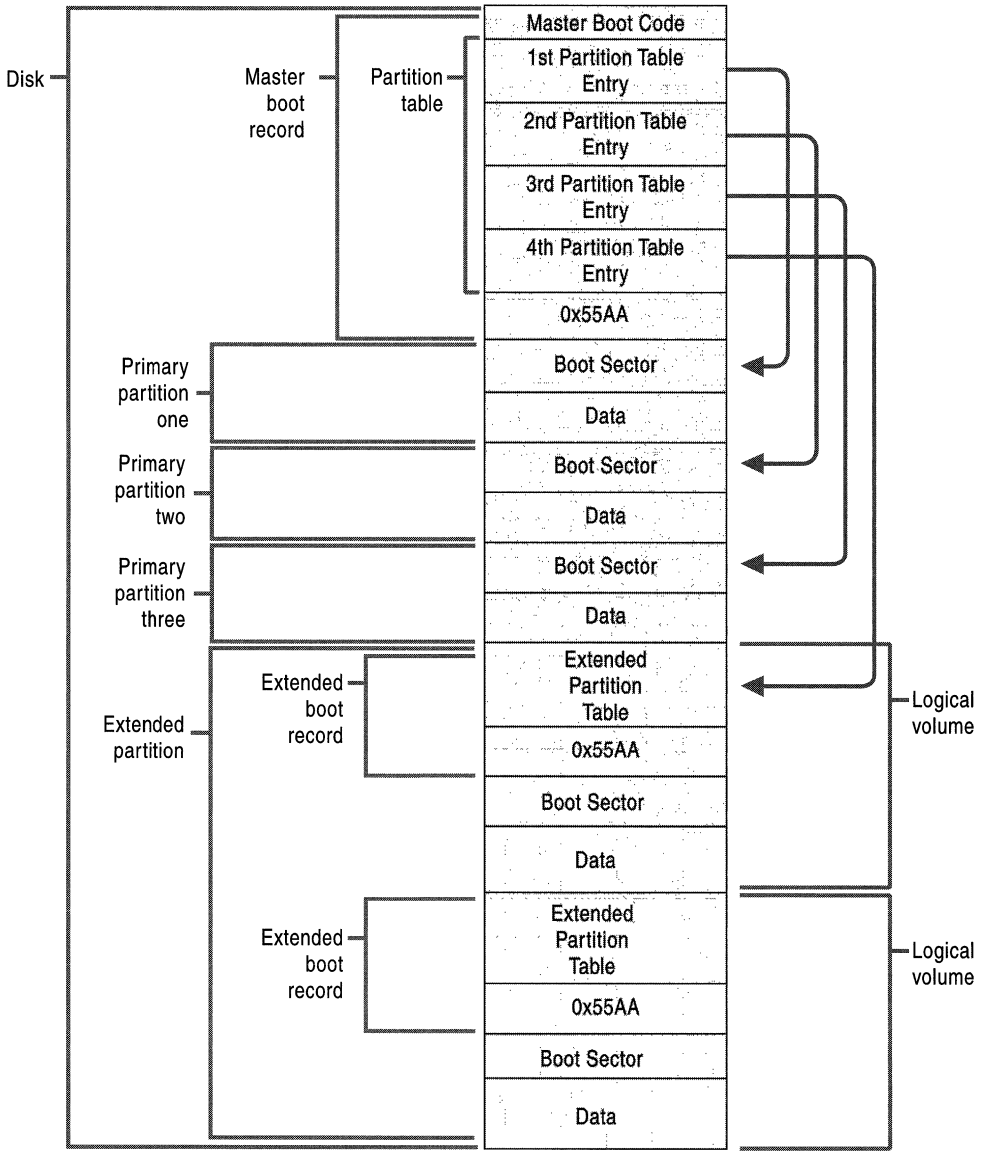


Figure 1.1 Detail of a Basic Disk with Four Partitions

The **Ending Cylinder** field in the partition table is 10 bits long, which limits the number of cylinders that can be described in the partition table to a range of 0–1,023. The **Starting Head** and **Ending Head** fields are each one byte long, which limits the field range to 0–255. The **Starting Sector** and **Ending Sector** fields are each six bits long, which limits the range of these fields to 0–63. However, the enumeration of sectors starts at 1 (not 0, as for other fields), so the maximum number of sectors per track is 63.

Because all hard disks are low-level formatted with a standard 512-byte sector, the maximum disk capacity described by the partition table is calculated as follows:

```
Maximum capacity = sector size x cylinders (10 bits) x heads (8 bits) x
sectors per track (6 bits)
```

Using the maximum possible values yields:

```
512 x 1024 x 256 x 63 (or 512 x 2^24) = 8,455,716,864 bytes or 7.8 GB
```

The calculation results in a maximum capacity of slightly less than 8 gigabytes (GB). Before BIOS INT 13h extensions drive geometry translation (also known as logical block addressing, or LBA) were introduced, the active, primary partition could not exceed 7.8 GB, regardless of the file system used.

Important When using the standard 512-byte sector, the maximum cluster size that you can use for FAT16 volumes while running Windows 2000 is 64 kilobytes (KB). Therefore, the maximum size for a FAT16 volume is 4 GB.

If you use a multiple-boot configuration with Windows 95, Windows 98, or MS-DOS, FAT16 volumes must be limited to 2 GB to be accessed from these operating systems. In addition, a Macintosh computer that accesses volumes on a computer running Windows 2000 cannot access a FAT16 volume that is larger than 2 GB. If you try to use a FAT16 volume larger than 2 GB when running MS-DOS, Windows 95, or Windows 98, or try to access such a volume from a Macintosh computer, you might get a message that zero bytes are available.

The maximum FAT16 volume size that you can use on a computer depends on the disk geometry and the maximum values that fit in the partition table entry fields. Table 1.4 shows the typical FAT16 volume size when LBA is enabled or disabled. The number of cylinders in both cases is 1,024 (0–1,023). When a primary partition or logical drive extends beyond the 1,023rd cylinder, all fields described in this section contain the maximum values.

Table 1.4 FAT16 Volume Size When LBA Is Enabled or Disabled

Translation Mode	Number of Heads	Sectors per Track	Maximum Size for System or Boot Partition
Disabled	64	32	1 GB
Enabled	255	63	4 GB

Warning Do not change the LBA setting on any hard disk containing data. You can adversely affect the process in which the system translates the disk attributes for storing data and corrupt all the files and partitions on the physical disk. Refer to your computer owner's manual before modifying this BIOS setting.

To accommodate sizes larger than 7.8 GB, Windows 2000 ignores the values in the **Starting** and **Ending Sector** fields of the partition table in favor of the **Relative Sectors** and **Total Sectors** fields.

Relative Sectors and Total Sectors Fields

The **Relative Sectors** field represents the offset from the beginning of the disk to the beginning of the volume, counting by sectors, for the volume described by the partition table entry. The **Total Sectors** field represents the total number of sectors in the volume.

Using the **Relative Sectors** and **Total Sectors** fields (resulting in a 32-bit number) provides eight more bits than the CHS scheme to represent the total number of sectors. This allows partitions containing up to 2^{32} sectors to be defined. With a standard sector size of 512 bytes, the 32 bits used to represent the **Relative Sectors** and **Total Sectors** fields translates into a maximum partition size of 2 terabytes (or 2,199,023,255,552 bytes).

This addressing scheme is only used in Windows 2000 with NTFS and FAT32.

Note In addition, the Format tool of Windows 2000 limits the maximum size of FAT32 volumes it can create to 32 GB. However, Windows 2000 can directly access larger FAT32 volumes created by Windows 95 OSR2 or Windows 98.

Windows 2000 uses the fields in the partition table entries to access all partitions. A partition that is formatted while Windows 2000 is running puts data into the **Starting** and **Ending CHS** fields to have compatibility with MS-DOS, Windows 95, and Windows 98, and to maintain compatibility with the BIOS INT 13h for startup.

Extended Boot Record

An EBR, which consists of an extended partition table and the signature word for the sector, exists for each logical drive in the extended partition. It contains the only information on the first side of the first cylinder of each logical drive in the extended partition. The boot sector in a logical drive is usually located at either Relative Sector 32 or 63. However, if there is no extended partition on a disk, there are no EBRs and no logical drives.

Note This information applies only to disks configured with basic disk.

The first entry in an extended partition table for the first logical drive points to its own boot sector. The second entry points to the EBR of the next logical drive. If no further logical drives exist, the second entry is not used and is recorded as a series of zeroes. If there are additional logical drives, the first entry of the extended partition table for the second logical drive points to its own boot sector. The second entry of the extended partition table for the second logical drive points to the EBR of the next logical drive. The third and fourth entries of an extended partition table are never used.

As shown in Figure 1.2, the EBRs of the logical drives in the extended partition are a linked list. The figure shows three logical drives on an extended partition, illustrating the difference in extended partition tables between preceding logical drives and the last logical drive.

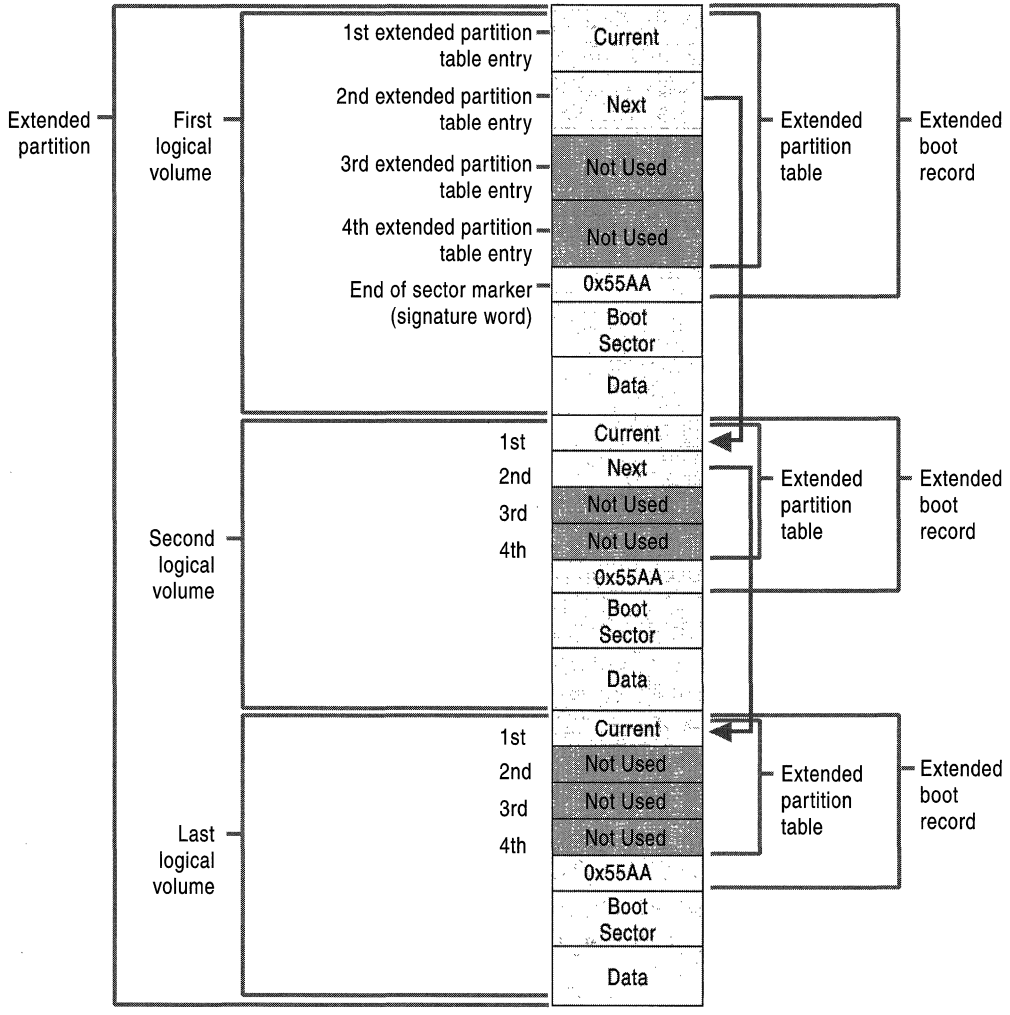


Figure 1.2 Detail of an Extended Partition

With the exception of the last logical drive on the extended partition, the format of the extended partition table, described in Table 1.5, is repeated for each logical drive: the first entry identifies the logical drive's own boot sector and the second entry identifies the next logical drive's EBR. The extended partition table for the last logical drive has only its own partition entry listed. The second through fourth entries of the last extended partition table are not used.

Table 1.5 Contents of Extended Partition Table Entries

Extended Partition Table Entry	Entry Contents
First	Information about the current logical drive in the extended partition, including the starting address for data.
Second	Information about the next logical drive in the extended partition, including the address of the sector that contains the EBR for the next logical drive. If no further logical drives exist, this field is not used.
Third	Not used
Fourth	Not used

The fields in each entry of the extended partition table are identical to the MBR partition table entries. See Table 1.2 for more information about partition table fields.

The **Relative Sectors** field in an extended partition table entry shows the number of bytes that are offset from the beginning of the extended partition to the first sector in the logical drive. The number in the **Total Sectors** field refers to the number of sectors that make up the logical drive. The value of the **Total Sectors** field equals the number of sectors from the boot sector defined by the extended partition table entry to the end of the logical drive.

Because of the importance of the MBR and EBR sectors, it is recommended that you run disk-scanning tools regularly as well as regularly back up all your data files to protect against losing access to a volume or an entire disk.

Boot Sector

The boot sector, located at sector 1 of each volume, is a critical disk structure for starting your computer. It contains executable code and data required by the code, including information that the file system uses to access the volume. The boot sector is created when you format a volume. At the end of the boot sector is a two-byte structure called a signature word or end of sector marker, which is always set to 0x55AA. On computers running Windows 2000, the boot sector on the active partition loads into memory and starts Ntldr, which loads the operating system.

The Windows 2000 boot sector consists of the following elements:

- An *x*86-based CPU jump instruction.
- The original equipment manufacturer identification (OEM ID).
- The BIOS parameter block (BPB), a data structure.
- The extended BPB.
- The executable boot code (or bootstrap code) that starts the operating system.

Note All Windows 2000 boot sectors contain these elements. However, the NTFS boot sector, the FAT16, and the FAT32 boot sectors are all formatted differently.

The BPB describes the physical parameters of the volume: the extended BPB begins immediately after the BPB. Due to differing types of fields and the amount of data they contain, the length of the BPB is different for FAT16, FAT32, and NTFS boot sectors.

The information in the BPB and the extended BPB is used by disk device drivers to read and configure volumes. The area following the extended BPB typically contains executable boot code, which performs the actions necessary to continue the startup process.

Boot Sector Startup Processes

Computers use the boot sector to run instructions during startup. The initial startup process is summarized in the following steps:

1. The system BIOS and the CPU initiate the power-on self test (POST).
2. The BIOS searches for a boot device (typically a disk).
3. The BIOS loads the first physical sector of the boot device into memory and transfers CPU execution to that memory address.

If the boot device is on a hard disk, the BIOS loads the MBR. The master boot code in the MBR loads the boot sector of the active partition, and transfers CPU execution to that memory address. On computers that are running Windows 2000, the executable boot code in the boot sector finds Ntldr, loads it into memory, and transfers execution to that file.

Note Windows 2000 cannot start up from a spanned, striped, or RAID-5 volume that is running dynamic disk. These disk structures cannot be registered into the MBR's partition table, so a system partition using these structures is not startable. Windows 2000 must be fully loaded into memory before they can be used.

If there is a floppy disk in drive A, the system BIOS loads the first sector (the boot sector) of the disk into memory. If the disk is startable—formatted by MS-DOS with core operating system files applied—the boot sector loads into memory and uses the executable boot code to transfer CPU execution to Io.sys, a core MS-DOS operating system file. If the floppy disk is not bootable, the executable boot code displays an error message such as:

```
Non-System disk or disk error
Replace and press any key when ready
```

Note This error will not appear on normally functioning systems that are configured to look for the startup files on drive C first. On many computers, an option in the CMOS setup program allows the user to set the sequence of installed disks that the system searches for the startup files.

If you get similar errors when trying to start the computer from the hard disk, the boot sector might be corrupted. For more information about troubleshooting boot sector problems, see "Damaged MBRs and Boot Sectors" later in this chapter.

Initially, the startup process is independent of disk format and operating system. The unique characteristics of operating and file systems become important when the boot sector's executable boot code starts.

Components of a Boot Sector

The MBR transfers CPU execution to the boot sector, so the first three bytes of the boot sector must be valid, executable *x*86-based CPU instructions. This includes a jump instruction that skips the next several nonexecutable bytes.

Following the jump instruction is the 8-byte OEM ID, a string of characters that identifies the name and version number of the operating system that formatted the volume. To preserve compatibility with MS-DOS, Windows 2000 records "MSDOS5.0" in this field on FAT16 and FAT32 disks. On NTFS disks, Windows 2000 records "NTFS."

Note You may also see the OEM ID "MSWIN4.0" on disks formatted by Windows 95 and "MSWIN4.1" on disks formatted by Windows 95 OSR2 and Windows 98. Windows 2000 does not use the OEM ID field in the boot sector except for verifying NTFS volumes.

Following the OEM ID is the BPB, which provides information that enables the executable boot code to locate Ntldr. The BPB always starts at the same offset, so standard parameters are in a known location. Disk size and geometry variables are encapsulated in the BPB. Because the first part of the boot sector is an x86 jump instruction, the BPB can be extended in the future by appending new information at the end. The jump instruction needs only a minor adjustment to accommodate this change. The BPB is stored in a packed (unaligned) format.

FAT16 Boot Sector

Table 1.6 describes the boot sector of a volume formatted with the FAT16 file system.

Table 1.6 Boot Sector Sections on a FAT16 Volume

Byte Offset	Field Length	Field Name
0x00	3 bytes	Jump Instruction
0x03	LONGLONG	OEM ID
0x0B	25 bytes	BPB
0x24	26 bytes	Extended BPB
0x3E	448 bytes	Bootstrap Code
0x01FE	WORD	End of Sector Marker

The following example illustrates a hexadecimal printout of the boot sector on a FAT16 volume. The printout is formatted in three sections:

- Bytes 0x00–0x0A are the jump instruction and the OEM ID (shown in bold print).
- Bytes 0x0B–0x3D are the BPB and the extended BPB.
- The remaining section is the bootstrap code and the end of sector marker (shown in bold print).

```
Physical Sector: Cyl 0, Side 1, Sector 1
00000000: EB 3C 90 4D 53 44 4F 53  - 35 2E 30 00 02 40 01 00   .<.MSDOS5.0..@..
00000010: 02 00 02 00 00 F8 FC 00  - 3F 00 40 00 3F 00 00 00   ........?.@.?...
00000020: 01 F0 3E 00 80 00 29 A8  - 8B 36 52 4E 4F 20 4E 41   ..>...)..6RNO NA
00000030: 4D 45 20 20 20 20 46 41  - 54 31 36 20 20 20 33 C0   ME    FAT16   3.
00000040: 8E D0 BC 00 7C 68 C0 07  - 1F A0 10 00 F7 26 16 00   ....|h......&..
00000050: 03 06 0E 00 50 91 B8 20  - 00 F7 26 11 00 8B 1E 0B   ....P.. ..&.....
00000060: 00 03 C3 48 F7 F3 03 C8  - 89 0E 08 02 68 00 10 07   ...H........h...
00000070: 33 DB 8F 06 13 02 89 1E  - 15 02 0E E8 90 00 72 57   3.............rW
00000080: 33 DB 8B 0E 11 00 8B FB  - 51 B9 0B 00 BE DC 01 F3   3.......Q.......
00000090: A6 59 74 05 83 C3 20 E2  - ED E3 37 26 8B 57 1A 52   .Yt... ...7&.W.R
000000A0: B8 01 00 68 00 20 07 33  - DB 0E E8 48 00 72 28 5B   ...h. .3...H.r([
000000B0: 8D 36 0B 00 8D 3E 0B 02  - 1E 8F 45 02 C7 05 F5 00   .6...>....E.....
```

```
000000C0: 1E 8F 45 06 C7 45 04 0E  - 01 8A 16 24 00 EA 03 00   ..E..E.....$....
000000D0: 00 20 BE 86 01 EB 03 BE  - A2 01 E8 09 00 BE C1 01   . ..............
000000E0: E8 03 00 FB EB FE AC 0A  - C0 74 09 B4 0E BB 07 00   .........t......
000000F0: CD 10 EB F2 C3 50 4A 4A  - A0 0D 00 32 E4 F7 E2 03   .....PJJ...2....
00000100: 06 08 02 83 D2 00 A3 13  - 02 89 16 15 02 58 A2 07   .............X..
00000110: 02 A1 13 02 8B 16 15 02  - 03 06 1C 00 13 16 1E 00   ................
00000120: F7 36 18 00 FE C2 88 16  - 06 02 33 D2 F7 36 1A 00   .6........3..6..
00000130: 88 16 25 00 A3 04 02 A1  - 18 00 2A 06 06 02 40 3A   ..%.......*...@:
00000140: 06 07 02 76 05 A0 07 02  - 32 E4 50 B4 02 8B 0E 04   ...v....2.P.....
00000150: 02 C0 E5 06 0A 2E 06 02  - 86 E9 8B 16 24 00 CD 13   ............$...
00000160: 0F 83 05 00 83 C4 02 F9  - CB 58 28 06 07 02 76 11   .........X(...v.
00000170: 01 06 13 02 83 16 15 02  - 00 F7 26 0B 00 03 D8 EB   ..........&.....
00000180: 90 A2 07 02 F8 CB 42 4F  - 4F 54 3A 20 43 6F 75 6C   ......BOOT: Coul
00000190: 64 6E 27 74 20 66 69 6E  - 64 20 4E 54 4C 44 52 0D   dn't find NTLDR.
000001A0: 0A 00 42 4F 4F 54 3A 20  - 49 2F 4F 20 65 72 72 6F   ..BOOT: I/O erro
000001B0: 72 20 72 65 61 64 69 6E  - 67 20 64 69 73 6B 0D 0A   r reading disk..
000001C0: 00 50 6C 65 61 73 65 20  - 69 6E 73 65 72 74 20 61   .Please insert a
000001D0: 6E 6F 74 68 65 72 20 64  - 69 73 6B 00 4E 54 4C 44   nother disk.NTLD
000001E0: 52 20 20 20 20 20 20 00  - 00 00 00 00 00 00 00 00   R       ........
000001F0: 00 00 00 00 00 00 00 00  - 00 00 00 00 00 00 55 AA   ..............U.
```

Tables 1.7 and 1.8 illustrate the layout of the BPB and the extended BPB for FAT16 volumes. The sample values correspond to the data in the preceding example.

Table 1.7 BPB Fields for FAT16 Volumes

Byte Offset	Field Length	Value	Field Name and Definition
0x0B	WORD	0x0002	**Bytes Per Sector**. The size of a hardware sector. Valid decimal values for this field are 512, 1024, 2048, and 4096. For most disks used in the United States, the value of this field is 512.
0x0D	BYTE	0x40	**Sectors Per Cluster**. The number of sectors in a cluster. Because FAT16 can track only a limited number of clusters (up to 65,536), large volumes are supported by increasing the number of sectors per cluster. The default cluster size for a volume depends on the volume size. Valid decimal values for this field are 1, 2, 4, 8, 16, 32, 64, and 128. Values that lead to clusters larger than 32 KB (**Bytes Per Sector** * **Sectors Per Cluster**) can cause disk and software errors.
0x0E	WORD	0x0100	**Reserved Sectors**. The number of sectors preceding the start of the first FAT, including the boot sector. The value of this field is always 1.

(continued)

Table 1.7 BPB Fields for FAT16 Volumes *(continued)*

Byte Offset	Field Length	Value	Field Name and Definition
0x10	BYTE	0x02	**Number of FATs**. The number of copies of the FAT on the volume. The value of this field is always 2.
0x11	WORD	0x0002	**Root Entries**. The total number of 32-byte file and folder name entries that can be stored in the root folder of the volume. On a typical hard disk, the value of this field is 512. One entry is always used as a Volume Label, and files and folders with long names use multiple entries per file. The largest number of file and folder entries is typically 511, but entries run out before you reach that number if long file names are used.
0x13	WORD	0x0000	**Small Sectors**. The number of sectors on the volume represented in 16 bits (< 65,536). For volumes larger than 65,536 sectors, this field has a value of zero and the **Large Sectors** field is used instead.
0x15	BYTE	0xF8	**Media Descriptor**. Provides information about the media being used. A value of 0xF8 indicates a hard disk and 0xF0 indicates a high-density 3.5-inch floppy disk. Media descriptor entries are a legacy of MS-DOS FAT16 disks and are not used in Windows 2000.
0x16	WORD	0xFC00	**Sectors Per FAT**. The number of sectors occupied by each FAT on the volume. The computer uses this number and the number of FATs and hidden sectors, to determine where the root directory begins. The computer can also determine where the user data area of the volume begins based on the number of entries in the root directory (512).
0x18	WORD	0x3F00	**Sectors Per Track**. Part of the apparent disk geometry used on a low-level formatted disk.
0x1A	WORD	0x4000	**Number of Heads**. Part of the apparent disk geometry used on a low-level formatted disk.
0x1C	DWORD	0x3F000000	**Hidden Sectors**. The number of sectors on the volume before the boot sector. This value is used during the boot sequence to calculate the absolute offset to the root directory and data areas.
0x20	DWORD	0x01F03E00	**Large Sectors**. If the value of the **Small Sectors** field is zero, this field contains the total number of sectors in the FAT16 volume. If the value of the **Small Sectors** field is not zero, the value of this field is zero.

Table 1.8 Extended BPB Fields for FAT16 Volumes

Byte Offset	Field Length	Value	Field Name and Definition
0x24	BYTE	0x80	**Physical Drive Number**. Related to the BIOS physical drive number. Floppy disk drives are identified as 0x00 and physical hard disks are identified as 0x80, regardless of the number of physical disk drives. Typically, this value is set prior to issuing an INT 13h BIOS call to specify the device to access. The value is only relevant if the device is a boot device.
0x25	BYTE	0x00	**Reserved**. FAT16 volumes are always set to zero.
0x26	BYTE	0x29	**Extended Boot Signature**. A field that must have the value 0x28 or 0x29 to be recognized by Windows 2000.
0x27	DWORD	0xA88B3652	**Volume Serial Number**. A random serial number created when formatting a disk, which helps to distinguish between disks.
0x2B	11 bytes	NO NAME	**Volume Label**. A field once used to store the volume label. The volume label is now stored as a special file in the root directory.
0x36	LONGLONG	FAT16	**File System Type**. A field with a value of either FAT, FAT12 or FAT16, depending on the disk format.

FAT32 Boot Sector

Table 1.9 describes the boot sector of a volume formatted with the FAT32 file system.

Note The FAT32 boot sector is structurally very similar to the FAT16 boot sector, but the FAT32 BPB contains additional fields. The FAT32 extended BPB uses the same fields as FAT16, but the offset addresses of these fields within the boot sector are different than those found in FAT16 boot sectors. Drives formatted in FAT32 are not readable by operating systems that are incompatible with FAT32.

Table 1.9 Boot Sector Sections on a FAT32 Volume

Byte Offset	Field Length	Field Name
0x00	3 bytes	Jump Instruction
0x03	LONGLONG	OEM ID
0x0B	53 bytes	BPB
0x40	26 bytes	Extended BPB
0x5A	420 bytes	Bootstrap Code
0x01FE	WORD	End of Sector Marker

The following example illustrates a hexadecimal printout of the boot sector on a FAT32 volume. The printout is formatted in three sections:

- Bytes 0x00–0x0A are the jump instruction and the OEM ID (shown in bold print).
- Bytes 0x0B–0x59 are the BPB and the extended BPB.
- The remaining section is the bootstrap code and the end of sector marker (shown in bold print).

```
Physical Sector: Cyl 878, Side 0, Sector 1
00000000: EB 58 90 4D 53 44 4F 53  - 35 2E 30 00 02 08 20 00   .X.MSDOS5.0... .
00000010: 02 00 00 00 00 F8 00 00  - 3F 00 FF 00 EE 39 D7 00   ........?....9..
00000020: 7F 32 4E 00 83 13 00 00  - 00 00 00 00 02 00 00 00   □2N.............
00000030: 01 00 06 00 00 00 00 00  - 00 00 00 00 00 00 00 00   ................
00000040: 80 00 29 8B 93 6D 54 4E  - 4F 20 4E 41 4D 45 20 20   ..)..mTNO NAME
00000050: 20 20 46 41 54 33 32 20  - 20 20 33 C9 8E D1 BC F4     FAT32   3.....
00000060: 7B 8E C1 8E D9 BD 00 7C  - 88 4E 02 8A 56 40 B4 08   {......|.N..V@..
00000070: CD 13 73 05 B9 FF FF 8A  - F1 66 0F B6 C6 40 66 0F   ..s......f...@f.
00000080: B6 D1 80 E2 3F F7 E2 86  - CD C0 ED 06 41 66 0F B7   ....?.......Af..
00000090: C9 66 F7 E1 66 89 46 F8  - 83 7E 16 00 75 38 83 7E   .f..f.F..~..u8.~
000000A0: 2A 00 77 32 66 8B 46 1C  - 66 83 C0 0C BB 00 80 B9   *.w2f.F.f.......
000000B0: 01 00 E8 2B 00 E9 48 03  - A0 FA 7D B4 7D 8B F0 AC   ...+..H...}.}...
000000C0: 84 C0 74 17 3C FF 74 09  - B4 0E BB 07 00 CD 10 EB   ..t.<.t.........
000000D0: EE A0 FB 7D EB E5 A0 F9  - 7D EB E0 98 CD 16 CD 19   ...}....}.......
000000E0: 66 60 66 3B 46 F8 0F 82  - 4A 00 66 6A 00 66 50 06   f`f;F...J.fj.fP.
000000F0: 53 66 68 10 00 01 00 80  - 7E 02 00 0F 85 20 00 B4   Sfh.....~.... ..
00000100: 41 BB AA 55 8A 56 40 CD  - 13 0F 82 1C 00 81 FB 55   A..U.V@........U
00000110: AA 0F 85 14 00 F6 C1 01  - 0F 84 0D 00 FE 46 02 B4   .............F..
00000120: 42 8A 56 40 8B F4 CD 13  - B0 F9 66 58 66 58 66 58   B.V@......fXfXfX
00000130: 66 58 EB 2A 66 33 D2 66  - 0F B7 4E 18 66 F7 F1 FE   fX.*f3.f..N.f...
00000140: C2 8A CA 66 8B D0 66 C1  - EA 10 F7 76 1A 86 D6 8A   ...f..f....v....
00000150: 56 40 8A E8 C0 E4 06 0A  - CC B8 01 02 CD 13 66 61   V@............fa
00000160: 0F 82 54 FF 81 C3 00 02  - 66 40 49 0F 85 71 FF C3   ..T.....f@I..q..
00000170: 4E 54 4C 44 52 20 20 20  - 20 20 20 0D 0A 4E 54 4C   NTLDR      ..NTL
00000180: 44 52 20 69 73 20 6D 69  - 73 73 69 6E 67 FF 0D 0A   DR is missing...
00000190: 44 69 73 6B 20 65 72 72  - 6F 72 FF 0D 0A 50 72 65   Disk error...Pre
000001A0: 73 73 20 61 6E 79 20 6B  - 65 79 20 74 6F 20 72 65   ss any key to re
000001B0: 73 74 61 72 74 0D 0A 00  - 00 00 00 00 00 00 00 00   start...........
000001C0: 00 00 00 00 00 00 00 00  - 00 00 00 00 00 00 00 00   ................
000001D0: 00 00 00 00 00 00 00 00  - 00 00 00 00 00 00 00 00   ................
000001E0: 00 00 00 00 00 00 00 00  - 00 00 00 00 00 00 00 00   ................
000001F0: 00 00 00 00 00 00 00 00  - 00 7B 8E 9B 00 00 55 AA   .........{....U.
```

Tables 1.10 and 1.11 illustrate the layout of the BPB and the extended BPB for FAT32 volumes. The sample values correspond to the data in the preceding example.

Table 1.10 BPB Fields for FAT32 Volumes

Byte Offset	Field Length	Value	Field Name and Definition
0x0B	WORD	0x0002	**Bytes Per Sector**. The size of a hardware sector. Valid decimal values for this field are 512, 1024, 2048, and 4096. For most disks used in the United States, the value of this field is 512.
0x0D	BYTE	0x08	**Sectors Per Cluster**. The number of sectors in a cluster. Because FAT32 can only track a finite number of clusters (up to 4,294,967,296), extremely large volumes are supported by increasing the number of sectors per cluster. The default cluster size for a volume depends on the volume size. Valid decimal values for this field are 1, 2, 4, 8, 16, 32, 64, and 128. The Windows 2000 implementation of FAT32 allows for the creation of volumes only up to a maximum of 32 GB. However, larger volumes created by other operating systems (Windows 95 OSR2 and later) are accessible in Windows 2000.
0x0E	WORD	0x0200	**Reserved Sectors**. The number of sectors preceding the start of the first FAT, including the boot sector. The decimal value of this field is typically 32.
0x10	BYTE	0x02	**Number of FATs**. The number of copies of the FAT on the volume. The value of this field is always 2.
0x11	WORD	0x0000	**Root Entries (FAT12/FAT16 only)**. For FAT32 volumes, this field must be set to zero.
0x13	WORD	0x0000	**Small Sectors (FAT12/FAT16 only)**. For FAT32 volumes, this field must be set to zero.
0x15	BYTE	0xF8	**Media Descriptor**. Provides information about the media being used. A value of 0xF8 indicates a hard disk and 0xF0 indicates a high-density 3.5-inch floppy disk. Media descriptor entries are a legacy of MS-DOS FAT16 disks and are not used in Windows 2000.
0x16	WORD	0x0000	**Sectors Per FAT (FAT12/FAT16 only)**. For FAT32 volumes, this field must be set to zero.
0x18	WORD	0x3F00	**Sectors Per Track**. Contains the "sectors per track" geometry value for disks that use INT 13h. The volume is broken down into tracks by multiple heads and cylinders.
0x1A	WORD	0xFF00	**Number of Heads**. Contains the "count of heads" geometry value for disks that use INT 13h. For example, on a 1.44-MB, 3.5-inch floppy disk this value is 2.

(continued)

Table 1.10 BPB Fields for FAT32 Volumes ***(continued)***

Byte Offset	Field Length	Value	Field Name and Definition
0x1C	DWORD	0xEE39D700	**Hidden Sectors**. The number of sectors on the volume before the boot sector. This value is used during the boot sequence to calculate the absolute offset to the root directory and data areas. This field is generally only relevant for media that are visible on interrupt 13h. It must always be zero on media that are not partitioned.
0x20	DWORD	0x7F324E00	**Large Sectors**. Contains the total number of sectors in the FAT32 volume.
0x24	DWORD	0x83130000	**Sectors Per FAT (FAT32 only)**. The number of sectors occupied by each FAT on the volume. The computer uses this number and the number of FATs and hidden sectors (described in this table), to determine where the root directory begins. The computer can also determine where the user data area of the volume begins based on the number of entries in the root directory.
0x28	WORD	0x0000	**Extended Flags (FAT32 only)**. The value of the bits in this two-byte structure are: Bits 0–3: Number of the active FAT (starting count at 0, not 1). It is only valid if mirroring is disabled. Bits 4–6: Reserved. Bit 7: A value of 0 means the FAT is mirrored at run time into all FATs. A value of 1 means only one FAT is active (referenced in bits 0-3). Bits 8–15: Reserved.
0x2A	WORD	0x0000	**File System Version (FAT32 only)**. The high byte is the major revision number, whereas the low byte is the minor revision number. This field supports the ability to extend the FAT32 media type in the future with concern for old FAT32 drivers mounting the volume. If the field is non-zero, back-level Windows versions will not mount the volume.
0x2C	DWORD	0x02000000	**Root Cluster Number (FAT32 only)**. The cluster number of the first cluster of the root directory. This value is typically, but not always, 2.
0x30	WORD	0x0100	**File System Information Sector Number (FAT32 only)**. The sector number of the File System Information (FSINFO) structure in the reserved area of the FAT32 volume. The value is typically 1. A copy of the FSINFO structure is kept in the Backup Boot Sector, but it is not kept up-to-date.
0x34	WORD	0x0600	**Backup Boot Sector (FAT32 only)**. A non-zero value indicates the sector number in the reserved area of the volume in which a copy of the boot sector is stored. The value of this field is typically 6. No other value is recommended.
0x36	12 bytes	0x000000000 00000000000 0000	**Reserved (FAT32 only)**. Reserved space for future expansion. The value of this field should always be zero.

Table 1.11 Extended BPB Fields for FAT32 Volumes

Byte Offset	Field Length	Value	Field Name and Definition
0x40	BYTE	0x80	**Physical Drive Number**. Related to the BIOS physical drive number. Floppy disk drives are identified as 0x00 and physical hard disks are identified as 0x80, regardless of the number of physical disk drives. Typically, this value is set prior to issuing an INT 13h BIOS call to specify the device to access. It is only relevant if the device is a boot device.
0x41	BYTE	0x00	**Reserved**. FAT32 volumes are always set to zero.
0x42	BYTE	0x29	**Extended Boot Signature**. A field that must have the value 0x28 or 0x29 to be recognized by Windows 2000.
0x43	DWORD	0xA88B3652	**Volume Serial Number**. A random serial number created when formatting a disk, which helps to distinguish between disks.
0x47	11 bytes	NO NAME	**Volume Label**. A field once used to store the volume label. The volume label is now stored as a special file in the root directory.
0x52	LONGLONG	FAT32	**System ID**. A text field with a value of FAT32.

NTFS Boot Sector

Table 1.12 describes the boot sector of a volume formatted with NTFS. The bootstrap code for an NTFS volume is longer than the 426 bytes, as shown in Table 1.12. When you format an NTFS volume, the format program allocates the first 16 sectors for the boot sector and the bootstrap code.

Table 1.12 Boot Sector Sections on an NTFS Volume

Byte Offset	Field Length	Field Name
0x00	3 bytes	Jump Instruction
0x03	LONGLONG	OEM ID
0x0B	25 bytes	BPB
0x24	48 bytes	Extended BPB
0x54	426 bytes	Bootstrap Code
0x01FE	WORD	End of Sector Marker

On NTFS volumes, the data fields that follow the BPB form an extended BPB. The data in these fields enables Ntldr to find the master file table (MFT) during startup. On NTFS volumes, the MFT is not located in a predefined sector, as on FAT16 and FAT32 volumes. For this reason, the MFT can be moved if there is a bad sector in its normal location. However, if the data is corrupted, the MFT cannot be located, and Windows 2000 assumes that the volume has not been formatted.

The following example illustrates the boot sector of an NTFS volume formatted while running Windows 2000. The printout is formatted in three sections:

- Bytes 0x00–0x0A are the jump instruction and the OEM ID (shown in bold print).
- Bytes 0x0B–0x53 are the BPB and the extended BPB.
- The remaining code is the bootstrap code and the end of sector marker (shown in bold print).

```
Physical Sector: Cyl 0, Side 1, Sector 1
00000000: EB 52 90 4E 54 46 53 20  - 20 20 20 00 02 08 00 00   .R.NTFS    .....
00000010: 00 00 00 00 00 F8 00 00  - 3F 00 FF 00 3F 00 00 00   ........?...?...
00000020: 00 00 00 00 80 00 80 00  - 4A F5 7F 00 00 00 00 00   ........J.□.....
00000030: 04 00 00 00 00 00 00 00  - 54 FF 07 00 00 00 00 00   ........T.......
00000040: F6 00 00 00 01 00 00 00  - 14 A5 1B 74 C9 1B 74 1C   ...........t..t.
00000050: 00 00 00 00 FA 33 C0 8E  - D0 BC 00 7C FB B8 C0 07   .....3.....|....
00000060: 8E D8 E8 16 00 B8 00 0D  - 8E C0 33 DB C6 06 0E 00   ..........3.....
00000070: 10 E8 53 00 68 00 0D 68  - 6A 02 CB 8A 16 24 00 B4   ..S.h..hj....$..
00000080: 08 CD 13 73 05 B9 FF FF  - 8A F1 66 0F B6 C6 40 66   ...s......f...@f
00000090: 0F B6 D1 80 E2 3F F7 E2  - 86 CD C0 ED 06 41 66 0F   .....?.......Af.
000000A0: B7 C9 66 F7 E1 66 A3 20  - 00 C3 B4 41 BB AA 55 8A   ..f..f. ...A..U.
000000B0: 16 24 00 CD 13 72 0F 81  - FB 55 AA 75 09 F6 C1 01   .$...r...U.u....
000000C0: 74 04 FE 06 14 00 C3 66  - 60 1E 06 66 A1 10 00 66   t......f`..f...f
000000D0: 03 06 1C 00 66 3B 06 20  - 00 0F 82 3A 00 1E 66 6A   ....f;. ...:..fj
000000E0: 00 66 50 06 53 66 68 10  - 00 01 00 80 3E 14 00 00   .fP.Sfh.....>...
000000F0: 0F 85 0C 00 E8 B3 FF 80  - 3E 14 00 00 0F 84 61 00   ........>.....a.
00000100: B4 42 8A 16 24 00 16 1F  - 8B F4 CD 13 66 58 5B 07   .B..$......fX[.
00000110: 66 58 66 58 1F EB 2D 66  - 33 D2 66 0F B7 0E 18 00   fXfX.-f3.f.....
00000120: 66 F7 F1 FE C2 8A CA 66  - 8B D0 66 C1 EA 10 F7 36   f......f..f....6
00000130: 1A 00 86 D6 8A 16 24 00  - 8A E8 C0 E4 06 0A CC B8   ......$.........
00000140: 01 02 CD 13 0F 82 19 00  - 8C C0 05 20 00 8E C0 66   ........... ...f
00000150: FF 06 10 00 FF 0E 0E 00  - 0F 85 6F FF 07 1F 66 61   ..........o..fa
00000160: C3 A0 F8 01 E8 09 00 A0  - FB 01 E8 03 00 FB EB FE   ................
00000170: B4 01 8B F0 AC 3C 00 74  - 09 B4 0E BB 07 00 CD 10   .....<.t........
00000180: EB F2 C3 0D 0A 41 20 64  - 69 73 6B 20 72 65 61 64   .....A disk read
00000190: 20 65 72 72 6F 72 20 6F  - 63 63 75 72 72 65 64 00    error occurred.
000001A0: 0D 0A 4E 54 4C 44 52 20  - 69 73 20 6D 69 73 73 69   ..NTLDR is missi
000001B0: 6E 67 00 0D 0A 4E 54 4C  - 44 52 20 69 73 20 63 6F   ng...NTLDR is co
000001C0: 6D 70 72 65 73 73 65 64  - 00 0D 0A 50 72 65 73 73   mpressed...Press
000001D0: 20 43 74 72 6C 2B 41 6C  - 74 2B 44 65 6C 20 74 6F    Ctrl+Alt+Del to
000001E0: 20 72 65 73 74 61 72 74  - 0D 0A 00 00 00 00 00 00    restart........
000001F0: 00 00 00 00 00 00 00 00  - 83 A0 B3 C9 00 00 55 AA   ..............U.
```

Table 1.13 describes the fields in the BPB and the extended BPB on NTFS volumes. The fields starting at 0x0B, 0x0D, 0x15, 0x18, 0x1A, and 0x1C match those on FAT16 and FAT32 volumes. The sample values correspond to the data in the preceding example.

Table 1.13 BPB and Extended BPB Fields on NTFS Volumes

Byte Offset	Field Length	Sample Value	Field Name
0x0B	WORD	0x0002	Bytes Per Sector
0x0D	BYTE	0x08	Sectors Per Cluster
0x0E	WORD	0x0000	Reserved Sectors
0x10	3 BYTES	0x000000	*always 0*
0x13	WORD	0x0000	*not used by NTFS*
0x15	BYTE	0xF8	Media Descriptor
0x16	WORD	0x0000	*always 0*
0x18	WORD	0x3F00	Sectors Per Track
0x1A	WORD	0xFF00	Number Of Heads
0x1C	DWORD	0x3F000000	Hidden Sectors
0x20	DWORD	0x00000000	*not used by NTFS*
0x24	DWORD	0x80008000	*not used by NTFS*
0x28	LONGLONG	0x4AF57F0000000000	Total Sectors
0x30	LONGLONG	0x0400000000000000	Logical Cluster Number for the file $MFT
0x38	LONGLONG	0x54FF070000000000	Logical Cluster Number for the file $MFTMirr
0x40	DWORD	0xF6000000	Clusters Per File Record Segment
0x44	DWORD	0x01000000	Clusters Per Index Block
0x48	LONGLONG	0x14A51B74C91B741C	Volume Serial Number
0x50	DWORD	0x00000000	Checksum

Protecting the Boot Sector

Because a normally functioning system relies on the boot sector to access a volume, it is highly recommended that you run disk scanning tools such as Chkdsk regularly, as well as back up all of your data files to protect against data loss if you lose access to a volume.

Troubleshooting Disk Problems

There are various causes of disk problems and means of recovering from them. The following are tools that you can use to troubleshoot disk problems:

- DiskProbe can be used to examine and change information on individual disk sectors.
- DiskMap can be used to display the layout of partitions and logical volumes on your disk.

Neither of these tools is designed for use with dynamic disks because they cannot read the dynamic Disk Management database. DiskProbe can change the values of individual bytes in any sector on a dynamic disk, but it cannot navigate the structure of a dynamic disk, so it might be impossible to find the sector you want to view or edit. Therefore it is generally recommended that these tools only be used on basic disks.

DiskProbe is part of the Support Tools collection and can be installed from the Support\Tools folder of the Windows 2000 product CD. For more information about using DiskProbe, see the document Dskprtrb.doc in the folder Program Files\Support Tools.

DiskMap is a tool included on the *Windows 2000 Resource Kit* companion CD and is installed with the rest of the Resource Kit tools. For more information about DiskMap, see the document Diskmap.doc in the folder C:\Program Files\Resource Kit.

Warning Be extremely cautious about making any changes to the structures of your hard disk! DiskProbe does not validate the proposed changes to records. Incorrect values in key data structures can render the hard disk inaccessible or prevent the operating system from starting.

You can easily make changes that have serious consequences, resulting in the following error messages:

```
You cannot start any operating system.
A volume is no longer accessible.
You have to recreate and reformat all of the partitions and logical
volumes.
```

DiskProbe displays a messages asking you to verify any change that you want recorded to disk. Please carefully consider any changes before accepting them.

With careful use of the disk tools, you can solve problems whether they occur through human error, hardware problems, power outages, or other events. It is a good idea to familiarize yourself with these tools in a test situation. Testing is especially important if your configuration has spanned, striped, mirror, or RAID-5 sets.

Note Using DiskProbe, you can save, restore, find, examine, and change the bytes of any sector on the disk, including the MBR and the boot sector. The MBR of disk 0 is used to start Windows 2000–based computers, and the system and boot volumes of disk 0 must be defined in the partition table, making the boot sectors easily located, regardless of the disk configuration used. As a result, DiskProbe can be used to back up and restore these disk structures on computers using dynamic disk.

Viruses

It is always important to take precautions to protect your computer and the data on it from viruses. Many computer viruses exploit the disk structures that your computer uses to start up by replacing, redirecting, or corrupting the code and data that start the operating system.

MBR Viruses

MBR viruses exploit the master boot code that runs automatically when the computer starts up. MBR viruses are activated when the BIOS activates the master boot code, before the operating system is loaded.

Many viruses replace the MBR sector with their own code and move the original MBR to another location on disk. Once the virus is activated, it stays in memory and passes the execution to the original MBR so that startup appears to function normally. Some viruses do not relocate the original MBR, causing all volumes on the disk to become inaccessible. If the active, primary partition's listing in the partition table is destroyed, the computer cannot start. Other viruses relocate the MBR to the last sector of the disk; if that sector is not protected by the virus, it might be overwritten during normal use of the computer, preventing the system from being restarted.

Boot Sector Viruses

As with the master boot code, the boot sector's executable code also runs automatically at startup, creating another vulnerable spot exploited by viruses. Boot sector viruses are activated before the operating system is loaded and run when the master boot code in the MBR identifies the active, primary partition and activates the executable boot code for that volume.

Many viruses update the boot sector with their own code and move the original boot sector to another location on disk. Once the virus is activated, it stays in memory and passes the execution to the original boot sector so that startup appears normal. Some viruses do not relocate the original boot sector, making the volume inaccessible. If the affected volume is the active, primary partition, the system cannot start. Other viruses relocate the boot sector to the last sector of the disk. If that sector is not protected by the virus, it might be overwritten by normal use of the computer, rendering the volume inaccessible or preventing the system from restarting, depending upon which volume was affected.

How MBR and Boot Sector Viruses Affect Windows 2000

A computer can contract an MBR or boot sector virus by one of two common methods: by starting up from an infected floppy disk; or by running an infected program, causing the virus to drop an altered MBR or boot sector onto the hard disk.

The function of an MBR or boot sector virus is typically contained once Windows 2000 has started. If a payload is not run during system startup and the virus preserved the original MBR or boot sector, Windows 2000 prevents the virus from self-replicating to other disks.

Windows 2000 is immune to viruses infecting these disk structures during normal operation, because it only accesses physical disks through protected mode disk drivers. Viruses typically subvert the BIOS INT 13h disk access routines, which are ignored once Windows 2000 has started. However, Windows 2000 computers that are multiple-booted with MS-DOS, Windows 95, or Windows 98 can become infected when Windows 2000 is not running the computer.

If a multiple-boot computer on which Windows 2000 has been installed becomes infected by an MBR or boot sector virus while running another operating system, Windows 2000 is vulnerable to damage.

Once the protected mode disk drivers have been activated, the virus cannot copy itself to other hard disks or floppy disks because the BIOS mechanism on which the virus depends is not used for disk access. However, viruses that have a payload trigger that executes during startup are a threat to computers that are running Windows 2000 because the trigger process is initiated before the control during the computer startup process passes to Windows 2000.

Treating an MBR or Boot Sector Virus Infection

To remove a virus from your computer, use a current, well-known, commercial antivirus program designed for Windows 2000, and update it regularly. In addition to scanning the hard disks in your computer, be sure to scan all floppy disks that have been used in the infected computer, in any other computers, or with other operating systems in an infected multiple-boot computer. Scan them even if you believe they are not infected. Many infections recur because one or more copies of the virus were not detected.

If the computer is already infected with a boot sector virus when Windows 2000 is installed, standard antivirus programs might not completely eliminate the infection because Windows 2000 copies the original MS-DOS boot sector to a file called Bootsect.dos and replaces it with its own boot sector. The Windows 2000 installation is not infected, but if the user chooses to start MS-DOS, Windows 95, or Windows 98, the infected boot sector is reapplied to the system, reinfecting the computer. Antivirus tools that are not specifically designed for Windows 2000 do not know to check Bootsect.dos for viruses.

AVBoot

Microsoft provides a customized antivirus tool that can be used for these types of viruses. AVBoot is located in the \Valueadd\3rdparty\Ca_antiv folder of the Windows 2000 Setup CD. Insert an empty, high-density, 3.5-inch floppy disk, and use Windows 2000 Explorer to locate and double-click Makedisk.bat to create a startup floppy disk that automatically runs AVBoot.

AVBoot scans the memory as well as the MBR and all boot sectors of every locally installed disk. If a virus is found, it offers to remove the virus.

Important Whether you use a third-party antivirus program or AVBoot, be sure to regularly update the virus signature files. Once you install an antivirus program, immediately update the signature files, usually through an Internet connection. Check with the software manufacturer's documentation for specific instructions. AVBoot includes update instructions in the installation folder and on the AVBoot floppy disk.

It is extremely important that you regularly update your antivirus program. In most cases, antivirus programs are unable to reliably detect and clean viruses of which they are unaware. False negative reports can result when using an out-of-date virus scanner. Most commercial antivirus software manufacturers offer monthly updates. Take advantage of the latest download to ensure that your system is protected with the latest virus defenses.

Fdisk /mbr

Do not depend on the MS-DOS command **Fdisk /mbr**, which rewrites the MBR on the hard disk, to resolve MBR infections. Many newer viruses have the properties of both file infector and MBR viruses, and restoring the MBR does not solve the problem if the virus immediately reinfects the system. In addition, running **Fdisk /mbr** on a system infected by MBR viruses that do not preserve or encrypt the original MBR partition table permanently prevents access to the lost partitions. If the disk was configured with a third-party disk management program, running this command eliminates the program overlay control and you cannot start up from the disk.

Important Running **Fdisk /mbr** overwrites only the first 446 bytes of the MBR, the portion known as the master boot code, leaving the existing partition table intact. However, if the signature word, the last two bytes of the MBR, has been deleted, the partition table entries are overwritten with zeroes. If an MBR virus overwrites the signature word, access to all partitions and logical volumes is lost.

Fixmbr command

The Recovery Console, a new troubleshooting tool in Windows 2000, offers a feature called **Fixmbr**. However, it functions identically to the **Fdisk /mbr** command, replacing only the master boot code and not affecting the partition table. For this reason, it is also unlikely to help resolve an infected MBR.

For more information about the Recovery Console, see "Repair, Recovery, and Restore" in this book.

Damaged MBRs and Boot Sectors

When you start a computer from the hard disk, the system BIOS code identifies the startup disk and reads the MBR. The master boot code in the MBR searches for the active, primary partition on the hard disk. If the first hard disk on the system does not contain an active partition, or if the master boot code cannot locate the system partition's boot sector to start the operating system, the MBR displays one of the following error messages:

```
Invalid partition table.
Error loading operating system.
Missing operating system.
```

There might not be an active partition on the hard disk that you want to use to start the computer, or the wrong partition might be identified as the active partition. In this case, use an MS-DOS startup floppy disk to start the computer and use the MS-DOS tool Fdisk to set or change the active partition.

Note Fdisk can only set primary partitions as the active partition. If MBR corruption prevents Fdisk from setting or changing the active partition, you might need to use a third-party, low-level disk editor that can work under MS-DOS to make this change manually. The partition table field that needs to be changed is the **System ID** field. For more information about the fields in the partition table, see "Master Boot Record" earlier in this chapter.

Using an Emergency Repair Disk

If the boot sector cannot find Ntldr, Windows 2000 cannot start. This can be caused by moving, renaming, or deleting Ntldr, corruption of Ntldr, or corruption of the boot sector. Under these circumstances, the computer might not respond to input or might display one of the following error messages:

```
A disk read error occurred.
NTLDR is missing
NTLDR is compressed.
```

If Ntldr is damaged or missing, or if the boot sector is corrupted, you can resolve either problem by starting the Emergency Repair Disk (ERD) and following the prompts for repairing the installation. For more information about the ERD, see Windows 2000 Server Help.

Using the Recovery Console

If the system cannot start due to a corrupted MBR or boot sector, you can recreate either of these disk structures by using the Recovery Console.

To start the Recovery Console, start the computer from the Windows 2000 Setup CD or the Windows 2000 Setup floppy disks. If you do not have Windows 2000 Setup floppy disks and your computer cannot start from the CD, use another Windows 2000–based computer to create the setup disks. For information about creating the Windows 2000 Setup floppy disks, see Windows 2000 Server Help.

Start the computer and enter Windows 2000 Setup. Press ENTER at the **Setup Notification** screen to go to the **Welcome to Setup** screen. Press R to repair a Windows 2000 installation, and then press C to use the Recovery Console.

The Recovery Console displays all valid installations of Windows 2000 on the computer. To access the hard disk, press the number key representing the Windows 2000 installation you want to repair (typically represented as 1: C:\WINNT), and then press ENTER.

Note If you press ENTER without typing a number, the Recovery Console quits and restarts the computer.

The Recovery Console may also show valid installations of Windows NT. However, as the Recovery Console was not specifically designed to work with Windows NT, the results of attempting to access a Windows NT installation can be unpredictable.

The Recovery Console then prompts you for the Administrator password.

Note To access the hard disks with Recovery Console, you must know the password for the local Administrator account. If you do not have the correct password, or if the security database for the installation of Windows 2000 you are attempting to access is corrupted, Recovery Console does not allow access to the local disks.

To replace the MBR, at the Recovery Console command prompt, type:

fixmbr

Verify if you want to proceed because depending upon the location and the cause of the corruption within the damaged MBR, this operation can cause the data on the hard disk to become inaccessible. Press Y to proceed, or N to cancel.

Important Running **Fixmbr** overwrites only the master boot code, leaving the existing partition table intact. If the corruption in the MBR affects the partition table, running **Fixmbr** may not resolve the problem.

To have the Recovery Console replace the boot sector, at the Recovery Console command prompt, type:

fixboot

If you do not specify a particular drive, the Recovery Console replaces the boot sector of the boot partition. If another volume's boot sector is corrupted, enter the **Fixboot** command, followed by a space, and then specify the drive letter with a colon immediately afterward.

You can also use DiskProbe to edit these disk structures. Because DiskProbe only runs under Windows 2000 and Windows NT, you can only use it to fix errors on a boot sector, a non-startup partition, or an MBR on a non-startup disk. For more information about using DiskProbe to edit boot sectors and MBRs, install the Support Tools from the Windows 2000 Setup CD and see the document Dskprtrb.doc.

For more information about the Recovery Console, see "Repair, Recovery, and Restore" in this book.

Other Disk Problems

Disk problems can occur that do not involve the MBR, partition table, extended partition table, or boot sector. Typically, the Windows 2000 disk tools cannot be used to troubleshoot these disk problems.

Stop 0x0000007B — Inaccessible Boot Device

This Stop message, also known as Stop 0x7B, indicates that Windows 2000 lost access to the system partition during the startup process.

This error can be caused by a number of factors, including the failure of the boot device driver to initialize, the installation of an incompatible disk or disk controller, an incompatible device driver, disk cabling problems, disk corruption, viruses, or incompatible logical block addressing (LBA).

The system BIOS allows access to fixed disks that use fewer than 1024 cylinders. Many later disks, however, exceed 1024 cylinders. LBA is used to provide support for these disks. Such support is often built into the system BIOS. However, there are potential problems with LBA, such as:

- If partitions are created and formatted with LBA disabled, and LBA is subsequently enabled, a STOP 0x7B can result. The partitions must be created and formatted while LBA is enabled.
- Some LBA schemes are not compatible with Windows 2000. Check with your vendor.

Warning Changing LBA modes from one scheme to another can force you to recreate and reformat the partitions.

For more information about Stop message 0x7B, see "Windows 2000 Stop Messages" in this book.

Volume Displays as Unknown

If you create and format a volume with NTFS, FAT16, or FAT32, but you cannot access files on it, and Disk Management displays the volume as Unknown, the boot sector for the volume might be corrupted. For NTFS volumes, there are two other possible causes for a volume to display as Unknown:

- Permissions for the volume have been changed.
- The master file table (MFT) is corrupted.

The boot sector can be corrupted by viruses. For more information about cleaning an infected computer, see "Viruses" earlier in this chapter.

Permission problems can occur when you perform the following tasks:

- Create a second volume.
- Remove the group Everyone from the access control list (ACL).
- Grant access to a specific user.

The single user has normal access, but if other users log on, or if Windows 2000 is reinstalled, Disk Management shows the drive as Unknown. To correct this problem, log on as an administrator and take ownership of all folders, or return full control to the group Everyone.

If the MFT file is corrupted, there is no general solution, and you need to contact Microsoft Product Support Services.

CMOS Problems

The CMOS typically stores configuration information about the basic elements of the computer, including RAM, video, and storage devices. If the CMOS is damaged or incapable of retaining its configuration data, the computer might be unable to start.

Each manufacturer and BIOS vendor can decide what a user can configure on the CMOS, and what the standard configuration is. You can access the CMOS by using either a keyboard sequence at startup or a software tool, depending on the manufacturer's specifications. It is recommended that you record or print all CMOS information.

The computer uses the CMOS checksum to determine if any CMOS values have been changed other than by using the CMOS Setup program. If the checksum is not correct, the computer cannot start.

After the CMOS is correctly configured, any CMOS problem is usually caused by one of the following problems:

- A weak battery, which can happen when the computer has been turned off for a long time.
- A loose or faulty connection between the CMOS and the battery.
- A damaged CMOS caused by static electric discharge.

Cables and Connectors

Another source of disk problems can be cabling and connectors. Cables can go bad, but if the cable works initially, it is likely to work for a long time. When new disks are added to the computer, check for cabling problems. New problems might stem from a previously unused connector on an existing cable or from a faulty, longer cable used to connect all the disks that might have replaced the working original. Also check the connections to the disk themselves. If the cables are tightly stretched, one or more connectors may work themselves loose over time, resulting in intermittent problems with the disks.

If your system has small computer system interface (SCSI) adapters, contact the manufacturer for updated Windows 2000 drivers. Try disabling **sync negotiation** in the SCSI BIOS, checking the SCSI identifiers of each device, and confirming proper termination. For enhanced integrated drive electronics (EIDE) devices, define the onboard EIDE port as **Primary only**. Also, check each EIDE device for the proper master, slave, or stand-alone setting. Try removing all EIDE devices except for hard disks.

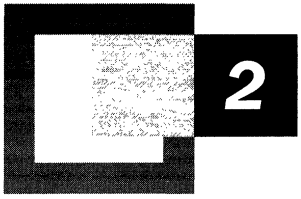
To make sure that any new disks and disk controllers are supported, see the Microsoft Windows 2000 Hardware Compatibility List (HCL) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Additional Resources

- For more information about basic and dynamic disks, see Windows 2000 Server Help.
- For more information about troubleshooting disk problems, see Windows 2000 Server Help.

CHAPTER 2

Data Storage and Management



Data storage and management innovations in Microsoft® Windows® 2000 include Removable Storage and Remote Storage. The overall storage architecture changes provided with Windows 2000 allow users more manageable storage flexibility. They also allow users to take advantage of current storage concepts, including bulk media changers and libraries, data-vaulting managers, and fault-tolerant storage subsystems that are being introduced by a variety of vendors.

In This Chapter

Overview of Data Management 47
Removable Storage 50
Remote Storage 78
Disk Management 101
Disk Quotas 102

Related Information in the Resource Kit

- For more information about NTFS and file systems, see "File Systems" in this book.
- For information about backing up data to Remote Storage, see "Backup" in this book.
- For more information about restoring data from Remote Storage, see "Repair, Recovery, and Restore" in this book.

Overview of Data Management

Significant changes have been made to the storage feature set in Windows 2000. These factors support growing storage requirements in large environments, scalability requirements of mission-critical applications, and support for innovation in the storage market. To address these needs, Windows 2000 features an improved NTFS file system and an extensive list of new storage features and applications.

Current Trends

The quantity of data stored in distributed systems has increased exponentially over the last decade. Migration of mission-critical systems to distributed environments, increases in the number of Internet and intranet applications, and general growth in the enterprise end-user community are all contributing factors. The result is that the number of storage devices in an organization increases in direct proportion to the number of client/server systems.

Recent advances in both the hardware and software required to meet the growing demands of client and server computing include new storage devices, media types, data transfer protocols, and management standards. Storage concepts, such as Remote Storage, bulk media changers and libraries, data vaulting managers, and fault-tolerant storage subsystems are being introduced by a variety of vendors.

Storage requirements continue to increase. As storage becomes more complex, administrators must be able to manage it effectively to accomplish their computing goals, both short term and long term.

Storage Features

Windows 2000 introduces several new storage and storage-related features. These features, available to users and administrators, provide better flexibility, enhanced administrative control, and more efficient usage of resources. Some of the features are enhanced versions of previous Windows applications; others are new. Correct use of these features can help to improve security and reduce management costs associated with storage.

Removable Storage

Removable Storage is a new core Windows 2000 service that manages removable storage media and robotic storage libraries. Removable Storage eliminates the need for independent software vendors (ISVs) to support these devices on a per-device basis. More importantly, Removable Storage enables multiple applications to share expensive removable media storage devices. This allows storage applications to concentrate on customer features rather than hardware issues.

As shown in Figure 2.1, Removable Storage provides a single set of application programming interfaces (APIs) that allows applications to catalog all removable media (except floppy disks and similar small capacity media), such as disk, tape, and optical media, which are either stored on shelves (offline) or in libraries (online). Also, by disguising the complexities of underlying robotic library systems, Removable Storage both lowers costs of developing and operating storage applications and provides consistency to customers who purchase these applications.

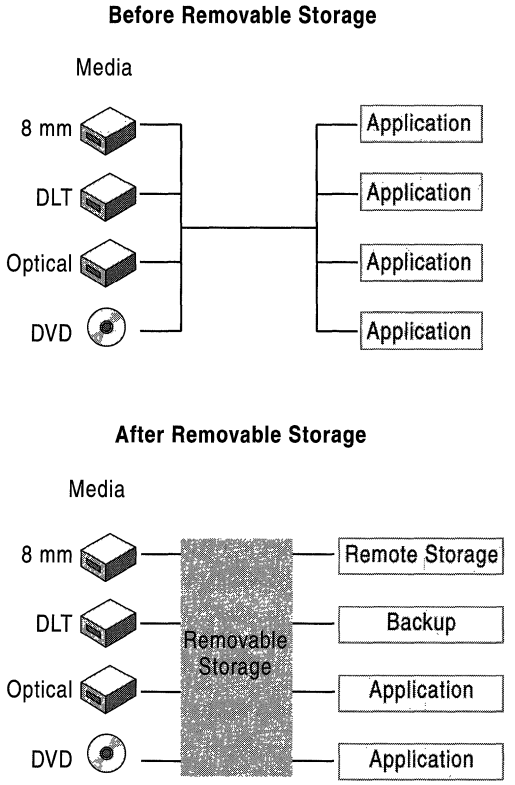


Figure 2.1 Removable Media with and Without Removable Storage

Removable Storage uses media pools to organize media. Media pools have several functions in the management of a media server, such as controlling access to media and grouping media by their use. Media pools allow media to be shared across applications, and they allow such sharing to be tracked.

Remote Storage

Remote Storage is a hierarchical storage management application that migrates data from primary storage to secondary storage. Hierarchical storage management makes sure that data is stored in the most cost-effective method possible. Frequently accessed data is stored on high-performance disks, while data that is not accessed as often is migrated to less expensive media until it is needed again.

Regardless of where the data is stored, the file system namespace continues to provide users with access to the file. When the file is accessed, Remote Storage retrieves the file from its storage location and restores it to the file system. Supported secondary storage in Windows 2000 is limited to tape, such as digital audio tape (DAT) and digital linear tape (DLT).

Remote Storage helps manage the cost associated with large quantities of data that must be accessible. Figure 2.2 illustrates the architecture of Remote Storage.

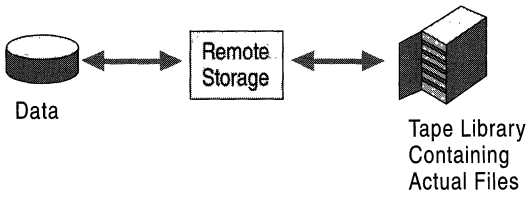


Figure 2.2 Remote Storage

Disk Management

The Disk Management function is enhanced in Windows 2000. Disk Management introduces new disk layouts that provide better manageability and recoverability. Enhancements to Disk Management include a remote administration Microsoft Management Console (MMC) snap-in and an expanded fault tolerance feature set. Moreover, most volume configuration changes do not require restarting the computer.

Disk Quotas

Disk quotas are a new feature in NTFS that provide more granular control of disk storage. Quotas allow administrators to limit the amount of disk space that a user can consume on an NTFS volume.

Disk quotas provide system administrators with a powerful tool for managing storage growth. Administrators can set both soft and hard limits: when a user exceeds a warning level (soft limit), an event is logged; when a user exceeds the hard limit, they receive an Out of Disk error.

Removable Storage

Windows 2000 Removable Storage provides services to applications and system administrators that facilitate the use, sharing, and management of the removable media devices attached to a computer running Microsoft® Windows® 2000 Professional or Windows 2000 Server. Using Removable Storage services allows applications to share local robotic media libraries and tape and disk drives. Removable Storage is not in the media data path, so managing media content is left to Removable Storage client applications, which are referred to as data-mover applications. For example, backup applications use Removable Storage to keep track of the identity of the tapes they use. Removable Storage mounts tapes when needed by the backup application, but the backup application itself keeps track of the backup sets stored on that media.

Removable Storage includes a Windows 2000 service program which runs as part of the service host program (Svchost.exe), the Removable Storage database, and the Removable Storage snap-in.

Removable Storage Service

Using low-level services provided with Windows 2000, Removable Storage acts on requests made by client applications, including:

- Mounting and dismounting media
- Cleaning drives
- Adding and removing media
- Library inventory
- Enabling and disabling libraries, drives, and media
- Accessing media and library attributes

Removable Storage provides these services to client applications by means of an API. This API, and the services provided through it, hide the details of the various drives and libraries. Removable Storage implements a generic changer model which incorporates all essential aspects of libraries and drives. Removable Storage drivers map real drives and changers to this model.

Moving media within a robotic-based library is one example of how Removable Storage is used. The media-moving hardware (usually referred to as a transport) on the hundreds of robotic libraries available today varies widely. Some robotic libraries have hands that grab media and move them from a home storage location (usually referred to as a slot) to a drive. Other types of robotic libraries move a magazine containing some number of media and then push one medium from the magazine into a drive. A mount request from a Removable Storage application mounts media of either description. Figure 2.3 shows the Removable Storage components.

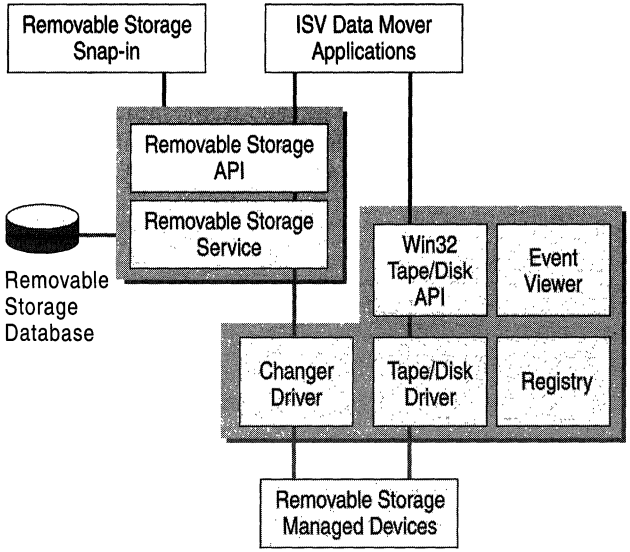


Figure 2.3 Removable Storage Components

Removable Storage Database

The Removable Storage database is the mechanism used by Removable Storage to track media-related system components, including the identity of media known by the system. It stores the status of these system components and maintains media inventories. The Removable Storage database includes the following information:

- The configuration and state of the library, drive, and media
- Media pool configuration and contents
- Library work list
- Operator requests

The Removable Storage database stores the properties of the objects (for example, libraries, drives, and media) that Removable Storage manages. Removable Storage maintains the data in the database and updates it whenever an administrator or an application makes a change in a computer running Removable Storage. For example, if an administrator wants to mount a medium in an online library, Removable Storage mounts the medium and updates the database to reflect the change.

Note The Removable Storage database is used internally by Removable Storage and is not directly available to applications or administrators.

Removable Storage Snap-in

The Removable Storage snap-in is an administrative interface to Removable Storage. It allows administrators to add Removable Storage objects, view and modify properties of Removable Storage objects, insert and eject media, perform inventories, mount and dismount media, and check status information.

Basic Concepts

Removable Storage can be described in terms of five basic concepts: media, physical locations, media pools, work queue items, and operator requests. The last four of these are the top level nodes in the Removable Storage snap-in. The first, media, is the most fundamental concept and affects all the others.

Media

Units of media store information. Each unit of media (or medium, also referred to as a "cartridge") is of a certain type, such as 8mm tape, magnetic disk, optical disk, or CD-ROM.

Most types of media have a single side. For example, a tape must always be oriented in a certain way. When the tape is placed in a drive, all of its data is accessible. Some types of media, such as magneto-optic (MO) disks, have two sides. An MO disk has an "A" side and a "B" side. When an MO disk is placed in a drive with the "A" side up, then the "A" side is accessible and the "B" side is not. To access the "B" side, the disk must be inserted with the "B" side up.

Removable Storage represents media physically and logically. Physical media are the tangible media that are inserted and removed from libraries and mounted in drives.

When an application wants to access data on a medium, Removable Storage generates a logical identifier (ID) that allows the application to request the data on that medium. Since access to the data occurs only through that ID, Removable Storage can manage the physical location of the data. For example, if the original medium begins to fail, Removable Storage can move the data to a new medium without having to notify the application.

Physical media and the sides they contain are tracked in Removable Storage and can take on various states as they are entered into a library and used. For more information about media and side states, see "Media Handling and Usage" later in this chapter.

Removable Storage provides the infrastructure necessary to share media among various applications. Removable Storage ensures that all media are used in such a way as to preserve the data they contain. Removable Storage accomplishes this by identifying and verifying the identity of each medium. For more information about how Removable Storage verifies the identity of media, see "Media Identification and Naming" later in this chapter.

Physical Locations

Removable Storage manages two classes of physical locations: libraries and the offline media physical location. Libraries include both media and the means to read and write them. The offline media physical location is a special holder for media that is cataloged by Removable Storage, but does not reside in a library.

The set of all physical locations includes the libraries and the offline media physical location.

Libraries

In its simplest form, a library is composed of data storage media and a means to read and write the media. A CD-ROM drive with a disc inserted is a simple library with one drive, no slots, and no transport. A more complex example of a library is a robotic-based tape library, which holds several (up to several thousand) tapes, has one or more tape drives, and has a mechanical means to move tapes into and out of the drives.

Specifically, libraries are composed of the following:

Slots Slots are storage locations in the library. For example, a tape library has one slot for each tape that the library can hold. A stand-alone drive library has no slots. However, most libraries have at least four slots. Sometimes slots are organized into collections of slots called magazines. Magazines are usually removable.

Drives A drive is a device that can read or write to a medium. An Iomega Jaz drive, for example, can read and write to a Jaz cartridge which is inserted into it. A library has at least one drive.

Transports A transport is the robotic device that moves a medium from its slot to a drive and back again. Robotic libraries usually have only one transport.

Bar Code Readers A label with a bar code is attached to the outside of a cartridge that is both human readable and computer readable. Libraries that hold media with bar codes attached might have a bar code reader. There is only one reader, which is usually mounted on the transport.

Doors A door is a means to gain unconstrained access to the media in a library. On larger libraries a door might resemble an actual door. When the door is open, an administrator can add and remove media from the library. Doors on smaller libraries might not always look exactly like doors, but they still provide the same functionality. For example, some small changers have all their slots in a magazine. When an administrator instructs Removable Storage to open the door, the changer pushes the magazine out through the front. The administrator can add, replace, or remove media and then reinsert the magazine into the changer. When an administrator adds media to a library, the media are placed directly into a slot. Some libraries have no doors, while others have several.

Insert/Eject Ports While doors provide unconstrained access to the media in a library, an insert/eject (IE) port provides very controlled access. When an administrator adds media to a library through an IE port, the media are not placed directly into a slot. Instead, the media are placed in the IE port and then the library uses the transport to move the media from the IE port to a slot. Some libraries have no IE ports, while others have several. Also called "mailslots," some IE ports handle only one cartridge at a time, while others may handle several.

Types of Libraries A robotic library can have any of the components described above. At a minimum, a robotic library has some media, slots to hold the media, one or more drives, a transport, and either a door or an IE port. Robotic libraries are sometimes referred to as changers or jukeboxes. No human intervention is required to place a medium in a library in one of its drives.

A stand-alone drive library or stand-alone drive is a special kind of library. A medium must be manually placed in a drive. The CD-ROM drive found on most desktop computers is a stand-alone drive library. Removable Storage treats stand-alone drives as libraries with one drive and an IE port. Removable Storage models these drives as libraries to make its program interfaces simpler. For this reason, an application need not know whether a medium is mounted by a transport or a human.

Offline Media Physical Location

The offline media physical location is where Removable Storage lists media that are not in a library. The physical location of media in an online library is the library in which it resides. Media that are not in any of these libraries, such as archived backup tapes on a shelf, are offline media and reside in the offline media physical location. When a user or administrator moves an offline medium into a library, Removable Storage changes its physical location to be the library into which it was placed. When a medium is taken out of an online library, Removable Storage notes that it now resides in the offline media physical location.

Media Pools

A media pool is a logical collection of media that share some common attributes. A media pool contains media of only one type, but media in the media pool can be in more than one library. Each medium is in a media pool. There are two classes of media pools: system and application. System media pools are created by Removable Storage for its own use and include the free, import and unrecognized pools. Application media pools are created by applications to group media. Grouping media is especially important if several applications are sharing the libraries attached to a system and the media they contain.

Each media pool has access permissions that control access to the media that belong to it. While these permissions do not control access to the data contained on the media, they do control the manipulation of the media, including an application's ability to move media from the pool or to allocate it for its own use.

Media pools can be used hierarchically. A media pool can be used to hold other media pools, or it can be used to hold media. An application that needs to group media of several types into one collection can create an application media pool for the whole collection and additional media pools within it, one for each media type. Removable Storage actually uses this technique for its system pools. Within the free pool, for example, is a media pool for each media type. Both sides of a two-sided medium are always in the same pool.

System Pools

The system pools are used to hold media that are not currently being used by an application. The free pool holds unused media that are available to applications, and the unrecognized and import pools are temporary holding places for media that have been newly placed in a library. For more information on how Removable Storage uses these pools, especially the unrecognized and import pools, see "Media Handling and Usage" later in this chapter.

Free Pools The free pools support sharing media among applications. They contain media that are freely available to any application, and they hold no useful data. An application can draw media from the free pools when it needs additional media, and it can return media to the free pools when the media are no longer needed and should be made available to other applications.

Unrecognized Pools When a medium is placed in a library, Removable Storage tries to identify it. If it has not seen this particular media before nor discerned its format or the application which last wrote data on it, Removable Storage places the medium in the unrecognized pool for its media type. Blank media are treated this way. Media in the unrecognized pools might have data on them, but they are not cataloged by Removable Storage nor is Removable Storage able to read any data on these media.

Import Pools If Removable Storage can identify the format or application associated with a medium that has just been placed in a library but has never been seen before, it places the medium in the import pool. For example, if an administrator placed a tape written by Backup on one system into a library attached to a second system, the instance of Removable Storage on the second system recognizes that the tape was written using Microsoft Tape Format (MTF) and places it in the proper media type import pool.

Application Pools

Each application that uses media managed by Removable Storage uses one or more application pools. Applications can create these pools, or they can be created using the Removable Storage snap-in. Permissions on application pools can be set up to allow applications to share pools, or to assign each application its own set of pools.

As mentioned earlier, a pool can either be created to hold other pools or it can be created to hold media. A pool created to hold media has additional properties that specify the actions to be taken when an application allocates or deallocates media.

The following example shows one way in which media pools might be configured to support the needs of several client applications. The example system contains two libraries—one optical disk library and one tape library—and two data mover applications—a backup application and a document management application.

Table 2.1 shows how the two applications use the two media types.

Table 2.1 Media Type and Data Mover Applications

Media Type	Backup Application	Document Management Application
Tape media	Full backup Incremental backup	Engineering archive documents
Optical disk media	Catalog storage	Accounting documents Engineering documents

Work Queue

When applications make a library request , Removable Storage places these requests in a queue and processes them as resources become available. For example, a request to mount a tape in a library results in a mount work queue item, which might wait until a drive is available.

Table 2.2 describes the states that a work queue item can have while it is being handled by Removable Storage.

Table 2.2 Work Queue States

State	Description
Queued	A work item is queued from the time an application issues a request until Removable Storage examines the request.
In Process	A work item is in process while Removable Storage is actively working on completing it.
Waiting	If Removable Storage examines the request and finds that one or more of the resources needed to satisfy the request is busy (for example, the requested drive is being used by another application), the request enters the waiting state.
Completed	When Removable Storage completes the request, the work queue item for that request enters the completed state.
Failed	If Removable Storage is unable to complete the request, the work queue item for that request enters the failed state.

Operator Requests

Note "Operator" as used here, is synonymous with "administrator."

Sometimes, even with robotic libraries, manual assistance is needed to complete a request or perform maintenance or support. If an application requests that a medium in the offline media physical location be mounted, the medium must be manually entered into an online library, generating a request to the operator to enter the cartridge.

Operator requests can be presented to an administrator through the Windows 2000 Messenger Service or the system tray. An administrator can refuse or complete the request. Definitions of the operator request states are shown in Table 2.3.

Table 2.3 Operator Request States

State	Description
Submitted	Removable Storage or an application has submitted a request for administrator action. This operator request is waiting for the operator to complete the requested operation.
Refused	An operator has indicated that the requested action will not be performed.
Completed	An administrator has indicated that the action was completed, or Removable Storage has detected that the action was completed.

Media Identification and Naming

As a manager of shared resources, Removable Storage ensures that client applications do not corrupt each other's data. Placing permissions on media pools prevents an application from modifying the data of another application. Another way to accomplish this is to assign each medium a unique name which applications can use to refer to it, and to track the identities of all media in a system, ensuring that the mapping between the unique name and the physical media is always accurate.

A library management system could simply use a media's home slot as its identity. A request to mount a medium in a drive could be as simple as "Mount the medium in slot 6 of drive 2." There are problems with this minimalist approach, however. An administrator could exchange the medium in slot 6 for that in slot 7 during a door access. An application executing the request might get a medium it did not expect and not know it. To prevent this, Removable Storage assigns each medium a unique name, which is used to refer to the medium programmatically. In keeping with Windows 2000 naming conventions, Removable Storage uses globally unique identifiers (GUIDs) as the form for these names.

Removable Storage identifies each medium side by reading its label. At various times, Removable Storage reads the label to ensure that the name it has for a medium maps correctly. Consider a refinement of the example given earlier. If the name of a medium simply referred to the medium in slot 7, an administrator could change the contents of slot 7 during a door access and the next time an application used the name it would get a different medium. Removable Storage prevents this by associating information stored in the medium's label with its name. Each time a medium is mounted, for example, Removable Storage reads the label to ensure that the medium it mounted is correct. An inventory is an operation that an administrator can request or that Removable Storage can initiate, which verifies the labels associated with each named medium in a library.

Since administrators also need to identify media uniquely, and since GUIDs are difficult to read in printed form, Removable Storage also assigns each medium a display name. These names are changeable so if an administrator has a naming scheme for media, a display name can be assigned that is different from the one Removable Storage initially assigned.

On-media Identifiers

On-media identifiers (OMIDs) are electronically recorded labels on each medium side in a Removable Storage system. Each label is composed of at least two parts: a type and an ID. A label type identifies the format used to record information on the medium. A label ID is an identifier that uniquely identifies each medium.

Different media types have different types of labels. Media sides with file systems (optical disk and Jaz cartridges, for example) have file system labels. An NTFS-formatted Jaz cartridge, for example, has an NTFS label type, and the ID is the volume serial number. Tapes using MTF have a label type of MTF and the ID is derived from various fields in the MTF header.

On-media identifiers are handled somewhat differently for CD-ROM and DVD-ROM media. Serial numbers for instances of these media might not be unique—a library can hold multiple copies of the same CD-ROM title, for example, with each having the same serial number—so Removable Storage allows non-unique label IDs in this case. This doesn't degrade the media identification objectives of Removable Storage; any of the CD-ROMs with identical IDs can be used interchangeably since their contents are identical and cannot be changed.

Duplicate Copies of Media

Although Removable Storage tries to uniquely identify each cartridge that it manages, it also handles media duplicates. It treats identical media as separate media that are identical in every way and can be used interchangeably. For example, if a user has two copies of the Windows 2000 Server Setup CD and places both in a changer, Removable Storage shows both media and each can be mounted, ejected, and so on, separately. Since each has its own record in the database, it is possible to attach different attributes to each media. You can give each a different display name, for example, although this is discouraged because there is no guarantee that these differing attributes will always be associated with the same cartridge. If both are ejected, the attributes may be reversed when both are inserted again.

Bar Codes

Bar code labels on physical media, in libraries with bar code readers, are used as a shortcut to determine the identity of physical media and its sides. Reading an on-media label can be time consuming since it requires reading data on the media, which requires that the medium be placed in a drive if it's not already there. With unique bar codes, a bar code scan is a much faster way of identifying a physical medium and the sides it contains.

Media Names

When new physical media are placed into a library, Removable Storage assigns an initial display name to the media that an administrator can use to visually identify the media. This name is derived from several sources, including the on-media label and the bar code. The rules include the following:

1. If the media has a bar code label, its alphanumeric value is used as the display name.
2. If the media is single-sided and has a recognized label type on its single side, the name is taken from the label. If the format is MTF, the label is taken from information in the MTF header. If the format is a file system, the name is taken from the volume name.
3. The sequence number is used.

Media Change Detection

Many removable media devices, including all those connected to the system by means of SCSI and IDE, cannot themselves report when media are entered or removed. Instead the host system must notice such changes either by querying the device specifically for this information or when performing some other operation on the drive.

Since Removable Storage controls changers and knows when media enter and leave drives, this is not a problem for drives in changers. It is an issue, however, for stand-alone drives. The information that Removable Storage has about the contents of a stand-alone drive might be incorrect if the state of the drive has changed. For example, the snap-in might show that a drive is empty for a period of time after a tape is inserted into a tape drive. An inventory operation on the stand-alone drive library brings the state up-to-date.

Some applications need to have current information. For these applications Removable Storage provides a media change detection mechanism that can be turned on and controlled through Removable Storage API calls. With media detection turned on for a stand-alone drive library, the state of the drive that Removable Storage presents to applications is updated almost immediately after the physical state of the drive changes.

Media Handling and Usage

Most of what Removable Storage does involves both moving media around within and between libraries, and controlling access to that media. While Removable Storage is handling them in these ways, the media in a Removable Storage system have states associated with them. These states determine which operations are permitted and which are prevented. While a medium is available, for example, any application can claim it for its own use, but a medium already in use by one application cannot be claimed by another.

There are two sets of media-related states that govern most of the handling and usage by Removable Storage and its applications: media states and side states.

Media States

The states associated with physical media reflect that the operations performed on them mostly involve movement.

Table 2.4 describes the Removable Storage media states.

Table 2.4 Physical Media States in Removable Storage

Physical Media State	Description
Idle	Medium is in a slot if in a library, or on a shelf if in the offline physical location.
In Use	Removable Storage is in the process of moving the medium.
Mounted	Medium is in a drive, but might not be accessible yet.
Loaded	Medium is in a drive, and the contents of one of its sides are accessible.
Unloaded	Medium is still in a drive, but the contents of the side that was loaded are no longer accessible.

A side is accessible if an application can perform I/O on it. This distinction is most marked in the case of certain tape drives. Inserting a tape into one of these drives does not make its contents accessible. A portion of the tape must be pulled out of the cartridge and wrapped around the tape drive head, a process sometimes referred to as *loading*.

Side States

Since data is stored on a medium side, states associated with sides reflect usage rather than physical location. Table 2.5 lists and describes the Removable Storage media states.

Table 2.5 Side States in Removable Storage

Side State	Description
Allocated	A side that has been claimed for use by an application. The side is not available to any other application.
Available	A side that is available to be claimed and used by any application.
Completed	A side that is in use, but can no longer be used for write operations. The side is physically or logically full.

(continued)

Table 2.5 Side States in Removable Storage *(continued)*

Side State	Description
Decommissioned	A side that can no longer be used because it has crossed a usage threshold. It has reached its allocation maximum (specified by the administrator or an application) and cannot be used again.
Unrecognized	A side whose label types and label IDs are not recognized by Removable Storage.
Imported	A side whose label type is recognized by Removable Storage, but whose label ID is not. It is a new side that Removable Storage has never seen before, but whose format it recognizes.
Inaccessible	A side of a multi-sided cartridge that is in a drive, but not the accessible side.
Incompatible	A side that is not compatible with the pool in which its medium was identified. The medium containing the side needs to be immediately ejected from the library.
Reserved	The second side of a two-sided medium. It is unavailable for allocation to all but the application which already has the first side allocated.
Unprepared	Side that is not claimed or used by any application, but which does not have a free label on it. Applications cannot allocate unprepared media. This is a temporary state. See the following discussion on the interplay between media states and pools.

State Transitions

The state a physical medium or side is in determines what can happen to it next. The transitions for physical media are fairly straightforward. A medium starts off idle and goes to in use when it's being moved to a drive or IE port. Once in a drive, it goes from mounted to loaded. When it is dismounted it goes from unloaded, to in use, and then to idle when it is back in its slot.

More interesting and complex are the state transitions associated with sides.

When a medium is inserted into a library, Removable Storage tries to identify it by reading the labels on its sides. If Removable Storage recognizes an OMID (both the label type and ID) this medium is reentered into a library. Removable Storage notes the change in location, and its sides keep the same state they had when the medium was ejected from the library. If Removable Storage does not recognize the label type, it sets the side's state to unrecognized. If Removable Storage recognizes the label type but not the label ID, Removable Storage sets the side's state to imported.

If a medium is found to not be compatible with the current library, Removable Storage sets its states to incompatible. This might happen if a medium with a form factor (size and shape) identical to that supported by a library, but different in the way information is recorded, is inserted into that library. For example, there are several tape media types that use an 8mm form factor. Since all attempts to read the medium fail, it is incompatible with the library. Media in the incompatible state cannot be changed to another state, and needs to be ejected from the library.

Removable Storage initially sets the state of inserted CD-ROM media to imported.

Sides that are unrecognized can only be changed to available. Generally, unrecognized sides are either on media that have never been used before or were used in a way that is unrelated to any of the Removable Storage client applications and so the contents are not useful. Figure 2.4 describes the legal state transitions for a side.

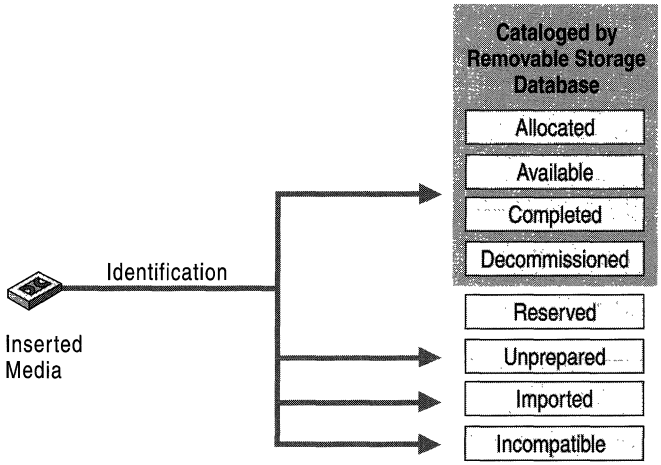


Figure 2.4 Media States

Free Media

Sides that are freely available for any applications to use are in the free pool in the available state. Removable Storage writes a special label (called a free label) on these media so that it can clearly identify these media as holding no useful data. While Removable Storage is in the process of writing a free label, it marks the side as unprepared. This is usually a transitory state, but can persist if, for example, the medium with the side being labeled is in the offline physical media location. Such a side stays in this state until inserted into a library. Any transition into the available state results in writing a free label and therefore passes through the unprepared state.

Application-Specific Media States

When an application needs to claim a side for its use it allocates an available side. This process changes the state of the side from available or imported to allocated. An application allocates a medium that is in the imported state when it recognizes that the new side has data that it needs. When it no longer needs any of the data stored on the side, the application deallocates the side, changing its state back to available.

When a medium is full in some sense—either the side can hold no more data or the application can write no more—an application can mark the medium as complete.

Removable Storage keeps track of the number of times a side is allocated. When this count crosses a threshold, the side is decommissioned. Sides in this state are past their useful life. This count is checked when a side is deallocated.

Since the sides of a two-sided medium are allocated separately, it is possible for an application to try to allocate both sides, or to have the second side claimed by a different application. Having two separate applications allocate the sides of a medium can be problematic, especially in circumstances where media are moved between systems. Removable Storage provides a mechanism that applications can use to eliminate this problem. When an application allocates the first side of a two-sided medium the second side is placed in the reserved state. Only the application that allocated the first side can allocate the reserved second side. If the application determines that it does not need the second side, it can change the state of the second side back to available.

When an application deallocates one side of a medium that contains reserved sides, Removable Storage changes the state of all reserved partitions to available.

Relationship Between Media Pools and States

There are relationships between the various system and application media pools and the states of the sides they contain.

The import pool can hold only media in the import state. Sides in the unrecognized pool can only be in the unrecognized state. In fact, these pools hold sides in these states until applications need them (for example, to move them to application pools). A side in the import state which is moved to an application pool remains in the import state until the application writes a new label on the side and informs Removable Storage.

Sides in the free pools are always in the available state (or the transitional unprepared state). Sides moved into one of the free pools automatically have free labels written on them. Application pools can hold media in any state, except reserved, decommissioned, and unrecognized.

Removable Storage provides a few shortcuts to move sides among pools while changing their state. Normally, an application allocates a side that is in its application pool. However, an application pool property allows Removable Storage to move a side into an application pool when an application tries to allocate a side. If the application tries to allocate a side in a pool in which there are no available sides, Removable Storage moves a side from the appropriate free pool into the application pool and puts it into the allocated state. Conversely, another application pool property indicates that when allocated sides in the pool are deallocated they need to be returned to the appropriate free pool along with the change to the available state.

Administering Removable Storage

Systems with a few stand-alone drives and a single Removable Storage-aware application typically require no Removable Storage administration. Systems with more complex configurations, such as those with tape or optical disk libraries or multiple Removable Storage-aware applications, do require administration, however. There are also rare cases in which a Removable Storage-aware application does require some administration on a system with a simple configuration.

Device Support

There are many drives and robotic libraries available, and not all of them can be supported. See the Microsoft Windows 2000 Hardware Compatibility List (HCL) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Each of the supported robotic libraries has its own configuration method and options. In order for them to work correctly with Removable Storage you must configure your robotic libraries with certain settings of these options. Please consult the HCL for the proper configuration settings for all supported drives and robotic libraries.

Installing and Configuring Removable Storage-Aware Applications

Under normal circumstances, Removable Storage-aware client applications can perform any Removable Storage configuration or setup that they need at the time they are installed. If such an application requires an application media pool, for example, then the pool needs to be created when the application is installed and the permissions need to be set accordingly at that time.

Some applications use their own format or labeling scheme. In order for Removable Storage to correctly process OMIDs on the sides written by these applications, it needs a special dynamic-link library (DLL) that can read the label and determine the OMID. Such a DLL is called a media label library (MLL), and client applications that use them must install them.

Preparing Media

Most applications draw available media from the free pool. Placing media in the free pool is also called "preparing" the media, and must be done either by a Removable Storage-aware application or by an administrator. If your Removable Storage-aware applications do not automatically prepare media, you might have to use the Removable Storage snap-in. You can prepare media in the unrecognized pool, and you can prepare media in the import pool if you are certain that the media contains no useful data. Available media in application pools can also be prepared, but Removable Storage does not prepare allocated media.

To prepare a tape, for example, in the Removable Storage snap-in, right-click the media in the details pane, and then click **Prepare**.

Using Operator Requests in Removable Storage

A Removable Storage operator request is a request for an administrator to perform a task. Operator requests can be issued by Removable Storage or by Removable Storage client applications. Removable Storage generates operator requests in the following situations:

- Media must be moved online because an application has initiated a mount request for a medium that is offline.
- There are no available media online. An application has asked for available media and there are none online in the specified application media pool or the appropriate free media pool. The administrator can supply new media or available media that is offline to satisfy the request.
- A device failed and requires service.
- A drive needs to be cleaned and there is no usable cleaner cartridge available in the library unit.

Operator requests are displayed in the Removable Storage snap-in. Administrators can satisfy or cancel operator requests. After satisfying an operator request, the administrator must acknowledge the request in the Removable Storage snap-in. When an administrator cancels an operator request, Removable Storage notifies the application that generated the request. Removable Storage saves operator requests for less than one hour after they have been satisfied or canceled.

Security

Removable Storage provides security for itself, media pools, and libraries. It contains an access control list (ACL) that controls who can connect to the service and who can work with operator requests. To edit this ACL, right-click the root node in the Removable Storage snap-in. Each library has an ACL that governs who can perform specific tasks, and each pool also contains an ACL. Table 2.6 describes what permissions are required for specific snap-in operations.

Table 2.6 Permissions Required For Removable Storage Snap-in Operations

Function	Media Pool			Library			Service		
	Use	Modify	Control	Use	Modify	Control	Use	Modify	Control
Connect to the service							X		
Create media pools		X							
Delete media pools			X						
Dismount media	X			X					
Mount media	X			X					
Move media from pool to pool		X							
Open the library door						X			
Delete a library					X				
Dismount a drive						X			
Eject media						X			
Insert media						X			
Inventory a library						X			
Cancel an operator request									X
Satisfy an operator request									X
Clean a drive						X			
Insert/eject a cleaner						X			

Controlling the Service

Removable Storage is configured in Windows 2000 to start automatically when you start your computer. It is possible to change the service to be manually started, but this is highly discouraged. Disabling the service breaks several applications that are included with Windows 2000, such as Backup and Remote Storage.

As with other Windows 2000 services, Removable Storage can be stopped, started, and restarted by means of the service control manager.

Service Startup Tasks

Removable Storage performs several tasks when it starts. During some of these tasks Removable Storage is unavailable to administrators and client applications. At other times during startup it can accept requests; however, execution might be delayed while Removable Storage completes this initialization. The following steps outline the process performed by Removable Storage during startup:

Note It is important to note that the process is the same whether Removable Storage is starting after a system start or simply restarted while the system is running.

---

1. Cancel all work items that are still in queue from the last time the service ran.
2. Configure libraries, including associating drives with changers and identifying stand-alone drives.
3. Inventory the contents of each library. Removable Storage performs an inventory in each library according to its set default. At this point Removable Storage determines which slots have media and which media need identification. It also empties the drives, if possible. If the drives are currently in use, Removable Storage leaves the media in the drive and examines the media in the drive at a later time.
4. Start accepting requests from applications.
5. Complete the default inventory work, identifying media as needed.

Device Configuration

Because Removable Storage manages removable media devices, it relies on Plug and Play to tell it which devices are attached to a system, but Removable Storage must match drives with robotic libraries. If Plug and Play indicates that a robotic library is attached to a system, Removable Storage must go through the list of drives that are also attached and detect which ones are actually inside the robotic library and which are stand-alone drive libraries. If certain rules are followed, Removable Storage can do this mapping entirely on its own. If these rules cannot be followed, you must manually map drives to robotic libraries.

Auto-Configuration

Removable Storage auto-configures robotic libraries if the following are true:

- Robotic library hardware units support drive element address reporting with the **ReadElementStatus** SCSI command. Consult the manufacturer to find out if your library hardware unit supports this feature.
- All drives inside a robotic library are on the same SCSI bus as the library.

Manual Configuration

Because not all library hardware units and system configurations support the Removable Storage auto-configuration feature, Removable Storage provides a method for manually configuring library hardware units. However, only use this method when it is necessary, because Removable Storage cannot detect manually configured changes. In general, manually configure Removable Storage only when it detects a robotic library that it cannot configure. After it is set for manual configuration, a changer does not participate in auto-configuration even if its configuration is changed.

In most cases, Removable Storage starts auto-configuration after hardware is installed, moved, or removed. This happens automatically when you restart after adding the device. For changers that cannot be auto-configured, Removable Storage adds incomplete registry entries and generates an operator request to manually configure such changers. The following procedure can be used to manually configure Removable Storage.

▶ To manually configure Removable Storage

You must complete all of the following steps to manually configure Removable Storage:

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

---

1. Stop Removable Storage.
2. Back up the Removable Storage database by copying the files in %SystemRoot%\System32\ntmsdata\ to a secure temporary directory.

3. Restart Removable Storage. Note all the drives that appear as stand-alone drive libraries. Removable Storage displays all drives that are not mapped to a changer as stand-alone drive libraries, including ones that are actually in the changer but are unmapped.
4. Empty all drives on your system.
5. Place a medium in a drive in the library you are trying to configure, either by opening the library door or through a front panel and IE port (see your changer's documentation for details about how to do this). Click **Refresh** in the snap-in for each of the stand-alone drives, and then find the one that now shows that it has media in it. Open the property sheet for that drive and note the device name on the **Device Info** property page. Complete this step for each drive in the changer that you are trying to configure.
6. From the **Start** menu, click **Run**, type **regedt32.exe** or **regedit.exe**, and then click **OK**.
7. In the Removable Storage configuration information in the registry subkey HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NtmsSvc \Config, create a REG_DWORD entry called **AutoCfg** and set the value to **0**.
8. Stop Removable Storage.
9. In the registry editor, navigate to HKEY_LOCAL_MACHINE\System \CurrentControlSet\Services\NtmsSvc\Config. The Config subkey contains a subkey for each changer (such as Changer0) and a subkey for each stand-alone drive. Each changer subkey contains an entry for each drive bay in the changer, such as **DriveBay0.**
10. For each drive bay entry that has the value "???", replace that value with the device name (without any initial "\" or "." characters, such as "Tape3") of the drive in that bay.
11. Close the registry editor.
12. Restart Removable Storage. Removable Storage reads the new configuration information and initializes the devices.
13. Using the snap-in, mount a medium in each drive in the library after Removable Storage is initialized. If any of the configurations are incorrect, Removable Storage generates an error message either during initialization or the mount.
14. If the manual configuration was unsuccessful, stop the Removable Storage process. Copy your backup version of the Removable Storage database files back to the %SystemRoot%\System32\ntmsdata\ directory to restore the database and restart the manual configuration process.

If Removable Storage does not generate any error messages, the manual configuration was successful, and you have completed the manual configuration process.

Database Backup and Recovery

Removable Storage stores its catalog of media in a database on disk. While Removable Storage can recover some of the information by inventorying the attached libraries, all of the application-oriented information might be lost if the database is lost. For this reason, the Removable Storage database must be backed up regularly. For more information about backing up the Removable Storage database, see "Backup" in this book.

Maintaining a Consistent Database

The default location for the Removable Storage database is %SystemRoot%\System32\ntmsdata, which normally contains four files and a folder. Unless you change the location where the database is stored Backup takes the proper steps to create a consistent backup of the database when ntmsdata is selected for backup.

If you decide to move the database or if you do not use Backup, you are responsible for ensuring that you get a consistent database when you do a backup.

Important The backup application that you use is responsible for backing up Removable Storage. If you do not use Backup, which is part of Windows 2000, and you do use Removable Storage, verify that the backup application backs up Removable Storage.

The database must be internally consistent to produce a backup that can be restored successfully. There are two ways of doing this: shutting down Removable Storage and *exporting* the database. The latter is what Backup does if you have not changed the location of the Removable Storage database and you select ntmsdata. If you can shut down Removable Storage while backing up the database, your backup operation can simply make copies of the database files themselves. If shutting down the service is not an option (for example, because a Removable Storage–aware application must remain running), you must provide a way for Removable Storage to export a consistent copy of the database for backup. You can also create a small Removable Storage–aware application that calls the **ExportNtmsDatabase** API. For more information about this API and about writing Removable Storage–aware applications, see the Microsoft Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Tracking the Backup Media

The administrator must always know the location of the most recent Removable Storage database backup. To help identify the backup during disaster recovery, the administrator needs to record and maintain the medium display name and the on-media identifier in a safe place.

You might need this information to help the backup application identify the media from which to restore the database. You can get the display name from the property page for the medium, and the on-media identifier from the property page on the property sheet for the medium side.

Recovering the Database Files

Note The information in this section assumes that there has been no change to the system configuration that affects Removable Storage since the backup was made.

To allow the recovery application to access media, it must be mounted in a drive. If you are going to restore the database using Backup, then Removable Storage must be running and you can use Removable Storage to mount the media, but without the full database you'll have to get Removable Storage running with enough of a database to mount the database backup tape for Backup. To do this, delete any files that might be in ntmsdata and start Removable Storage. Removable Storage then inventories each of the libraries attached to the system.

Tip For large libraries, this might take hours. To minimize the time for this process, perform standard backups often.

Tapes made with Backup show up in the import pool. Start Backup and have it move these tapes to its own application pool. Select the ntmsdata directory and start a restore process. Backup will store the database files and set Removable Storage to run on the recovered database after the next service restart.

If you are not using Backup, you need to use other means to mount the media, restore the files, and recover the database. Most libraries have doors or front panels that you can use to mount media manually. Next, copy or restore the database files to disk. If Removable Storage is not running, you can copy the database files to a directory that holds the Removable Storage database (%SystemRoot%\system32\ntmsdata is the default directory). Then start Removable Storage. If you must have Removable Storage running while you are restoring the database files, copy or restore the database files to the directory \Export, which is a subdirectory of the one that contains the database, and run a utility that calls **ImportNtmsDatabase**. For more information about creating this utility, see the Microsoft Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. For more information about recovering Removable Storage, see "Repair, Recovery, and Restore" in this book.

Updating the Restored Database

The recovered database might not be synchronized with the state of media when the database was lost. Table 2.7 describes some reasons why the Removable Storage database might not be synchronized and how Removable Storage-aware applications can resolve each problem.

Table 2.7 Synchronizing Restored Database Files

Problem	Solution
A medium is deallocated by an application after the backup but before the disaster and remains in the application media pool. Removable Storage recognizes this medium as allocated and puts it in the application media pool.	The application can deallocate this medium again.
A medium is allocated by an application after the backup but before the disaster. Removable Storage puts the medium in the import media pool.	The application can allocate this medium directly out of the import pool.

Note An application can recover the Removable Storage database when it has been lost,there is no backup copy, and the application has not lost the database that refers to the lost Removable Storage database. This process requires significant knowledge on the part of the administrator of the application.

Using the Removable Storage Snap-in

The Removable Storage snap-in allows you to perform several tasks. The snap-in can be found beneath the storage node in the Computer Management snap-in. It can also be started directly from the command line, by running Ntmsmgr.msc.

Inserting and Ejecting Media

Removable Storage uses a library's IE port, if it has one, for inserting and ejecting media, or it uses the door. Selecting an insert task on a library with a door but no IE port results in the same behavior as if you selected a door access.

When using a door access to enter or remove media, make sure that you consult the property page for the library and the document for the changer to determine the proper slot numbers. Most changers label each slot clearly, but some do not.

Caution If during a door access you exchange media in libraries without bar codes, you need to run a full inventory after completing the door access. The change is not detected by a fast inventory and an identity mismatch occurs the next time media in the slot is mounted.

Tables 2.8, 2.9, and 2.10 show the results of moving CD media, tape media, and optical/rewritable media among media pools.

Note Because CD media are read-only, Removable Storage does not allow each medium to be placed in the free pool.

CD media appear in the unrecognized pool only if they are formatted with a file system that Windows 2000 does not recognize.

Table 2.8 Moving CD Media Among Media Pools

From/To	Import Pool	Unrecognized Pool	Application Pool
Import		Not Allowed	OK
Unrecognized	Not Allowed		Not Allowed
Application	OK	Not allowed	

Table 2.9 Moving Tape Media Among Media Pools

From/To	Free Pool	Import Pool	Unrecognized Pool	Application Pool
Free		Not Allowed	Not Allowed	Retains Free Label until application writes new label.
Import	Write Free Label		Not Allowed	Not Allowed
Unrecognized	Write Free Label	Not Allowed		Not Allowed
Application	Write Free Label	Not Allowed	Not Allowed	Retains current Label

Table 2.10 Moving Optical/Rewritable Media Among Media Pools

From/To	Free Pool	Import Pool	Unrecognized Pool	Application Pool
Free		Not Allowed	Not Allowed	Retains Free Label until application writes new label.
Import	Write Free Label		Not Allowed	Retains label already on media.
Unrecognized	Write Free Label	Not Allowed		Not Allowed
Application	Write Free Label	Not Allowed	Not Allowed	Retains current Label. Both sides are moved.

Inventories

There are two types of inventories, fast and full. You can set which is to be used as the default inventory method in the **Library** property page in the Removable Storage snap-in.

A fast inventory only checks for slot state changes between full and empty. If the Removable Storage database indicates that a slot has a cartridge in it, but the fast inventory shows that it no longer has a cartridge, Removable Storage marks the cartridge that was in the slot as offline. If a slot was empty and is now full, Removable Storage identifies the cartridge in the slot. Slots that remain full are assumed to contain the same cartridge. Full inventory actually identifies each medium. This can take a while, unless the media has bar code labels. A full inventory of bar code–labeled media only reads the bar codes. Full inventories of media that are not bar coded read the on-media identifier on each medium in the library.

Cleaning Drives

Some drives, especially tape drives, require periodic cleaning. Usually there is a light on the front of the drive that indicates that a drive is dirty, and in this state most I/O operations fail. Under most circumstances a drive detects that it is dirty while a Removable Storage client application is running. How the client application handles this situation is typically described in the documentation for that application, but for some changers Removable Storage can automatically clean the drive for you after the application is finished using the drive.

In the Removable Storage model of device maintenance, each library unit can contain one cleaner cartridge. There is a wizard available through the snap-in, which you can use to insert a cleaning cartridge into each library that supports automatic cleaning.

Removable Storage maintains a usage count for each cleaner cartridge. When a cleaner reaches its maximum usage count, Removable Storage generates an operator request. If the administrator ejects a cleaner cartridge before it has reached its maximum usage count, Removable Storage displays the usage count information.

Caution Problems occur if Removable Storage attempts to identify a cleaning cartridge as a regular medium. Most drives treat cleaning cartridges differently from regular media, and this different behavior can result in Removable Storage error messages that might appear as if Removable Storage or the library is malfunctioning. Never start Removable Storage for the first time with a cleaning cartridge in the library, or attach a new library with a cleaning cartridge inside. In both these cases Removable Storage performs a full inventory of the library, which includes trying to identify each cartridge. In addition, never insert a cleaning cartridge using the same mechanism used for ordinary media. If there is any doubt about the Removable Storage database's consistency, please remove the cleaning cartridges from the attached libraries. These problems might appear if the database becomes inconsistent or is restored from an old backup.

Work Queue Items

You can use the property page about the work queue item to control how long completed and failed work queue items are retained. If you are having problems you might set the work queue to retain failed items so that you can investigate. The property page for each failed item shows the reason that it failed.

On startup you might see a number of canceled work queue items. When Removable Storage is shut down there might be work queue items still queued. These are canceled the next time Removable Storage starts up.

Operator Requests

Use the property page on the root node of the Removable Storage snap-in to set a method to receive notification of outstanding operator requests. The two available methods are Windows 2000 Messenger Service and a system tray icon. When the messenger service is selected, an application message appears whenever there is an operator request. Make sure that the Messenger Service is running if this option is selected, and then start the Removable Storage snap-in to view the operator request queue and determine what Removable Storage is asking you to do.

If the system tray method is selected, a system tray icon appears whenever there is an outstanding operator request. Clicking on the system tray icon displays a snap-in of operator requests, making it easier to mark them refused or complete. All operator requests must be completed or refused.

The property page of the operator request node in the Removable Storage snap-in lets you set how long to retain completed and failed (including canceled) operator requests. It also provides buttons that can be used to process deleted, completed, and failed operator requests immediately. The **Default** button deletes all requests specified by the controls on the properties sheet, while the **Delete all now** button deletes all completed and failed requests.

Library and Drive States

A library is online while it is operating and connected to the computer that is running Removable Storage. A library is not present if it is not operating or if it is disconnected from the computer. Both robotic and stand-alone libraries can be either online or not present. If a library is removed, the administrator must manually delete the library from Removable Storage. Removable Storage does not automatically delete libraries to avoid deleting information that might still be valuable if the library was disconnected inadvertently or temporarily.

Drives in offline libraries are offline, but media in offline libraries are considered online since they reside in a library. Offline media physical locations, then, are different from offline libraries.

Scripting

You can create command scripts to perform routine or automated activities using the Removable Storage command-line program Rsm.exe. You can use this program to have Removable Storage perform a number of activities, such as ejecting tapes, creating medial pools and so on.

Troubleshooting

Problems with Removable Storage can be caused by hardware or software. The information required to help you resolve either type of problem is contained in the following paragraphs.

Configuration

If you are having problems with Removable Storage correctly configuring your devices, please consult the HCL to make sure your device is supported. If it is, make sure you have configured the device according to the guidelines listed for your device on the HCL. Many changers support many operational configurations, while Removable Storage requires certain settings of these configurations.

Make sure that the hardware is configured correctly. If the device is attached to the host by means of a SCSI bus, ensure that the bus is configured correctly, with no SCSI ID collisions, with proper termination and in accordance with all cable length and SCSI controller parameters, and so on. IDE devices need to be properly configured as master or slave devices and the like.

If all the hardware is configured properly, make sure that Windows 2000 has found the devices and loaded the drivers for them. Make sure you should see your changer under **Media Changers** in the Device Manager, and any drives under **Disk drives** or **Tape drives**. If a driver is loaded, check the system Event Log to see if the driver encountered an error while initializing the device.

If all the devices are working properly, but Removable Storage is still unable to configure them automatically, try manually configuring the devices.

Operation

If you are getting failed work queue items, the property sheet gives a reason for the failure, which can help you diagnose the problem.

If your library is configured correctly but begins malfunctioning, the first place to look for the cause is the system log. Look for Removable Storage messages, but also look for change, drive, and controller error messages. If these devices are experiencing errors, take appropriate actions to clear the errors, such as power cycling or resetting the device.

Operations can begin failing if the system runs low on system resources, such as memory and disk space. You can determine if this is the case by checking the system Event Log.

If the devices and the rest of the system appear to be operating normally, stopping and restarting Removable Storage might clear problems.

If media that are associated with client applications known to Removable Storage are always placed in the unrecognized pool when inserted, or mount requests fail because of an OMID mismatch, the MLL might be missing or installed incorrectly. Information about MLLs is contained in the registry entry HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control \NTMS\OMID\Tape.

Typically, the directory %SystemRoot%\System32 contains all media label library DLLs and is accessible only by administrators.

Remote Storage

Remote Storage runs on computers running Windows 2000 Server. There are two layers to the storage hierarchy. The higher layer, local storage, is made up of the local NTFS volumes of the Windows 2000 file server hosting the Remote Storage software. NTFS volumes are managed at the root of the physical volume. The lower layer, remote storage, stores the data that has been copied from local storage.

Important Remote Storage is available on computers running Windows 2000 Server. It is not part of Windows 2000 Professional.

Remote storage uses a single or multiple tape library. Within a single library, two drives at most are used at a particular time; one for managed data entering remote storage, and one for recalled data leaving remote storage.

Basic Concepts

Remote Storage manages data in local storage, moving less-used data to remote storage, making local storage available for new data. Remote Storage automatically manages the movement of data between local storage and remote storage according to administrator-defined guidelines set for each local storage volume. The administrator-defined guidelines consist of:

- Desired Free Space. The amount of free space specified to be kept on the managed volume.
- File Selection Criteria. Criteria that determine which data is eligible for movement to remote storage.

To prepare for the need to free space on local storage, Remote Storage regularly premigrates (copies) the unnamed data attribute of all eligible NTFS files on local storage to remote storage making these files premigrated files. The last access time of the file does not change when a file is premigrated. When free space on local storage is less than the Desired Free Space, premigrated files are automatically and immediately converted to placeholders until the desired amount of free space is available. A placeholder is an NTFS file which points to the copy of its unnamed data attribute in remote storage and has its unnamed data attribute truncated (removed) from local storage. The placeholder is marked with FILE_ATTRIBUTE_OFFLINE. Even though Remote Storage has changed the physical size of the file on local storage, the logical size and the date/time (create, last modified, last accessed) of the file remains unchanged. Users or applications (for example, disk quotas) that utilize file size are not affected by Remote Storage data migration. Users see all files regardless of storage location, as shown in Figure 2.5.

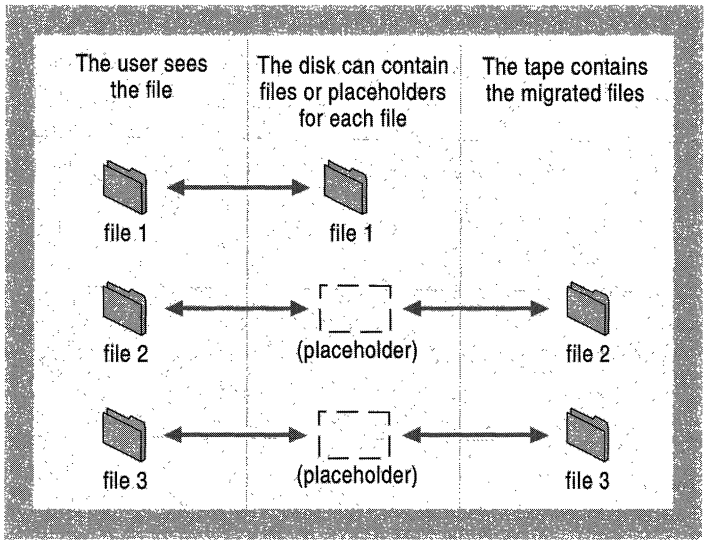


Figure 2.5 File Migration Representation

If a user or application reads, writes, or memory maps a placeholder that was opened using the normal NTFS open request (with no special flags), the unnamed data attribute of the file is automatically recalled from remote storage to local storage, returning this file to the premigrated file status, and the request completes. The latency of the first read, write, or memory map depends on the following criteria:

- The type of remote storage (random access or serial access).
- If remote storage is removable media, whether or not the media is mounted.
- The speed of the device accessing the data (drives or head speed).
- The availability of the devices in remote storage.
- Whether or not there is space to recall the file on local storage.

Notification can be issued to the user when the recall takes a while to complete. If remote storage is inaccessible, the open request fails.

Note FILE_FLAG_OPEN_NO_RECALL is a new flag supported for Windows 2000. If the open request is issued with the new FILE_FLAG_OPEN_NO_RECALL flag, the requested amount of data is read and cached internally and returned to the caller. However, the file remains truncated. The file is read-only, and the data is placed directly into the read buffer, bypassing local storage. The file remains truncated.

If the open request is issued with the FILE_FLAG_BACKUP_SEMANTICS flag, the unnamed data attribute of the file is not recalled. This allows backup programs prior to Windows 2000 to interact with Remote Storage. FILE_FLAG_OPEN_NO_RECALL is added for the legacy backup programs so that the file can be backed up without being recalled.

If you do not want legacy backup applications to back up Remote Storage files, you can set the registry value HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RSFilter\Parameters\SkipFilesForLegacyBackup to 1. Restarting or stopping and starting Remote Storage is required for this change to take effect.

Placeholders and premigrated files can be renamed without affecting the management of the data. When placeholders or premigrated files are copied, the file is recalled and the entire file contents are copied to the new file. The original file remains a premigrated file, while the new file is neither a placeholder nor a premigrated file.

Remote Storage provides disaster prevention and recovery features, including:

- The ability to generate multiple media copies of removable media.
- The ability to replace damaged removable media while Remote Storage is running.
- The ability to recover from loss of Remote Storage metadata.

Remote Storage interfaces are integrated with Windows 2000 Explorer and the Disk Management snap-in. The Remote Storage snap-in allows an administrator to:

- Allocate and configure Remote Storage remote storage devices and media.
- Set Remote Storage systemwide feature options.
- Configure Disk Management settings for Remote Storage–managed volumes.
- View information about Remote Storage activity.
- Recover from media disasters.
- Create and submit jobs.

Remote Storage restricts the number of data paths accessing remote storage. Objects are copied into remote storage using one data path while objects are recalled to local storage using the second data path. Data might be entering and leaving remote storage at the same time, but multiple data paths in the same direction are not allowed.

Note The maximum size of data managed is equal to the amount of media within a storage pool.

The minimum size of managed data is the size which results in space being reclaimed in local storage. This is equal to the amount of space used in the master file table (MFT) record for the file to hold the placeholder data.

Benefits

Remote Storage provides the following benefits to its users:

- A virtual expansion of local storage space using lower cost remote storage.
- Transparent automatic access to data in remote storage.
- Automation of the labor-intensive overhead associated with daily manual data management operations.
- Centralized sharing of remote storage for multiple volumes.

- Integration with the following features of the Windows 2000 operating system:
  - MMC interface
  - Windows 2000 Explorer
  - Windows 2000 Disk Management
  - Event logs
  - Windows 2000 Security
  - NTFS
  - Job Scheduler
  - Microsoft® Systems Management Server
  - Active Directory™ directory service
  - Indexing Services

Local Storage Management

Remote Storage manages NTFS volumes on a computer that is running Windows 2000 Server. Remote Storage does not support clustering.

Remote Storage manages free space on a logical volume. Management begins at the root of the physical volume.

NTFS volumes that reside on removable media can be managed. However, the placeholders on the removable media are not recallable if the volume is plugged into any computer other than the original computer being managed.

Disk Management Attribute

Remote Storage provides a single, system wide, Desired Free Space setting that can be applied to each managed volume. This setting establishes a threshold that triggers an automatic file truncation to free storage on the managed volume when free space is too low. The Desired Free Space setting can be specified by the administrator.

When you initially configure the Desired Free Space for a new managed volume, a reasonable predetermined default value is displayed. Separate from the default value, the system allows an administrator to set a new Desired Free Space for several different managed volumes all at the same time.

The single, system wide, Disk Management attribute is used for truncating premigrated files. The Disk Management attribute is not used to determine how much data is premigrated at a time.

Files are selected for Remote Storage based on the following criteria:

- Path name.
- File names, including wildcard characters in names.
- Inclusion or exclusion of specifications.

The following criteria can be applied to an entire volume but not at an individual file level:

- Last access date or inactivity period
- Owner
- File size
- System and hidden files

By default, the Windows 2000 system files or any files marked with the system attribute are excluded from premigration by Remote Storage.

The administrator uses these criteria to specify the eligible files. The inclusion and exclusion rules are applied such that the more specific a rule is to a file, the higher its priority. For example, the rules of "Exclude *.*" and "Include \PROG*.EXE" causes the file "\Prog\Test.exe" to be included and "\Test.exe" to be excluded.

The implementation of the selection criteria decision engine is extensible using a Component Object Model (COM) interface, so that it can be enhanced in the future and can allow user modifications.

Manage Files

The administrator schedules the management of files based on the specified file selection criteria. All files which meet the criteria are premigrated. The Desired Free Space is not used to determine how many files to premigrate. Premigrated files are tracked to allow rapid file truncation.

Even though Remote Storage changes the physical size of the file on local storage, the logical size and the date/time (create, last modified, last accessed) of the file remains unchanged.

Remove Volume From Management

This action removes the volume from management, and can optionally recall all files from Remote Storage.

Automatic File Truncation

Remote Storage initiates Automatic File Truncation whenever a managed volume's free space level is less than the Desired Free Space. Eligible premigrated files are truncated to free space on local storage. The premigrated files are quickly identified and truncated, based on the last access date. If the premigrated file has not been modified, then the file is truncated and becomes a placeholder. If the premigrated file has been modified, the file is not considered premigrated any longer and is returned to "normal file" status. Additionally, the capacity flow meter is decremented and the data for the premigrated file in remote storage is marked as "deleted" to facilitate space reclamation. Reclamation is the process of deleting from the Remote Storage media pool files that are no longer of interest or are no longer necessary, thereby making that storage space available for reuse.

The automatic file truncation continues until the free space on the volume is greater than the Desired Free Space.

Scheduled File Truncation

Scheduled File Truncation can be scheduled by the administrator to force the truncation of premigrated files to placeholders. The premigrated files that meet the selection criteria are truncated regardless of the free space level of the volume. Premigrated files are checked to make sure that they have not been modified since they were premigrated. If the premigrated file has been modified, then the file is not considered premigrated and the file is returned to "normal file" status. Additionally, the data for the premigrated file in remote storage is marked as "deleted" to facilitate space reclamation.

This action allows proactive space management by an administrator. The action can be scheduled to reduce volume usage in advance of a known volume-intensive event, such as the installation of a very large application.

Validate Managed Files

Validate Managed Files can be scheduled by the administrator to validate placeholders and premigrated files that meet the selection criteria. Validation includes detecting placeholders moved among volumes, disconnected placeholders and modified premigrated files. Corrective actions are taken immediately to synchronize the placeholders and premigrated files with remote storage.

Validate Managed Files is used after restoring files to a volume or after disk errors on a volume. It can be scheduled to run on a regular basis (once a month) to validate the local storage and correct any inconsistencies.

Note Recalls are logged in the Windows 2000 Event Log.

Local Storage

The unnamed data attribute of an NTFS file has the information being tracked by NTFS that is needed to manage the data. The modification time, logical size, physical size, and access time of the file are all kept relative to the unnamed data attribute. The modification time and logical file size are never changed for the file even when it is a placeholder.

First Read, Write, and Memory Map

The file is recalled if necessary on the first read, write, or memory map request. If the data is in remote storage but is not in the library, the recall does not succeed. Automatic recall of data that is not in the library is not available and fails with the message STATUS_FILE_IS_OFFLINE. The administrator needs to manually insert the media into the library and then try the recall again.

If a request is made against a premigrated file, it is not affected by Remote Storage. The data for the file exists locally on the NTFS volume and is accessed normally.

The client making the request on the server might time out. Client computers have their own configured time-out which is independent of Remote Storage, and if recalling the data takes longer than the time-out, the input/output request fails on the client. In Windows 2000, the time-out for accessing off-line Remote Storage files is 15 minutes.

Note Changing attributes or ACLs or named streams in a file does not recall the file.

NTFS "File Open"

By default, when a placeholder is opened on an NTFS volume, the data is copied directly from remote storage to local storage and the file is considered a premigrated file. The last access time of the file is changed. Specific error cases arise because of the state of the system or the location of the data in remote storage. When the data cannot be brought back to local storage, the open request *always* fails with the message STATUS_FILE_IS_OFFLINE.

The FILE_FLAG_OPEN_NO_RECALL flag is honored if the file is opened for read access only, or if the open fails.

If the open request is made with the FILE_FLAG_OPEN_NO_RECALL flag the data is recalled at read time, cached, and placed directly into the application's read buffer. This allows backup systems to get an image of the file that appears (and restores) as a premigrated file but that was not recalled by the backup application.

Note The migrated data appears only briefly and is not actually copied to local disk storage.

Runaway Recall Detection and Prevention

Runaway recalls can be prevented by specifying that no more than a certain number of files can be recalled by a user from a volume in an hour. This is an administrator-defined setting, both in terms of enabling or disabling the feature, and in specifying the number of recall requests that are handled in a particular period of time.

Administrators can select exemption from any recall limit.

Deletion of a Premigrated File or Placeholder

No action is taken upon the deletion of a placeholder or premigrated file. It is not possible to synchronously determine when a file is going to be deleted.

Renaming, Moving, or Copying Placeholders and Premigrated Files

Files cannot be renamed across volume boundaries. Placeholders and premigrated files can be renamed only on the same volume. Renaming a file does not cause the data in remote storage to be recalled. Truncation and recall of these files are not affected.

Copying or moving placeholders between volumes causes the data to be recalled and the entire file, including all migrated data, is copied. At the end of a move operation the original placeholder file is deleted. Copying a file has the same behavior within a volume or across volumes. Moving placeholders on the same volume is a rename operation.

A placeholder can be moved to another volume within the same system by using Backup to back up and restore the placeholder to another volume. In this case, the moved placeholder points to remote storage and is valid to initiate a recall. A user-defined action to validate placeholders corrects any potential inconsistencies between the placeholder and remote storage. Even without the validation of a placeholder, opening the moved placeholder recalls the correct data; however, a volume decommission does not work properly.

Table 2.11 summarizes the results of the specific actions when the destination of the action is the same or a different volume.

Table 2.11 Results of Premigrated File Actions

Action	Destination Same Volume	Destination Different Volume
Copy File	Recall	Recall
Cut File	Recall and delete original	Recall and delete original
Move File	No Recall (is a Rename)	Recall and delete original
Rename File	No Recall	N/A

Volume Decommission

When a managed volume is decommissioned, the physical volume is not accessible. All information needed to decommission the volume is already known. Any data associated with the volume that is in remote storage is treated as available during space reclamation.

If placeholders are moved between volumes, the placeholders must be validated so that correct volume information is known about the placeholders. If the placeholders have not been validated, the volume decommissioning can cause incorrect data within remote storage to be treated as available during space reclamation. For more information about validating volumes, see "Local Storage Management" earlier in this chapter.

Restoring Placeholders from Backup and Disconnected Placeholders

A primary storage backup application, such as Backup, protects placeholders. If the placeholder is deleted, lost, or destroyed the only recourse is to retrieve a copy from the backup media. This is the same mechanism that must be used if any other file on the volume is deleted, lost, or destroyed.

The data in remote storage is accessible only through a placeholder. If the placeholder is deleted, that data is inaccessible. If a placeholder is restored from backup, the placeholder is reconnected to that data.

Disconnected placeholders are placeholders whose file contents have been removed from remote storage. These placeholders might have been restored from backup after the space in remote storage was reclaimed or the data within remote storage might be physically unavailable because of media failures. Disconnected placeholders do not point to valid locations in remote storage and are cleaned up by validating placeholders. If the placeholders are disconnected, they are removed from the system to synchronize the remaining placeholders with remote storage.

Interaction with other Reparse Point Types

Volume mount points and symbolic links cannot be changed into placeholders. The system explicitly ignores these types of files.

Any file system scan that encounters a mount point or symbolic link does not follow the link: physical volume space is the scarce resource and mount points and symbolic links are virtual concepts that extend the directory view.

Placeholders Binding to a Specific Remote Storage

Placeholders identify the specific remote storage system which contains the managed data, tying a placeholder to a specific Remote Storage Engine. Placeholders cannot be moved between multiple remote storage systems.

Windows 2000 Security Integration

Remote Storage supports NTFS security features. Remote Storage only recalls files if the user has valid access to a placeholder. A placeholder is identified as a special remote storage reparse point. The security of the placeholder is determined by the security of the reparse point mechanism.

Administrators and regular users go through normal Windows 2000 user security checks. The Remote Storage Administrative user interface uses Windows 2000 permissions to grant or deny access to the user interface. Only user accounts in the Administrators group are allowed access to the user interface functions.

File Collocation

Files are grouped together by their volume and by the time that they were originally premigrated. The files remain grouped during all remote storage operations. No other grouping of files is supported.

Replication

Files that are being replicated can be excluded from the Manage Files action by the administrator using the file selection criteria.

Remote Storage Engine

Remote storage is the part of the system that holds the copies of the data. The Remote Storage Engine consists of the media used to store the data, libraries, and drives.

The Remote Storage Engine service runs on computers that are running Windows 2000 Server. Only one Remote Storage Engine can exist on a server. The Remote Storage Engine does not run on Windows 2000 Professional.

Flow Meter Capacity Metric

Remote Storage maintains a flow meter capacity metric monitoring the amount of data that is contained in remote storage, including overflowed and shelved data. Copying data from local storage to remote storage increases the value of the capacity metric. Removing local storage references to data in remote storage decreases the value of the capacity metric. The change in the capacity metric is immediate, but some operations are not immediately detected.

Remote Storage increments the value of the flow meter capacity metric whenever one of the following occurs:

- A new file is managed.
- A previously deleted placeholder is restored to a volume and reconnected to remote storage.

Remote Storage decrements the value of the flow meter capacity metric whenever one of the following occurs:

- A premigrated file is modified.
- A placeholder or premigrated file is deleted.
- A file is unmanaged.

There might be a latency associated with the decrementing of the flow meter capacity metric since the detection of these conditions can happen at a later time.

Storage Pool

All remote storage in Remote Storage exists in a single storage pool: there is no storage pool management required.

Relocation is the movement of data from one remote storage pool to another. Since Remote Storage only supports a single remote storage pool, relocation is not supported in Remote Storage.

Remote Storage supports one storage library within the single tape storage pool. The library must have at least one drive, and two or more drives to support media copies. When multiple drives exist within a library, all drives must be identical in type (for example, density or firmware).

Handling of Physically Full Remote Storage

Remote Storage provides the following options for handling physically full remote storage situations:

- Stop managing new files. Remote Storage becomes a recall-only system; this mode is invoked only after any possible space reclamation is performed.
- Shelf Storage. Remote Storage cycles removable media to "shelf"; recalls of files stored on shelved media require administrator interaction.

Remote storage is considered full either when there is not enough scratch media or when the scratch media are not online. In either case, the administrator must provide manual intervention.

A notice is issued when physical remote storage utilization reaches the administrator-defined warning level.

Shelf Media

Shelf media for Remote Storage are media that are not currently accessible by the system. The medium with the data on it can be removed from the device and the media management subsystem tracks its location. Shelf media allow a system to have more data in remote storage than the device can physically contain.

For removable media, the administrator can remove media and optionally replace them with blank media. Remote Storage continues to premigrate data using the new media, and the removed media are considered shelf media. If a user needs access to data on shelf media, the administrator must place the shelf media in the library, and the new media become shelf media. Media do not need to be full to be shelf media.

Media Status and Location

When a medium is marked as having errors, the administrator is notified of the reason for the status change; the information is also kept in the Windows 2000 Event Log for later reference.

Media within a library are not tied to specific locations or slots. The system adjusts to media being in different locations, so media can be shuffled in a library.

Upgrading Remote Storage Libraries

Removable Storage provides Remote Storage with the ability to replace a library with another library containing identical media and data read/write mechanisms. The administrator must physically move the media from the old library to the new library, that is, physically replace the library, and then use the user interface to register configuration changes within Remote Storage.

Remote Storage Device Support

Remote Storage supports all 4mm, 8mm, and DLT tape devices supported by Removable Storage. Because Remote Storage supports only a single media type, a device cannot be a mixed media device.

Kernel-Level Drivers

Remote Storage utilizes Windows 2000 kernel-level drivers to support the management of tape devices. The Microsoft Certified device drivers provide the backbone of device support for Remote Storage.

Sharing Libraries with External Applications

Remote Storage cooperatively shares storage libraries with external applications using a common media management facility such as Removable Storage. A simple media management facility exists within Remote Storage that is usable by other applications.

Minimally, Remote Storage supports nonconcurrent library sharing with external applications. Specific device slots can be designated as in use by an external application.

User Interfaces

The Remote Storage snap-in presents a hierarchical view of the Remote Storage system. Each node in the hierarchy represents a different component of the Remote Storage system. This allows the system administrator to configure the components of the Remote Storage by invoking their corresponding configuration screens or property sheets.

The top level component of the Remote Storage hierarchy is the Windows 2000 Server hosting remote storage. All other components in the Remote Storage snap-in are descendants of the Windows 2000 Server hosting remote storage.

The Remote Storage user interface (UI) is comprised of four parts: the Remote Storage Administrative user interface; the Recall Notification user interface; Windows Explorer integration; and Disk Management integration. The Remote Storage snap-in runs on Windows 2000 Server and Windows 2000 Professional. The nonadministrative UIs (Recall Notification, Windows Explorer integration, and Disk Management integration) run on the same platforms.

Remote Storage Snap-in

The Remote Storage snap-in provides the following components:

- Snap-in registration and initialization
- Namespace node enumeration
- Command architecture action enumeration
- Toolbar and toolbar button enumeration
- Menu and menu item enumeration
- Drag-and-drop registration and negotiation
- Persistent storage handling

The characteristics of the Remote Storage snap-in include the following:

- A two-paned representation of the system in which the console tree contains a hierarchical view of the objects in an administered system, and the right view contains information, properties, and images describing the objects.
- Easy access to frequently used features; features used less often don't clutter the interface.
- Direct manipulation of objects (drag-and-drop, inserting, deleting with single keystrokes, or mouse movements).
- Use of visual metaphors: similar actions have similar interfaces.
- Minimal intervention by the administrator when the Remote Storage is behaving correctly. However, when a problem arises, the Remote Storage snap-in provides very clear and unambiguous notification of the problem.
- All administrative functionality housed in the user interface. No separate, stand-alone administrative utilities exist, although distributed end-user user interface tools exist outside the Remote Storage snap-in.
- Integration of system configuration and monitoring in a single tool.

Recall Notification

On Windows 2000 recall notification can monitor currently active client initiated recalls. The name of the file is shown to the user when a recall is in progress.

This application allows the administrator to cancel the recall. This has the effect of freeing the application (the read, write, or memory map request fails) but, if data transfer has started, the file is still recalled.

Windows Explorer Integration

Placeholders in Windows Explorer are visually different from normal (untruncated) files, but premigrated files appear the same as normal files. Remote Storage integration with Windows Explorer requires adding a new property page to both the file and directory property sheets (accessible through Windows Explorer) to represent storage management properties. The new storage management pages provide information on migration status, premigrated file data location in remote storage, premigrated date and time, and possibly other relevant information. Users have read-only access to the new Remote Storage property pages, though they can force immediate premigration of individual files or entire directories by setting a **Premigrate Now** option on the Remote Storage page.

Specific integration areas include (in order of priority):

1. Windows Explorer does not open the unnamed data stream of placeholder files unless the user needs access to the file data. For less essential tasks, such as showing the icon in the file list, Windows Explorer must use a default icon or an alternative storage location.
2. Windows Explorer must preserve the last access date when a file is opened for behind-the-scenes use (for example, preserving the last access date when searching files for content indexing or when finding a string to satisfy a user query).
3. Content indexing or search operations needs to skip placeholders unless the user has been made aware of the consequences (such as the extended wait to fetch the data from remote storage) and wants to proceed. If the search includes placeholders, use FILE_OPEN_NO_RECALL on the open request. This open mode allows the file to be read without recalling it to primary storage.
4. When a list of files is displayed, the application needs to indicate which files are placeholders.
5. When showing file properties, those that are contained in the unnamed data stream are not be shown by default if the file is a placeholder. When the file is a placeholder, the user is alerted to this fact and made to take extra steps to retrieve this information.
6. Defer the read, write, or memory map of a placeholder until the last moment to avoid unnecessary recall in case the user aborts the operation. For example, do not open the file when Copy or Cut is selected; wait until the Paste operation is performed.

Remote Storage introduces a new property page into the property sheet for managed volumes, which shows statistics and graphics indicating the amount of free space, used space, space taken up by placeholders, and a representation of how much virtual disk space the placeholders represent.

Disk Management Integration

Windows 2000 Disk Management allows the user to bring up a property sheet for each volume on a computer. This property sheet is the same property sheet accessible through My Computer. The Remote Storage property sheet shows total used space, free space, premigrated file space, truncated files (placeholders), untruncated file disk usage, premigration space savings, truncated file compression ratio, percent of files that are placeholders, and other volume report information. The property sheet used by Windows 2000 Disk Management (and by My Computer) is Remote Storage–aware, but the Remote Storage snap-in is not required to use Disk Management.

Miscellaneous Windows 2000 Shell Integration

In addition to integration with Windows Explorer and the Windows 2000 Disk Management snap-in, the Remote Storage snap-in also integrates with the Windows 2000 shell in the following ways:

- Major Remote Storage system events are logged with and reported from the Windows 2000 Event Log.
- Controls, dialogs, and property pages are consistent with Microsoft® Windows® 95 and Microsoft® Windows® 98.

Local Storage Configuration and Management

Remote Storage supports the configuration and management of volumes for local storage. Volumes can be managed individually or in groups.

Disk Management (All Volumes Being Managed)

Disk Management, a subfolder of Remote Storage Computer in the snap-in, deals with the administration and configuration of all managed volumes.

Single Managed Volume

Individual managed volumes are controlled at the Disk Management level.

Directories and Files on a Single Managed Volume

Inside each managed volume are directories and files. Placeholders are shown differently from normal, untruncated files. If you are viewing the file in Windows Explorer, it appears with an overlay icon. From the command prompt, the **Dir** command displays the file size in parentheses.

Single Unmanaged Volume

The administrator has configuration control over volumes that have Remote Storage installed, but which are not currently managed by Remote Storage. Although a volume might not be currently managed, it can still have placeholders on it that can be recalled.

Local Storage Management Monitoring

Remote Storage supports the monitoring of local storage system activity by means of the Remote Storage snap-in. This monitoring occurs at the Disk Management level, the single managed volume level, for directories and files on a single managed volume, and for single unmanaged volumes.

Remote Storage Configuration and Management

Remote storage administration involves managing and configuring the storage devices and media that make up remote storage. The ability to configure and manage remote storage requires administrator privileges.

Remote Storage Management Monitoring

Remote Storage supports monitoring remote storage system activity by means of the Remote Storage snap-in. This monitoring occurs at the overall remote storage level and the volume level or libraries level.

Job Management and Monitoring

The Remote Storage snap-in provides wizards to create and define jobs.

The Remote Storage snap-in provides monitoring of jobs currently running in the system. This provides visual feedback of current system activity. All currently running jobs can be viewed in one monitor screen.

Setup, Installation, and License Management

System installation and setup are integrated with Microsoft® BackOffice® setup and license management. Remote Storage provides Systems Management Server extensions to improve distributed administration. The installation procedure provides an acceptable default configuration of the Remote Storage system that can be run in batch or interactive mode. The installation addresses the following areas:

- Provides instructions and warnings on how the installation proceeds.
- Explains license agreement.
- Gathers user information (such as name and company).
- Allows installer to select components to install.
- Allows installer to specify location of Remote Storage server, UI administration tool, and other tools.
- Selects premigrate volumes.
- Selects media copying policy.

- Selects program folder.
- Copies files and installing, setting up the registry, and performing other tasks.
- Sets up licensing defaults.
- Reboots the system, if desired.

Remote Storage and Windows 2000

Remote Storage is integrated closely with Windows 2000. Table 2.12 provides details of this integration and describes the interfaces between Remote Storage and Windows 2000.

Table 2.12 Remote Storage and Windows 2000 Integration

Interface	Description
Placeholder format and reparse points	A placeholder has the system-defined $REPARSE_POINT attribute set with information that can identify and retrieve the unnamed data attribute from remote storage. The unnamed data attribute is in remote storage and its length is zero bytes. The reparse point type is Remote Storage. The recall mechanism is based on the reparse point being identified by NTFS and is an efficient mechanism that uses this information when recalling a file.
File size of a placeholder	The logical file size of the unnamed data attribute of a placeholder is the size of the unnamed data attribute as it was before it was truncated. The physical size of the unnamed data attribute of a placeholder is zero bytes.
Disk quotas and remote storage	Disk quotas do not change. Disk quotas are based on the logical file size which is not changed by Remote Storage, therefore the amount of space in use on a volume is not reduced or changed by having data in remote storage.
Remote Storage stage change	Remote Storage supports a Distributed Component Object Model (DCOM) interface–based mechanism that can change the system status.
Event Log usage	Major Remote Storage system events are logged in the Windows 2000 Event Log. The Windows 2000 Event Viewer is used to see the logged events. The following events are optionally logged in the Windows 2000 Event Log: ▪ Files being recalled ▪ Files being managed ▪ Files being truncated ▪ Media being mounted ▪ Jobs being run ▪ Files being scanned

(continued)

Table 2.12 Remote Storage and Windows 2000 Integration *(continued)*

Interface	Description
Windows 2000 Job Scheduler	Remote Storage uses the Windows 2000 Job Scheduler to schedule Remote Storage jobs. Job status can be monitored from within the Remote Storage snap-in. Depending on the Job Scheduler, a job window can be specified that limits the amount of time spent in a single scan. This is useful because the system might have a large amount of data to manage and the scan might initially take too much time. A bookmark is kept where the scan stopped so the scan can continue from that point the next time it is run.
Windows 2000 registry usage	Remote Storage uses the registry to keep persistent information about startup, including the programs needed to initialize setup.
Windows 2000 client time-out on recall	The registry entry HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Services\LanmanWorkstation\parameters \OfflineFileTimeOutIntervalInSeconds controls the time-out period that the client uses when sending requests to a server. The default is value is 900 (or 15 minutes). You can increase this value on all Windows 2000 clients that open files on the volumes managed by Remote Storage.

Multiple Remote Storage Interactions

A Remote Storage installation is limited to a single system running Windows 2000 Server in terms of hosting, Disk Management, and Remote Storage. There is no limit, however, to the number of stand-alone Remote Storage instances that can run on a homogeneous Windows 2000 network.

Moving Data Between Instances of Remote Storage to Another Remote Storage

To move managed data from one Remote Storage installation to another, the administrator recalls managed data and moves the data to another volume managed by Remote Storage that manages the data.

Recalling Across Remote Storage Instances

Remote Storage does not support recalling placeholders from a volume which has been managed at different times by two different Remote Storage instances.

Version Compatibility

Remote Storage does not provide an upgrade path for those running third-party remote storage products. Remote Storage provides a subset of features provided in previous remote storage products and is therefore not sufficient for upgrading the system. For example, an existing system with more than one device cannot upgrade to Remote Storage.

Remote Storage–Aware Products

Because Remote Storage manages data transparently for applications and users, there is no absolute requirement that other applications running on NTFS be Remote Storage–aware, that they contain specific code to recognize that Remote Storage is running. However, products that regularly open many files might cause a great deal of data to be recalled and reduce the benefits provided by Remote Storage. For best results, use Remote Storage-aware applications.

Important Although the features included with Windows 2000 are Remote Storage–aware, many other applications are not and may result in time-outs or runaway recalls. In all cases, it is best to check with your ISV to see if they provide a version that is Remote Storage-aware.

Backup Programs

A Remote Storage–aware backup product must have the following abilities:

- A guarantee that a full backup of the file is kept. The backup product must understand the difference between backing up the placeholder and the full data of the file, and must keep the full backup copy.
- Backup a placeholder (reparse point) correctly.
- Use of the FILE_FLAG_OPEN_BACKUP_SEMANTICS parameter when opening a file to back it up.
- Understanding and proper use of FILE_FLAG_OPEN_NO_RECALL.

Backup applications that open placeholders to read and copy without using the FILE_FLAG_OPEN_BACKUP_SEMANTICS parameter cause the data for the file to be recalled. If these products are performing a full system backup, all placeholders are replaced with recalled data.

Remote Storage relies on the primary backup to protect placeholder data. The administrator must be aware of when placeholder backups occur. Specifically, if the backup rotations are used, the administrator must know when the primary backup databases contain a complete copy of the data and when they contain placeholder data.

Backup applications that are Backup-enabled protect placeholder data and do not cause recalls.

Antivirus Programs

Antivirus programs typically open files to check for infection by viruses. These products cause migrated data to be recalled. Remote Storage–aware antivirus programs can operate in one of the following manners:

- Detect placeholders and skip the check on such files. This assumes that the data was checked by the scanner before being premigrated. This might be risky since Remote Storage manages the unnamed data stream and does not manage named data streams. Since data in remote storage remains unchanged, the check for a virus is unnecessary.
- Open the file with the FILE_OPEN_NO_RECALL parameter. This allows the scanner to check the unnamed data stream using local storage without recalling the data. This assumes a linear search of the file data, which might not always be the case.
- Don't update the last access date/time.

Document Management Programs

Many document management products open the data that they are managing, scan for key data, and then store references in additional files or alternate data streams leaving the original data unchanged. A document management product that is Remote Storage–aware opens placeholders with the FILE_OPEN_NO_RECALL parameter allowing the migrated unnamed data to be read but not recalled.

In addition, once the index files are generated, some document management products provide a file viewer much like Windows Explorer. As a Remote Storage-unaware product opens each file, it experiences synchronous recalls of each file and any associated delay. The Remote Storage-aware product indicates which files are migrated, and allows the user to issue asynchronous recalls of the data, while exhibiting improved performance.

Policies for managing files of the document management system exclude index files and other special files so they are quickly accessible to the document management system.

Note The Indexing Service that is included with Windows 2000 is Remote Storage-aware.

Remote Storage Protection and Recovery

Media in remote storage are copied to protect data. Copies are updated as the original media are updated to ensure that a current backup of data is available. For more information about recovering data from Remote Storage, see "Repair, Recovery, and Restore" in this book.

Remote Storage Media Copies

Remote storage media (tape cartridges) are protected by media copies. Media copies are replicas that can be substituted for the original media. Data is migrated only to the original media, but can be recalled from an original or a copy. Copies are made before the original media are completely filled and are updated as the original is updated. When a media copy is completed, it can be removed from the library located with other media in the copy set, and a new media copy is started. Media copies are only created if two or more drives are available to Remote Storage in the library. Other media must be protected by outside sources. Replacing an original with a copy requires administrator intervention. Media copies can be uniquely identified.

When a media copy is used to replace damaged or lost remote storage media, Remote Storage automatically replicates the media again to ensure copy set completeness.

Partial media copies are made by the system to protect the data. These partial copies can be removed from the library and are updated when reinserted in the library. If the source of the media copy is on the shelf, the source media must be reinserted in the library for the completion of the media copy.

Remote Storage Media Copy Sets and Set Rotation

As many as three copy sets of remote storage media are supported. Media for the additional sets can be moved to shelf storage when they are in synch. The administrator can view the status of media copies.

Media for Media Copies

For tape libraries having two or more drives available to Remote Storage, scratch media are used for making a media copy. An administrator can remove the updated media copies on a certain day after several new media copies have been made. This is especially useful for a weekend, during vacations, or initial startup. Once a copy is started, it must be returned to the library to be updated.

Remote Storage Metadata Protection

Remote Storage metadata exists in files which can be protected by the Windows 2000 primary backup system. Only the account used by Remote Storage has access to the metadata, by using Windows 2000 file security. Any operations requiring multiple metadata updates are transactional.

Metadata protection is provided by the Microsoft® Jet database.

Disk Management

Disk Management in Windows 2000 is responsible for creating, deleting, altering, and maintaining storage volumes in a system. Windows 2000 features significant improved manageability and recoverability of volumes in a Windows 2000 environment. This is achieved by the introduction of dynamic disks and a new snap-in. Windows 2000 supports two volume managers: basic disks and legacy Ftdisk volume sets are managed by FTDisk and the Ftdisk.sys driver; dynamic disks and all new volume sets are managed by Logical Disk Manager (LDM) and the Dmio.sys driver. Figure 2.6 shows the Disk Management architecture.

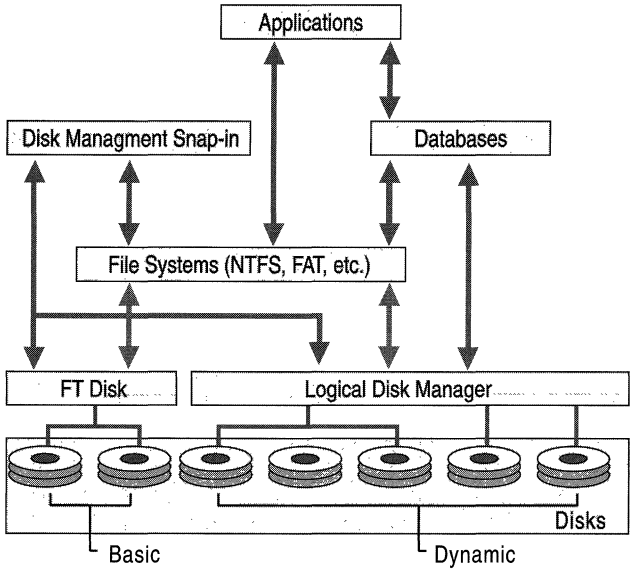


Figure 2.6 Disk Management Architecture

Important Unlike in Microsoft® Windows NT® version 4.0 and earlier, volume set configuration metadata is stored and replicated with other Disk Management metadata and is not stored in the registry.

Fault-Tolerant Disk Management

Ftdisk was used in Windows NT to manage partitions and all fault-tolerant volumes. In Windows 2000, FTDisk is used to manage basic partitions and fault-tolerant volumes from Windows NT 4.0.

Ftdisk manages basic disks. Basic disks can be partitioned with and recognized by Microsoft® MS-DOS®, Windows 95, Windows 98, and Windows NT. A basic partition does not provide fault tolerance or multiple disk volume functionality. Basic partition configuration is recorded in a few kilobytes of data at the beginning of the disk. Existing fault-tolerant volumes (Ftdisk sets) configured on Windows NT 4.0 can still be used on computers running Windows 2000 or converted to dynamic volumes.

Dynamic Disks

As shown in Figure 2.6, Logical Disk Manager manages dynamic disks, which can only be used on Windows 2000. Dynamic disk configuration uses a private database at the end of the disk in addition to the same kilobytes at the beginning of the disk that are used by basic disks. Any disk containing a volume managed by Disk Management contains this database.The database is replicated among all dynamic disks in the system. Dynamic disks can contain simple volumes, concatenated volumes, stripe volumes, mirrored volumes, and redundant array of independent disks (RAID) level 5 volumes. Dynamic volumes offer features, such as volume extension and fault tolerance configurations. Disk Management user interfaces, such as the Disk Management snap-in included with Windows 2000, interact with both Ftdisk and Disk Management functionality.

For more information about basic and dynamic disks, see "Disk Concepts and Troubleshooting" in this book.

Disk Quotas

Windows 2000 supports disk quotas for volumes formatted as NTFS. You can use disk quotas to monitor and limit disk-space use.

Disk quotas are tracked on a per-user, per-volume basis, and users are charged *only* for the files they own. Quotas are tracked independently for different volumes, even if the volumes are different partitions on the same physical drive. However, if you have shares on the *same* volume, the quotas assigned to that volume apply to all of these shares collectively, and users' utilization of both shares cannot exceed the assigned quota on that volume.

System administrators can use the **Quota** tab of the **Properties** dialog box to perform the following tasks:

- Enable or disable disk quotas on a volume.
- Prevent users from saving new data when their disk quota is exceeded.
- Set the default disk quota warning level and disk quota limit assigned to new volume users.
- View disk quota information for each user using the **Quota Entries** view.

Disk quotas track and control disk space usage for volumes. System administrators can configure Windows 2000 to perform the following tasks:

- Prevent further disk space use and log an event when a user exceeds a specified disk space limit.
- Log an event when a user exceeds a specified disk space warning level.

When you enable disk quotas, you can set both the disk quota limit and the disk quota warning level. The limit specifies the amount of disk space that is allocated to a user. The warning level specifies when a user is nearing the limit. For example, you can set a user's disk quota limit to 50 megabytes (MB), and the disk quota warning level to 45 MB. The user can store no more than 50 MB of data on the volume, and if more than 45 MB are stored on the volume, you can have the disk quota system log a system event.

When you enable disk quotas for a volume, volume usage is automatically tracked for new users, but existing volume users have no disk quotas applied to them. You can apply disk quotas to existing volume users by adding new quota entries in the **Quota Entries** window.

For more information about setting disk quotas, see Windows 2000 Server Help.

Disk Quotas and Free Space

Disk quotas are transparent to the user. When a user asks how much space is free on a disk, the system reports only the user's available quota allowance. If the user exceeds this allowance, the system indicates that the disk is full.

To obtain more free disk space after exceeding the quota allowance, the user must do one of the following:

- Delete files.
- Have another user claim ownership of some files.
- Have the administrator increase the quota allowance.

The following conditions apply when you use disk quotas:

- Disk quotas set on a volume apply only to that volume.
- Disk quotas cannot be set on individual files or folders.
- Disk quotas are based on uncompressed file sizes. You cannot increase the amount of free space by compressing the data.
- If your computer is configured as a multiple-boot system with Windows 2000 and Windows NT 4.0, you can exceed your limit when you are running Windows NT 4.0. However, when you are running Windows 2000, you must move files to a different partition or delete files until you are under your limit.
- To support disk quotas, a disk volume must be formatted with NTFS. Volumes formatted with previous versions of NTFS are upgraded automatically by Windows 2000 Setup.
- To administer quotas on a volume, you must be a member of the Administrators group on the computer where the drive resides.
- If the volume is not formatted with NTFS, or if you are not a member of the Administrators group on the local computer, the **Quota** tab is not displayed on the volume's **Properties** page.

Disk Quota Limits

The disk space used by each file is charged directly to the user who owns the file. The file owner is identified by the security identifier (SID) in the security information for the file. The total disk space charged to a user is the sum of the length of all data streams, and property set streams and resident user data streams affect the user's quota. Compressing or decompressing files does not affect the disk space reported for the files. Therefore, quota settings on one volume can be compared to settings on another volume.

The following are types of disk quota limits.

Warning threshold You can configure the system to generate a system log file entry when the disk space charged to the user exceeds this value.

Hard quota You can configure the system to generate a system log file entry or deny additional disk space to the user when the disk space charged to the user exceeds this value.

NTFS automatically creates a user quota entry when a user first writes to the volume. Entries that are created automatically are assigned the default warning threshold and hard quota limit values for the volume.

Disk Quotas States

The administrator can turn quota enforcement on and off. There are three quota states, as shown in Table 2.13.

Table 2.13 Disk Quota States

State	Description
Quota disabled	Quota usage changes are not tracked, but the quota limits are not removed. In this state, performance is not affected by disk quotas. This is the default state.
Quota tracked	Quota usage changes are tracked, but quota limits are not enforced. In this state, no quota violation events are generated and no file operations fail because of disk quota violations.
Quota enforced	Quota usage changes are tracked and quota limits are enforced.

Administering Disk Quotas

Disk quotas monitor volume use to prevent users from affecting others' use of the volume. For example, if a user saves 50 MB on a volume on which each user has been allocated 50 MB of space, some of this data must be moved or deleted before additional data is written to the volume. Other users can continue to save up to 50 MB of space on that volume.

Note Disk Quotas do not prevent adminstrators from allocating more space than is available on the disk. For example, on a 1 GB volume that is used by 100 users, each user might be allocated 100 MB of space to allow each user a reasonable amount of disk space.

Disk quotas are based on file ownership and are independent of the location of the files on the volume. If a user moves files from one folder to another on the same volume, volume space usage does not change; if the user copies the files to a different folder on the same volume, the volume space usage doubles.

The administrator can set default quotas for the volume or quotas for specific users on a volume. A new user receives the default quota unless the administrator established a quota specifically for that user. The administrator can view the level of quota tracking, the default quota limits, and the per-owner quota information. The per-user quota information contains the user's hard quota limit, warning threshold, and quota usage.

If you do not want to use the default disk space limit and warning threshold values for a particular user, use the New Quota Entry feature to set up quota thresholds and limits before the user actually writes data to the volume.

User quota entries cannot be deleted if a user still owns files on the volume; all files owned by that user must either be deleted or moved to another volume, or ownership of the files must be transferred to another user.

Enabling Disk Quotas

When you enable quotas on a volume that already contains files, the disk space used by all users who have copied, saved, or taken ownership of files on the volume up to that point is calculated. The quota limit and warning level are then applied to all current users and to all new users. You can then disable or set different quotas for specific users. You can also set quotas for specific users who have not yet copied, saved, or taken ownership of files on the volume.

For example, you can set a quota of 5 MB for all users of \\Main\General, while ensuring that two users who work with larger files have a 10 MB limit. If both users already have files stored on \\Main\General, select both users and set their quota limit to 10 MB. However, if one or both users do not have files stored on the server when you enable quotas, use the **Select Users** property sheet to set their quota limit to a value higher than the default for new users.

Local and Remote Implementations

Disk quotas can be enabled on volumes residing on both local computers and remote computers. On local computers, quotas can be used to limit the amount of space available to users who log on to the local computer. On remote computers, quotas can limit volume usage by remote users.

You can use quotas to ensure the following:

- Multiple users can share resources on the same computer.
- Disk space on public servers is not monopolized by one or more users.
- Users do not use excessive disk space on a shared folder on your computer.

To enable quotas on remote volumes, they must be formatted as the version of NTFS included with Windows 2000 and be shared from the root directory of the volume. Also, you must be a member of the Administrators group on the remote computer to enable and manage quotas.

System files are included in the volume usage of the person who installed Windows 2000 on the local computer. When implementing disk quotas on a local volume, make sure to take into account the disk space used by these files. Depending on the free space available on the volume, you might want to set a high quota limit or no limit for the user who installed the operating system.

Auditing Disk Space Use

Enabling quotas causes a slight increase in server overhead and a slight decrease in file server performance. By periodically enabling and then disabling quotas, you can take advantage of the auditing capabilities provided by Windows 2000 disk quotas without reducing the performance of your file server.

To create a record of the audit, save a copy of the data to another application, such as Microsoft® Excel.

Exceeding Disk Quota Limits

When you select the **Deny disk space to users exceeding quota limit** option, users who exceed their limit receive an "insufficient disk space" error and cannot write additional data to the volume without deleting or moving files. Individual programs determine their own error handling for this condition. To the program, it appears that the volume is full.

By leaving this option cleared, you can allow users to exceed their limit. This is useful when you do not want to deny users access to a volume, but want to track disk space use on a per-user basis. You can also specify whether or not to log an event when users exceed either their quota warning level or their quota limit.

When you select the **Log event when a user exceeds their quota limit** option, an event is written to the system log when a user exceeds the limit. Administrators can view these events with Event Viewer, filtering for Disk event types. Unless you set a trigger to do so, users are not warned of this event.

Event Viewer builds a historical, chronological record of which users exceeded their quota warning level and quota limits, and when they exceeded them. However, it does not provide information about which users are currently over their quota warning level.

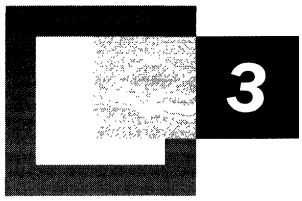
For more information about enabling disk quotas, see Windows 2000 Server Help.

Additional Resources

- For more information about storing and managing data, see the Microsoft Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- For more information about Disk Management and disk quotas, see Windows 2000 Server Help.

CHAPTER 3

File Systems



Microsoft® Windows® 2000 supports the NTFS file system, two file allocation table (FAT) file systems (FAT16 and FAT32), the compact disc file system (CDFS), and the Universal Disk Format (UDF). The structures of the volumes formatted by each of these file systems, as well as the way each file system organizes data on the disk, are significantly different. The capabilities and limitations of these file systems must be reviewed to determine their comparative features. The version of NTFS included with Windows 2000 includes reparse points, the change journal, encryption, sparse file support, and several other new features.

In This Chapter

About Windows 2000 File Systems 111
Comparing FAT16, FAT32, and NTFS 130
File and Folder Compression 141
NTFS Recoverability 148
Features Built on Reparse Points 150
File System Tools 151

Related Information in the Resource Kit

- For information about disks, see "Disk Concepts and Troubleshooting" in this book.
- For information about disk storage, see "Data Storage and Management" in this book.
- For information about system recovery, see "Repair, Recovery, and Restore" in this book.

About Windows 2000 File Systems

An operating system's ability to access files on a volume depends on the file system with which the volume was formatted. Table 3.1 shows the file system format used by various operating systems.

Table 3.1 Operating System and File System Compatibility

Operating system	File System Format of Volume
Windows 2000	NTFS
	FAT16
	FAT32
Microsoft® Windows NT®	NTFS
	FAT16
Microsoft® Windows® 95 OEM Service Release 2 (OSR2) and Microsoft® Windows® 98	FAT16
	FAT32
Windows 95 prior to version OSR2	FAT16
Microsoft® MS-DOS®	FAT16

Note FAT16 and FAT32 are referred to synonymously as FAT unless the differences between them must be noted.

You can use long and short file names in both NTFS and FAT volumes. A long file name can be up to 256 characters long. Short file names contain an eight-letter file name and a three-letter file name extenstion and use the format *xxxxxxxx.yyy*. Short file names are compatible with MS-DOS.

FAT File System

The FAT file system has the file allocation table located at the beginning of a logical volume. FAT was designed for small disks and simple folder structures. Two copies of the file allocation table are stored on the volume. In the event that one copy of the file allocation table is corrupted, the other file allocation table is used.

FAT16 File System

FAT16 is included in Windows 2000 for the following reasons:

- It provides backward compatibility in the form of an upgrade path for earlier versions of Windows-compatible products.
- It is compatible with most other operating systems.

For Windows 2000 and Windows NT, the maximum size for a FAT16 volume is 4,095 megabytes (MB).

A volume formatted with FAT16 is allocated in clusters. The default cluster size is determined by the volume size, and can be as large as 64 kilobytes (KB). The cluster size must be a power of 2 between 512 and 65,536 bytes. Table 3.2 shows the default cluster sizes for FAT16 volumes. You can specify a different cluster size if you format the volume with the **format** command from the command prompt. However, the size you specify must be listed in Table 3.2.

Table 3.2 FAT16 Cluster Sizes

Volume Size	Sectors Per Cluster	Cluster Size
0 MB–32 MB	1	512 bytes
33 MB–64 MB	2	1 KB
65 MB–128 MB	4	2 KB
129 MB–255 MB	8	4 KB
256 MB–511 MB	16	8 KB
512 MB–1,023 MB	32	16 KB
1,024 MB–2,047 MB	64	32 KB
2,048 MB–4,095 MB	128	64 KB

FAT16 is not recommended for volumes larger than 511 MB because, when relatively small files are placed on a FAT16 volume, FAT uses disk space inefficiently. You cannot use FAT16 on volumes larger than 4 gigabytes (GB), regardless of the cluster size.

Note On volumes with fewer than 32,680 sectors, the cluster sizes can be up to 8 sectors per cluster. The format program, whether you format the volume using Disk Management or by typing **format** at the command prompt, creates a 12-bit FAT. Volumes less than 16 MB are usually formatted for a 12-bit FAT, but the exact size depends on the disk geometry. The disk geometry also determines the point at which a larger cluster size is needed because the number of clusters on the volume must fit into 16 bits. Therefore, you might have a 33-MB volume that still has only 1 sector per cluster.

FAT12 is the original implementation of FAT and is intended for very small media. The file allocation table for FAT12 is smaller than the file allocation table for FAT16 and FAT32, because it uses less space for each entry. This leaves more space for data. All 5.25-inch floppy disks are formatted with FAT12, and 1.44-MB 3.5-inch floppy disks are generally formatted with FAT12. Volumes on Iomega Zip and Jaz drives are formatted with FAT16.

Structure of a FAT16 Volume

Figure 3.1 illustrates how FAT maps out clusters on a volume. The file allocation table (areas FAT1 and FAT2 in Figure 3.1) identifies each cluster in the volume as one of the following:

- Unused
- Cluster in use by a file
- Bad cluster
- Last cluster in a file

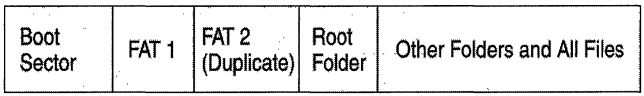


Figure 3.1 Organization of a FAT Volume

The only difference between the root folder and other folders is that the root folder is at a specified location and has a fixed number of entries (for a hard disk). The number of entries on a floppy disk depends on the size of the disk.

Note Each folder and file in the root folder uses one or more entries. For example, if the fixed number of entries is 512 and you have 100 folders, you can only create 412 files.

Folders have a 32-byte entry for each file and folder contained in the folder. Assuming short file names are used, the entry includes the following information:

- Name (8.3) xxxxxxxx.yyy. (88 bits)
- Attribute byte (8 bits of information, described later in this section).
- One reserved byte.
- Create time (24 bits).
- Create date (16 bits).
- Last access date (16 bits).
- Two reserved bytes.
- Last modified time (16 bits).
- Last modified date (16 bits).
- Starting cluster number in the file allocation table (16 bits).
- File size (32 bits).

In a FAT folder structure, files are given the first available location on the volume. The starting cluster number is the address of the first cluster used by the file. Each cluster contains a pointer to the next cluster in the file, or an end-of-file (EOF) indicator at (0xFFFF) which indicates that this cluster is the end of the file. These pointers and end-of-file indicators are shown in Figure 3.2.

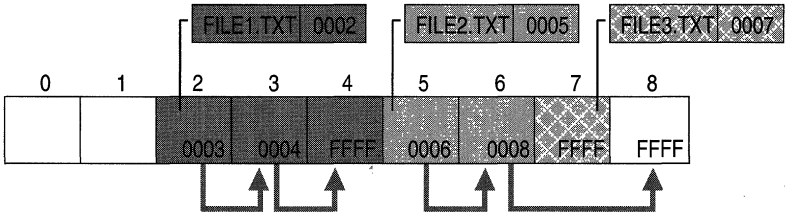


Figure 3.2 Files on a FAT Volume

Figure 3.2 shows three files in a folder. File1.txt is large enough to use three clusters. File2.txt is a fragmented file that also requires three clusters. The third file, File3.txt, fits completely in one cluster. In each case, the folder entry points to the first cluster of the file.

The information in the folder is used by all operating systems that support FAT. Windows 2000 can store additional time stamps in a FAT folder entry. These time stamps show when the file was created or last accessed.

Because all entries in a directory are the same size, the attribute byte for each entry in a directory describes what kind of entry it is. For example, one bit indicates that the entry is for a subdirectory and another bit marks the entry as a volume. Typically, only the operating system controls the settings of these bits.

The attribute byte includes four bits that can be turned on or off by the user—archive, system, hidden, and read-only.

FAT32 File System

Support for FAT32 is new in Windows 2000. FAT16 supports volumes up to 4 GB, whereas theoretically FAT32 can manage volumes up to 2 terabytes. The FAT32 on-disk format and features on Windows 2000 are similar to those on Windows 95 OSR2 and Windows 98.

The size of a FAT32 cluster can range in size from 1 sector (512 bytes) to 64 sectors (32 KB), incremented in powers of 2.

Since FAT32 requires 4 bytes to store cluster values, many internal and on-disk data structures have been revised or expanded. Most programs are unaffected by these changes; however, disk utilities which read the on-disk format must be updated to support FAT32.

Two application programming interfaces (APIs) are disabled on FAT32. Since the BIOS parameter block (BPB) structure grows from 25 bytes on FAT16 to 53 bytes on FAT32, and FSCTL_QUERY_FAT_BPB is defined to return only as much of the boot sector as contains the 25 byte form of the BPB, it is disabled on FAT32. To retrieve the BPB on FAT32, applications should read the volume directly. This also works for FAT16. Extended attributes are disabled on FAT32 since increasing the cluster number to 4 bytes requires the use of the field previously used to index the extended attribute database.

Structure of a FAT32 Volume

The main difference between FAT16 and FAT32 is the logical partition size. FAT32 breaks the 2-GB logical drive limitation of FAT16 volumes by extending a single logical drive capacity to at least 127 GB. If you have a 2-GB FAT16 drive, you must use a 32-KB cluster. With FAT32, the range for a 4-KB cluster, for example, includes drive sizes between 512 MB and 8 GB.

The largest possible file for a FAT32 drive is 4 GB minus 2 bytes. FAT32 uses 4 bytes per cluster within the file allocation table. This differs from FAT16, which uses 2 bytes per cluster within the file allocation table. Table 3.3 shows the default cluster sizes for FAT32.

Table 3.3 FAT32 Cluster Sizes

Partition Size	Default Cluster Size
Less than 8 GB	4 K
Greater than or equal to 8 GB, and less than 16 GB	8 K
Greater than or equal to 16 GB, and less than 32 GB	16 K
Greater than or equal to 32 GB	32 K

A FAT32 volume must have at least 65,527 clusters. Also, the cluster size on a FAT32 volume cannot be such that the file allocation table is greater than (16 MB – 64 KB)/4, or almost 4 million clusters.

FAT16 and FAT32 do not scale well. As the volume gets bigger, the file allocation table gets bigger, which dramatically increases the amount of time it takes Windows 2000 to compute how much free space is on the boot volume when the system is restarted.

For this reason, you may not create a FAT32 volume larger than 32 GB using the Format utility. However, the Windows 2000 Fastfat driver enables you to mount and fully support a FAT32 volume larger than 32 GB.

Use NTFS to format volumes larger than these. For more information about why you should format all Windows 2000 partitions with NTFS, see "Advantages of NTFS" later in this chapter.

File Names on FAT Volumes

Files created or renamed on FAT volumes use attribute bits to support long file names in a way that does not interfere with how MS-DOS gains access to the volume.

Whenever you create a file with a long file name, Windows 2000 creates a conventional 8.3 name for the file and one or more secondary folder entries for the file, one for each 13 characters in the long file name. Each secondary folder entry stores a corresponding part of the long file name in Unicode.

Windows 2000 marks the secondary folder entries as part of a long file name by setting the volume ID, read-only, system, and hidden attribute bits. MS-DOS generally ignores folder entries with all four of these attribute bits set, so these entries are invisible to these operating systems. MS-DOS accesses the file by using the conventional 8.3 file name contained in the folder entry for the file.

Figure 3.3 shows all of the folder entries for the file Thequi~1.fox, which has a long name of The quick brown.fox. The long name is in Unicode, so each character in the name uses 2 bytes in the folder entry. The attribute field for the long-name entries has the value **0x0F**. The attribute field for the short name has the value **0x20**.

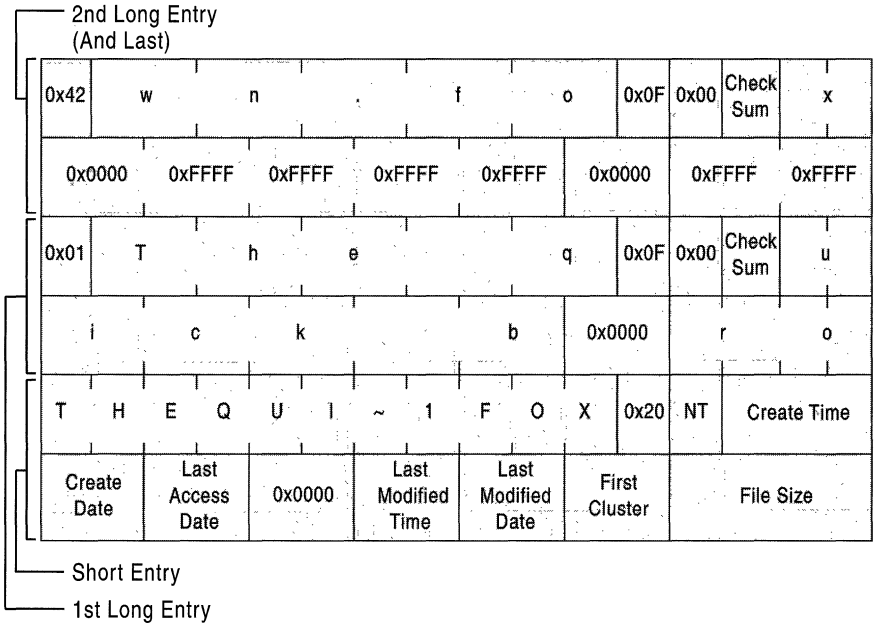


Figure 3.3 Long File Name on a FAT Volume

Note Windows NT and Windows 2000 do not use the same algorithm to create long and short file names as Windows 95 and Windows 98. However, on computers that multiple-boot these operating systems, files that you create when running one operating system can be accessed when running the other.

For information about how Windows 2000 creates short file names, see "Using Long File Names" later in this chapter.

By default, Windows 2000 supports long file names on FAT volumes. You can prevent a FAT file system from creating long file names by setting the value of the **Win31FileSystem** registry entry (in HKEY_LOCAL_MACHINE\System \CurrentControlSet\Control\FileSystem\Win31FileSystem) to **1**.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

This value prevents Windows 2000 from creating new long file names on all FAT volumes, but it does not affect existing long file names.

Using FAT with Windows 2000

FAT16 works the same way in Windows 2000 as it does in MS-DOS, Windows 3.*x*, Windows 95, and Windows 98. FAT32 works the same way in Windows 2000 as it does in Windows 95 OSR2 and Windows 98. In fact, you can install Windows 2000 on an existing FAT primary partition or logical drive. When running Windows 2000, you can move or copy files between FAT and NTFS volumes.

Note You cannot use Windows 2000 with any compression or partitioning software that requires disk drivers to be loaded by MS-DOS.

The NTFS File System

Windows 2000 comes with a new version of NTFS. This newest version of NTFS provides performance, reliability, and functionality not found in FAT. Some new features in Windows 2000, such as Active Directory™ directory service and the storage features based on reparse points are only available on volumes formatted with NTFS.

NTFS also includes security features required for file servers and high-end personal computers in a corporate environment, and data access control and ownership privileges important for data integrity.

Multiple Data Streams

NTFS supports multiple data streams, where the stream name identifies a new data attribute on the file. A handle can be opened to each data stream. A data stream, then, is a unique set of file attributes. Streams have separate opportunistic locks, file locks, and sizes, but common permissions.

This feature enables you to manage data as a single unit. The following is an example of an alternate stream:

```
myfile.dat:stream2
```

A library of files might exist where the files are defined as alternate streams, as in the following example:

```
library:file1
       :file2
       :file3
```

A file can be associated with more than one application at a time, such as Microsoft® Word and Microsoft® WordPad. For instance, a file structure like the following illustrates file association, but not multiple files:

```
program:source_file
       :doc_file
       :object_file
       :executable_file
```

You can use the Win32 advanced programming interface (API) **CreateFile** to create an alternate data stream. Or, at the command prompt, you can type commands such as:

```
echo text>program:source_file
more <program:source_file
```

Caution Because NTFS is not supported on floppy disks, when you copy an NTFS file to a floppy disk, data streams and other attributes not supported by FAT are lost without warning.

Reparse Points

Reparse points are new file system objects in the version of NTFS included with Windows 2000. Reparse points have a definable attribute containing user-controlled data and are used to extend functionality in the input/output (I/O) subsystem.

For more information about reparse points, see the Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Change Journal

The change journal is used by NTFS to provide a persistent log of all changes made to files on the volume. For each volume, NTFS uses the change journal to track information about added, deleted, and modified files. The change journal is much more efficient than time stamps or file notifications for determining changes in a given namespace.

The change journal is implemented as a sparse stream in which only a small active range uses any disk allocation. The active range initially begins at offset 0 in the stream and moves monotonically forward. The unique sequence number (USN) of a particular record represents its virtual offset in the stream. As the active range moves forward through the stream, earlier records are deallocated and become unavailable. The size of the active range in a sparse file can be adjusted. For more information about the change journal and sparse files, see the Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Encryption

File and directory-level encryption is implemented in the version of NTFS included with Windows 2000 for enhanced security in NTFS volumes. Windows 2000 uses Encrypting File System (EFS) to store data in encrypted form, which provides security when the storage media are removed from a system running Windows 2000. For more information about EFS, see the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

Sparse File Support

Sparse files allow programs to create very large files, but to consume disk space only as needed. A sparse file is a file with an attribute that causes the I/O subsystem to allocate the file's meaningful (nonzero) data. All nonzero data is allocated on disk, whereas all nonmeaningful data (large strings of data composed of zeros) is not. When a sparse file is read, allocated data is returned as it was stored, and nonallocated data is returned, by default, as zeros in accordance with the C2 security requirement specification.

NTFS includes full sparse file support for both compressed and uncompressed files. NTFS handles read operations on sparse files by returning allocated data and sparse data. It is possible to read a sparse file as allocated data and a range of data without having to retrieve the entire data set, although, by default, NTFS returns the entire data set.

You can set a user-controlled file system attribute to take advantage of the sparse file function in NTFS. With the sparse file attribute set, the file system can deallocate data from anywhere in the file and, when an application calls, yield the zero data by range instead of storing and returning the actual data. File system APIs allow for the file to be copied or backed as actual bits and sparse stream ranges. The net result is efficient file system storage and access. Figure 3.4 shows how data is stored with and without the sparse file attribute set.

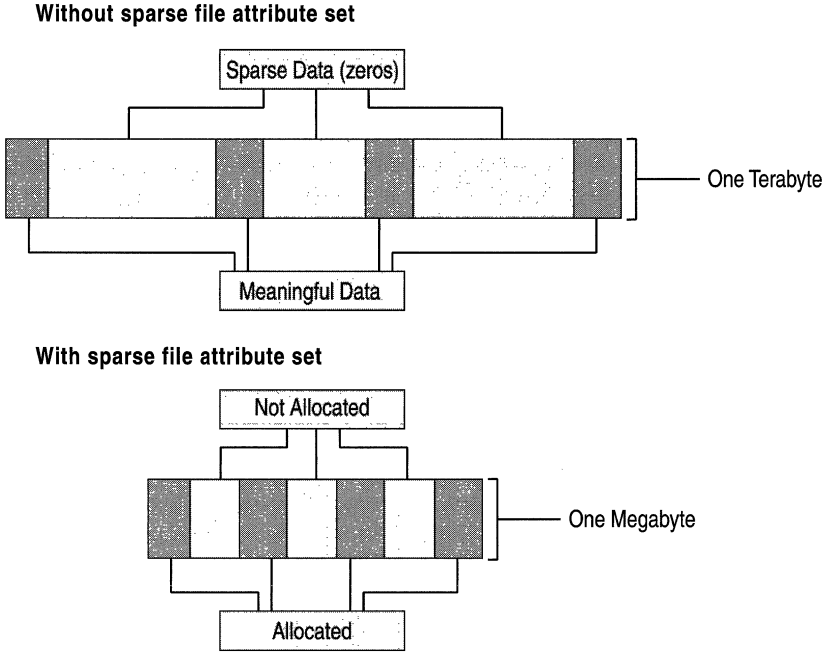


Figure 3.4 Sparse Data Storage

Disk Quotas

Disk quotas are a new feature in NTFS that provide more precise control of network-based storage. Disk quotas are implemented on a per-volume basis and enable both hard and soft storage limits to be implemented on a per-user basis. For more information about disk quotas, see "Data Storage and Management" in this book.

The introduction of distributed file system (Dfs), NTFS directory junctions, and volume mount points also creates situations where logical directories do not have to correspond to the same physical volume. Available disk space is based on user context, and the space reported for a volume is not necessarily representative of the space available to the user. For this reason, do not rely on space queries to make assumptions about the amount of available disk space in directories other than the current one. For more information about Dfs, see the *Distributed Systems Guide*. For more information about volume mount points, see "Volume Mount Points" later in this chapter.

Distributed Link-Tracking

Windows 2000 provides a distributed link-tracking service that enables client applications to track link sources that have been moved locally or within a domain. Clients that subscribe to this link-tracking service can maintain the integrity of their references because the objects referenced can be moved transparently. Files managed by NTFS can be referenced by a unique object identifier. Link tracking stores a file's object identifier as part of its tracking information.

The distributed link-tracking service tracks shell shortcuts and OLE links within NTFS volumes on computers running Windows 2000. For example, if a shell shortcut is created to a text document, distributed link-tracking allows the shortcut to remain correct, even if the target file moves to a new drive or computer system. Similarly, in a Microsoft® Word document that contains an OLE link to a Microsoft® Excel spreadsheet, the link remains correct even if the Excel file moves to a new drive or computer system.

If a link is made to a file on a volume formatted with the version of NTFS included with Windows 2000, and the file is moved to any other volume with the same version of NTFS within the same domain, the file is found by the tracking service, subject to time considerations. Additionally, if the file is moved outside the domain or within a workgroup, it is likely to be found.

Converting to Windows 2000 File Systems

The on-disk format for NTFS has been enhanced in Windows 2000 to enable new functionality. The upgrade to the new on-disk format occurs when Windows 2000 mounts an existing NTFS volume. The upgrade is quick and automatic; the conversion time is independent of volume size. Note that FAT volumes can be converted to NTFS format at any time using the Convert.exe utility.

Important Performance of volumes that have been converted from FAT is not as high as volumes that were originally formatted with NTFS.

Multiple Booting of Windows NT and Windows 2000

Your ability to access your NTFS volumes when you multiple-boot Windows NT and Windows 2000 depends on which version you are using. (Redirected clients using NTFS volumes on file and print servers are not affected.)

Windows NT Compatibility with the Version of NTFS Included with Windows 2000

When a Windows 2000 volume is mounted on a system running Windows NT 4.0 Service Pack 4, most features of the version of NTFS included with Windows 2000 are not available. However, most read and write operations are permitted if they do not make use of any new NTFS features. Features affected by this configuration include the following:

- **Reparse points.** Windows NT cannot use any features based on reparse points, such as Remote Storage and volume mount points.
- **Disk quotas.** When running Windows NT, Windows 2000 disk quotas are ignored. This allows you to allocate more disk space than is allowed by your quota.
- **Encryption.** Windows NT cannot perform any operations on files encrypted by Windows 2000.
- **Sparse files.** Windows NT cannot perform any operations on sparse files.
- **Change journal.** Windows NT ignores the change journal. No entries are logged when files are accessed.

Cleanup Operations on Windows NT Volumes

Because files on volumes formatted with the version of NTFS included with Windows 2000 can be read and written to by Windows NT, Windows 2000 may need to perform cleanup operations to ensure the consistency of the data structures of a volume after it was mounted on a computer that is running Windows NT. Features affected by cleanup operations are explained below.

Disk quotas If disk quotas are turned off, Windows 2000 performs no cleanup operations. If disk quotas are turned on, Windows 2000 cleans up the quota information.

If a user exceeds the disk quota while the NTFS volume is mounted by a Windows NT 4.0 system, all further disk allocations of data by that user will fail. The user can still read and write data to any existing file, but will not be able to increase the size of a file. However, the user can delete and shrink files. When the user gets below the assigned disk quota, he or she can resume disk allocations of data. The same behavior occurs when a system is upgraded from a Windows NT system to a Windows 2000 system with quotas enforced.

Reparse points Because files that have reparse points associated with them cannot be accessed by computers that are running Windows NT 4.0 or earlier, no cleanup operations are necessary in Windows 2000.

Encryption Because encrypted files cannot be accessed by computers that are running Windows NT 4.0 or earlier, no cleanup operations are necessary.

Sparse files Because sparse files cannot be accessed by computers that are running Windows NT 4.0 or earlier, no cleanup operations are necessary.

Object identifiers Windows 2000 maintains two references to the object identifier. One is on the file; the other is in the volume-wide object identifier index. If you delete a file with an object identifier on it, Windows 2000 must scan and clean up the leftover entry in the index.

Change journal Computers that are running Windows NT 4.0 or earlier do not log file changes in the change journal. When Windows 2000 starts, the change journals on volumes accessed by Windows NT are reset to indicate that the journal history is incomplete. Applications that use the change journal must have the ability to accept incomplete journals.

Structure of an NTFS Volume

Like FAT, NTFS uses clusters as the fundamental unit of disk allocation. In the Disk Management snap-in, you can specify a cluster size of up to 4 KB. If you type **format** at the command prompt to format your NTFS volume, but do not specify an allocation unit size using the **/A:<size> switch**, the values in Table 3.4 will be used.

Table 3.4 Default Cluster Sizes for NTFS

Volume Size	Sectors Per Custer	Default Cluster Size
512 MB or less	1	512 bytes
513 MB–1,024 MB (1 GB)	2	1,024 bytes (1 KB)
1,025 MB–2,048 MB (2 GB)	4	2,048 bytes (2 KB)
Greater than 2,049 MB	8	4 KB

Note Windows 2000, like Windows NT 3.51 and Windows NT 4.0, supports file compression. Since file compression is not supported on cluster sizes above 4 K, the default NTFS cluster size for Windows 2000 never exceeds 4 K. For more information about NTFS compression, see "File and Folder Compression" later in this chapter.

Boot Sector

The first information found on an NTFS volume is the boot sector. The boot sector starts at sector 0 and can be up to 16 sectors long. It consists of two structures:

- The BIOS parameter block, which contains information on the volume layout and file system structures.
- Code that describes how to find and load the startup files for the operating system being loaded. For Windows 2000, this code loads the file Ntldr. For more information about the boot sector, see "Disk Concepts and Troubleshooting" in this book.

Master File Table and Metadata

When a volume is formatted with NTFS, a Master File Table (MFT) file and other pieces of metadata are created. Metadata are the files NTFS uses to implement the file system structure. NTFS reserves the first 16 records of the MFT for metadata files.

Note The data segment locations for both $Mft and $MftMirr are recorded in the boot sector. If the first MFT record is corrupted, NTFS reads the second record to find the MFT mirror file. A duplicate of the boot sector is located at the end of the volume.

Table 3.5 lists and briefly describes the metadata stored in the MFT.

Table 3.5 Metadata Stored in the Master File Table

System File	File Name	MFT Record	Purpose of the File
Master file table	$Mft	0	Contains one base file record for each file and directory on an NTFS volume. If the allocation information for a file or directory is too large to fit within a single record, other file records are allocated as well.
Master file table 2	$MftMirr	1	A duplicate image of the first four records of the MFT. This file guarantees access to the MFT in case of a single-sector failure.
Log file	$LogFile	2	Contains a list of transaction steps used for NTFS recoverability. Log file size depends upon the volume size. It is used by Windows 2000 to restore consistency to NTFS in the event of a system failure. For more information about the log file, see "NTFS Recoverability" later in this chapter.

(continued)

Table 3.5 Metadata Stored in the Master File Table ***(continued)***

System File	File Name	MFT Record	Purpose of the File
Volume	$Volume	3	Contains information about the volume, such as the volume label and the volume version.
Attribute definitions	$AttrDef	4	A table of attribute names, numbers, and descriptions.
Root file name index	$	5	The root directory.
Cluster bitmap	$Bitmap	6	A representation of the volume showing which clusters are in use.
Boot sector	$Boot	7	Includes the bootstrap for the volume if it is a bootable volume.
Bad cluster file	$BadClus	8	Contains bad clusters for the volume.
Security file	$Secure	9	Contains unique security descriptors for all files within a volume.
Upcase table	$Upcase	10	Converts lowercase characters to matching Unicode uppercase characters.
NTFS extension file	$Extend	11	Used for various optional extensions such as quotas, reparse point data, and object identifiers.
		12–15	Reserved for future use.

The remaining records of the MFT contain the file and directory records for each file and directory on the volume.

NTFS creates a file record for each file and a directory record for each directory created on an NTFS volume. The MFT includes a separate file record for the MFT itself. These file and directory records are stored on the MFT. The attributes of the file are written to the allocated space in the MFT. Besides file attributes, each file record contains information about the position of the file record in the MFT.

Each file usually uses one file record. However, if a file has a large number of attributes or becomes highly fragmented, it may need more than one file record. If this is the case, the first record for the file, called the base file record, stores the location of the other file records required by the file. Small files and directories (typically 1,500 bytes or smaller) are entirely contained within the file's MFT record.

Directory records contain index information. Small directories might reside entirely within the MFT structure, while large directories are organized into B-tree structures and have records with pointers to external clusters that contain directory entries that could not be contained within the MFT structure.

NTFS File Attributes

Every allocated sector on an NTFS volume belongs to a file. Even the file system metadata is part of a file. NTFS views each file (or folder) as a set of file attributes. Elements such as the file's name, its security information, and even its data, are all file attributes. Each attribute is identified by an attribute type code and, optionally, an attribute name.

When a file's attributes can fit within the MFT file record for that file, they are called resident attributes. Information such as file name and time stamp are always resident attributes. When the information for a file is too large to fit in its MFT file record, some of the file attributes are nonresident. Nonresident attributes are allocated one or more clusters of disk space and stored as an alternate data stream in the volume. NTFS creates the Attribute List attribute to describe the location of both resident and nonresident attribute records.

Table 3.6 lists the file attributes defined by NTFS, although other file attributes might be defined in the future.

Table 3.6 NTFS File Attribute Types

Attribute Type	Description
Standard Information	Includes information such as time stamp and link count.
Attribute List	Lists the location of all the attribute records that do not fit in the MFT record.
File Name	A repeatable attribute for both long and short file names. The long name of the file can be up to 255 Unicode characters. The short name is the MS-DOS-readable, 8.3, case-insensitive name for the file. Additional names, or hard links, required by POSIX can be included as additional file name attributes.
Security Descriptor	Shows information about who owns the file and who can access the file.
Data	Contains file data. NTFS allows multiple data attributes per file. Each file typically has one unnamed data attribute. A file can also have one or more named data attributes, each using a particular syntax.
Object ID	A volume-unique file identifier. Used by the link tracking service. Not all files have object identifiers.
Logged Tool Stream	Similar to a data stream, but operations on a logged tool stream are logged to the NTFS log file just like NTFS metadata changes. Used by EFS.
Reparse Point	Used for directory junction points and volume mount points. They are also used by file system filter drivers to mark certain files as special to that driver.

(continued)

Table 3.6 NTFS File Attribute Types *(continued)*

Attribute Type	Description
Index Root	Used to implement folders and other indexes.
Index Allocation	Used to implement folders and other indexes.
Bitmap	Used to implement folders and other indexes.
Volume Information	Used only in the $Volume system file. Contains the volume version.
Volume Name	Used only in the $Volume system file. Contains the volume label.

MS-DOS-Readable File Names on NTFS Volumes

By default, Windows NT and Windows 2000 generate MS-DOS-readable file names on all NTFS volumes. To improve performance on volumes with many long, similar names, you can change the default value of the registry entry **NtfsDisable8dot3NameCreation** (in HKEY_LOCAL_MACHINE\System \CurrentControlSet\Control\FileSystem) to **1**.

Windows 2000 does not generate short (8.3) file names for files created by POSIX-based applications on an NTFS volume, regardless of the value of the **NtfsDisable8dot3NameCreation** registry entry. This means that MS-DOS-based and 16-bit Windows-based applications cannot view these file names if they are not valid 8.3 file names. Use standard MS-DOS 8.3 naming conventions if you want to use files that are created by a POSIX application with MS-DOS-based or Windows-based applications.

Using Long File Names

File names on Windows NT and Windows 2000 platforms can be up to 255 characters, and can contain spaces, multiple periods, and special characters that are forbidden in MS-DOS file names. Windows 2000 makes it possible for other operating systems to access files with long names by automatically generating an MS-DOS-readable (8.3) name for each file. Files are accessible over a network by computers using MS-DOS and Windows 3.*x*, as well as by computers using Windows 95, Windows 98, Windows NT, and Windows 2000 operating systems.

By creating 8.3 file names for files, Windows 2000 also enables MS-DOS-based and Windows 3.*x*–based applications to recognize and load files that have long file names. In addition, when an application saves a file on a computer running Windows 2000, both the 8.3 file name and long file name are retained.

If the long name of a file or folder contains spaces, you must surround the name with quotation marks. For example, if you have a program called Dump Disk Files that you want to run from the command line and you enter the name without quotation marks, it generates the error message "Cannot find the program Dump or one of its components."

You must also use quotation marks when a path typed at the command line includes spaces, as in the following example:

```
move "c:\This month's reports\*.*" "c:\Last month's reports"
```

Use wildcard characters such as the asterisk (*) and question mark (?) carefully in conjunction with the **del** and **copy** command-line commands. Windows 2000 searches both long and short file names for matches to the wildcard character combination you specify, which can cause additional files to be deleted or copied.

Both FAT and NTFS use the Unicode character set for their names, which contain several forbidden characters that MS-DOS cannot read in any file name. To generate a short MS-DOS-readable file name for a file, Windows 2000 deletes all of these characters from the long file name and removes any spaces. Because an MS-DOS-readable file name can have only one period, Windows 2000 also removes all extra periods from the file name. Next, Windows 2000 truncates the file name, if necessary, to six characters and appends a tilde (~) and a number. For example, each nonduplicate file name is appended with **~1**. Duplicate file names end with **~2**, then **~3**, and so on. After the file names are truncated, the file name extensions are truncated to three or fewer characters. Finally, when displaying file names at the command line, Windows 2000 translates all characters in the file name and extension to uppercase.

Note You can permit extended characters by setting the value of HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem \NtfsAllowExtendedCharacterIn8dot3Name to 1.

When there are five or more files that would result in duplicate short file names, Windows 2000 uses a slightly different method for creating short file names. For the fifth and subsequent files, Windows 2000:

- Uses only the first two letters of the long file name.
- Generates the next four letters of the short file name by mathematically manipulating the remaining letters of the long file name.
- Appends **~1** (or another number, if necessary, to avoid a duplicate file name) to the result.

This method provides substantially improved performance when Windows 2000 must create short file names for a large number of files with similar long file names. Windows 2000 uses this method to create short file names for both FAT and NTFS files.

Table 3.7 shows the short file names for files that were created in the order test 1 through test 6.

Table 3.7 Short File Names Created by Windows 2000 – Example One

Long File Name	Short File Name
This is test 1.txt	THISIS~1.TXT
This is test 2.txt	THISIS~2.TXT
This is test 3.txt	THISIS~3.TXT
This is test 4.txt	THISIS~4.TXT
This is test 5.txt	TH0FF9~1.TXT
This is test 6.txt	THFEF5~1.TXT

If the long file names in Table 3.7 are created in a different order, their short file names are different, as shown in Table 3.8.

Table 3.8 Short File Names Created by Windows 2000 – Example Two

Long File Name	Short File Name
This is test 2.txt	THISIS~1.TXT
This is test 3.txt	THISIS~2.TXT
This is test 1.txt	THISIS~3.TXT
This is test 4.txt	THISIS~4.TXT
This is test 5.txt	TH0FF9~1.TXT
This is test 6.txt	THFEF5~1.TXT

To see both the long and short file names for each file in the folder, type the following on the command line:

dir /x

Compact Disc File System

Windows 2000 provides support for the ISO 9660–compliant CDFS, which supports long file names as listed in the ISO 9660 Level 2 standards.

When creating a CD-ROM to be used on Windows 2000, you must adhere to the following standards:

- All directory and file names must have fewer than 32 characters.
- All directory and file names must be in capital letters.
- The directory tree cannot exceed 8 levels from the root.
- File name extensions are not mandatory.

Important CDFS adheres to ISO 9660 standards and therefore does not support lowercase file names. When you use CDFS to access a file or directory with a lowercase file name on a CD-ROM, you get a "File not found" error message.

Universal Disk Format

The UDF is new for Windows 2000. UDF is an ISO 13346–compliant, standards-based file system designed for interchanging data on digital video disk (DVD) and CD-ROM. The primary function of UDF is to support read-only DVD-ROM media.

Note Windows 2000 reads only UDF versions 1.02 and 1.50.

Comparing FAT16, FAT32, and NTFS

You can use FAT16, FAT32, NTFS, or a combination of these file systems on a Windows 2000 system. The choice you make depends on such things as:

- How the computer is used.
- The hardware platform.
- The size and number of hard disks.
- Security considerations.

Important It is recommended that you format all Windows 2000 partitions with NTFS except certain multiple-boot configurations. For more information, see "Advantages of NTFS" later in this chapter.

Comparing FAT File Systems

The numerals in the names FAT16 and FAT32 refer to the number of bits required for a file allocation table entry.

- FAT16 uses a 16-bit file allocation table entry (2^{16} allocation units).
- Windows 2000 reserves the first 4 bits of a FAT32 file allocation table entry, which means FAT32 has a maximum of 2^{28} allocation units. However, this number is capped at 32 GB by the Windows 2000 format utilities.

FAT16 vs. FAT32

Table 3.9 provides a comparison of FAT16 and FAT32 cluster sizes according to drive size.

Table 3.9 Cluster Sizes of FAT16 and FAT32

Drive Size	Default FAT16 Cluster Size	Default FAT32 Cluster Size
260 MB–511 MB	8 KB	Not supported
512 MB–1,023 MB	16 KB	4 KB
1,024 MB–2 GB	32 KB	4 KB
2 GB–8 GB	Not supported	4 KB
8 GB–16 GB	Not supported	8 KB
16 GB–32 GB	Not supported	16 KB
> 32 GB	Not supported	32 KB

There are additional differences between FAT32 and FAT16:

- FAT32 allows finer allocation granularity (approximately 4 million allocation units per volume).
- FAT32 allows the root directory to grow (FAT16 holds a maximum of 512 entries, and the limit can be even lower due to the use of long file names in the root folder).

Advantages of FAT16

Advantages of FAT16 are:

- MS-DOS, Windows 95, Windows 98, Windows NT, Windows 2000, and some UNIX operating systems can use it.
- There are many tools available to address problems and recover data.
- If you have a startup failure, you can start the computer with an MS-DOS bootable floppy disk.
- It is efficient, both in speed and storage, on volumes smaller than 256 MB.

Disadvantages of FAT16

Disadvantages of FAT16 are:

- The root folder can manage a maximum of 512 entries. The use of long file names can significantly reduce the number of available entries.
- FAT16 is limited to 65,536 clusters, but because certain clusters are reserved, it has a practical limit of 65,524. Each cluster is fixed in size relative to the logical drive. If both the maximum number of clusters and their maximum size (32 KB) are reached, the largest drive is limited to 4 GB on Windows 2000. To maintain compatibility with MS-DOS, Windows 95, and Windows 98, a FAT16 volume should not be larger than 2 GB.
- The boot sector is not backed up.
- There is no built-in file system security or file compression with FAT16.
- FAT16 can waste file storage space in larger drives as the size of the cluster increases. The space allocated for storing a file is based on the size of the cluster allocation granularity, not the file size. A 10-KB file stored in a 32-KB cluster wastes 22 KB of disk space.

Advantages of FAT32

FAT32 allocates disk space much more efficiently than previous versions of FAT. Depending on the size of your files, there is a potential for tens and even hundreds of megabytes more free disk space on larger hard disk drives. In addition, FAT32 provides the following enhancements:

- The root folder on a FAT32 drive is now an ordinary cluster chain, so it can be located anywhere on the volume. For this reason, FAT32 does not restrict the number of entries in the root folder.
- It uses space more efficiently than FAT16. FAT32 uses smaller clusters (4 KB for drives up to 8 GB), resulting in 10 to 15 percent more efficient use of disk space relative to large FAT16 drives. FAT32 also reduces the resources necessary for the computer to operate.
- FAT32 is more robust than FAT16. FAT32 has the ability to relocate the root directory and use the backup copy of the FAT instead of the default copy. In addition, the boot record on FAT32 drives has been expanded to include a backup of critical data structures. This means that FAT32 volumes are less susceptible to a single point of failure than FAT16 volumes.

Disadvantages of FAT32

Disadvantages of FAT32 include:

- The largest FAT32 volume Windows 2000 can format is limited in size to 32 GB.
- FAT32 volumes are not accessible from any other operating systems other than Windows 95 OSR2 and Windows 98.
- The boot sector is not backed up.
- There is no built-in file system security or compression with FAT32.

Advantages of NTFS

It is recommended that you format all Windows 2000 partitions with NTFS except multiple-boot configurations where non-Windows 2000 and non-Windows NT startups are necessary.

Formatting your Windows 2000 partitions with NTFS instead of FAT allows you to use features that are available only on NTFS. These include:

Recoverability The recoverability designed into NTFS is such that a user should seldom have to run a disk repair program on an NTFS volume. NTFS guarantees the consistency of the volume by using standard transaction logging and recovery techniques. In the event of a system failure, NTFS uses its log file and checkpoint information to automatically restore the consistency of the file system. For more information about recovering your system, restoring data, and creating an emergency repair disk (ERD), see "Repair, Recovery, and Restore" in this book.

Compression Windows 2000 supports compression on an individual file basis for NTFS volumes. Files that are compressed on an NTFS volume can be read and written by any Windows-based application without first being decompressed by another program. Decompression happens automatically during the read of the file. The file is compressed again when it is closed or saved.

In addition, formatting your volumes with NTFS instead of FAT16 or FAT32 provides the following advantages:

- There are some Windows 2000 operating system features that require NTFS.
- Faster access speed. NTFS minimizes the number of disk accesses required to find a file.

- File and folder security. On NTFS volumes, you can set file permissions on files and folders that specify which groups and users have access to them, and what level of access is permitted. NTFS file and folder permissions apply both to users working at the computer where the file is stored and to users accessing the file over the network when the file is in a shared folder. With NTFS you can also set share permissions that operate on shared folders in combination with file and folder permissions.
- Windows 2000 can format volumes up to 2 terabytes in size with NTFS.
- The boot sector is backed up to a sector at the end of the volume.
- NTFS supports a native encryption system called Encrypting File System (EFS), using public-key security to prevent unauthorized access to file contents.
- NTFS functionality can be extended by using reparse points, enabling new features such as volume mount points.
- Disk quotas can be set, limiting the amount of space users can consume on an NTFS volume.

Disadvantages of NTFS

While NTFS is the recommended file system for users of Windows 2000, it is not appropriate in all circumstances. Disadvantages of NTFS include:

- NTFS volumes are not accessible in MS-DOS, Windows 95, and Windows 98. Due to upgrades made to NTFS in Windows 2000, the advanced features of the Windows 2000 implementation of NTFS are not available in Windows NT 4.0 and earlier.
- When very small volumes contain mostly small files, the overhead of managing NTFS may cause a slight performance drop in comparison to FAT.

Formatting the System Partition in Multiple-Boot Configurations

If you want to start another operating system, such as Windows 95, Windows 98, Windows for Workgroups, or MS-DOS, use FAT for your system partition and the boot partitions for the other operating systems. You can use NTFS for your Windows 2000 boot partition and other volumes on the computer, as long as those volumes will not be accessed by an operating system other than Windows 2000.

Which Is Faster – FAT16, FAT32, or NTFS?

For small volumes, FAT16 or FAT32 usually provide faster access to files than NTFS because:

- The FAT structure is simpler.
- The FAT folder size is smaller for an equal number of files.
- FAT has no controls regulating whether a user can access a file or a folder; therefore, the system does not have to check permissions for an individual file or whether a specific user has access to the file or folder. This advantage is minimal because Windows 2000 still has to determine if the file is read-only, or whether the file is on a FAT or NTFS volume.

NTFS minimizes the number of disk accesses and time needed to find a file. In addition, if a folder is small enough to fit in the MFT record, NTFS reads the entire folder when it reads its MFT record.

A FAT folder entry contains an index of the file allocation table, which identifies the cluster number for the first cluster of the folder. To view a file, FAT has to search the folder structure.

In comparing the speed of operations performed on large folders containing both long and short file names, the speed of a FAT operation depends on the operation itself and the size of the folder. If FAT searches for a file that does not exist, it has to search the entire folder—an operation that takes longer on a FAT structure than on the B-tree structure used by NTFS. In mathematical terms, the average time to find a file on a FAT folder is a function of $N/2$, where N is the number of files. On an NTFS folder, the average time is a function of Log N.

Several factors affect the speed with which Windows 2000 reads or writes a file:

- Fragmentation of the file. If a file is badly fragmented, NTFS usually requires fewer disk accesses than FAT to find all of the fragments.
- Cluster size. For both file systems, the default cluster size depends on the volume size, and is always a power of 2. FAT16 addresses are 16 bits, FAT32 addresses are 32 bits, and NTFS addresses are 64 bits.
- The default FAT cluster size is based upon the fact that the file allocation table can have at most 65,535 entries, so the cluster size is a function of the volume size divided by 65,535. Therefore, the default cluster size for a FAT volume is almost always larger than the default cluster size for an NTFS volume of the same size. The larger cluster size for a FAT volume means that there might be less fragmentation in files on a FAT volume.
- Location of small files. With NTFS, small files are entirely contained within the MFT record. The file size that fits in the MFT record depends upon the the number of attributes for the file.

Maximum Volume Sizes

The maximum size of a volume depends on the file system used to format the volume. Windows 2000 allows you to format volumes with three different file systems: NTFS, FAT16, and FAT32.

Windows 2000 has the capability to combine noncontiguous disk areas when creating volume sets and stripe sets, but these volumes have the same maximum size limitations of a single volume.

Maximum Sizes on FAT16 Volumes

FAT16 can support a maximum of 65,535 clusters per volume. Table 3.10 lists FAT16 size limits.

Important For Windows NT and Windows 2000, the cluster size of FAT16 volumes between 2 GB and 4 GB is 64 KB. This cluster size is known to create compatibility issues with some applications. For this reason, it is recommended that FAT32 be used on volumes that are between 2 GB and 4 GB. One of the known compatibility issues involves setup programs that do not compute volume free space properly on a volume with 64 KB clusters and will not run because of a perceived lack of free space. The Format program in Windows 2000 displays a warning and asks for a confirmation before formatting a volume with 64 KB clusters.

Table 3.10 FAT16 Size Limits

Description	Limit
Maximum file size	2^{32} - 1 bytes
Maximum volume size	4 GB
Files per volume	2^{16}

Maximum Sizes on FAT32 Volumes

The FAT32 volume must have at least 65,527 clusters. The maximum number of clusters on a FAT32 volume is 4,177,918. Windows 2000 creates volumes up to 32 GB, but you can use larger volumes created by other operating systems such as Windows 98. Table 3.11 lists FAT32 size limits.

Table 3.11 FAT32 Size Limits

Description	Limit
Maximum file size	2^{32} - 1 bytes
Maximum volume size	32 GB (This is due to the Windows 2000 format utility. The maximum volume size that Windows 98 can create is 127.53 GB).
Files per volume	Approximately 4 million

Important Windows 2000 can format new FAT32 volumes up to 32 GB in size but can mount larger volumes (for example, up to 127.53 GB and 4,177,918 clusters from a volume formatted with the limits of Windows 98). It is possible to mount volumes that exceed these limits, but doing so has not been tested and is not recommended.

Maximum Sizes on NTFS Volumes

In theory, the maximum NTFS volume size is 2^{32} clusters. However, even if there were hardware available to supply a logical volume of that capacity, there are other limitations to the maximum size of a volume.

One of these limitations is partition tables. By industry standards, partition tables are limited to 2^{32} sectors. Sector size, another limitation, is a function of hardware and industry standards, and is typically 512 bytes. While sector sizes might increase in the future, the current size puts a limit on a single volume of 2 terabytes (2^{32} * 512 bytes, or 2^{41} bytes).

For now, 2 terabytes should be considered the practical limit for both physical and logical volumes using NTFS.

The maximum number of files on an NTFS volume is 2^{32} - 1. Table 3.12 lists NTFS size limits.

Table 3.12 NTFS Size Limits

Description	Limit
Maximum file size	2^{64} bytes - 1 KB (On disk format)
	2^{44} bytes - 64 KB (Implementation)
Maximum volume size	2^{64} allocation units (On disk format)
	2^{32} allocation units (Implementation)
Files per volume	2^{32} - 1

Controlling Access to Files and Folders

On NTFS volumes, you can set file permissions on files and folders that specify which groups and users have access to them, and what level of access is permitted. NTFS file and folder permissions apply both to users working at the computer where the file is stored and to users accessing the file over the network when the file is in a shared folder. With NTFS you can also set share permissions, which operate on shared folders in combination with file and folder permissions. File attributes (read-only, hidden, system) also limit file access. Figure 3.5 shows the permissions listed on the **Security** tab of the **Properties** dialog box.

FAT16 and FAT32 allow you to set file attributes on files but they do not provide file permissions.

The version of NTFS included with Windows 2000 offers an important new feature for managing security—inheritable permissions. The **Security** dialog box offers the option to **Allow inheritable permissions from parent to propagate to this file object** which is enabled by default.

This feature significantly reduces the time and I/O work required to change the permissions of many files and subfolders. For example, suppose a user wants to change the permissions on a tree consisting of several thousand files. With Windows NT 4.0, each file and folder needs to be individually changed. However, with Windows 2000, if the subfolders and files inherit permissions, they only need to be set for the top-level folder.

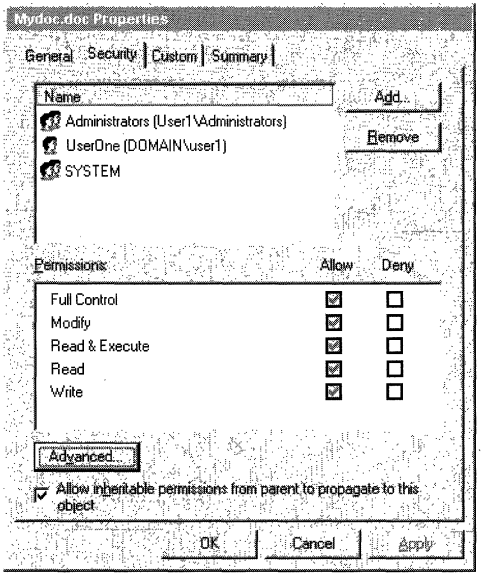


Figure 3.5 Permissions Dialog Box

Figure 3.6 shows the Permissions listed when you select the **Advanced** button on the **Security** tab of the **Properties** dialog box.

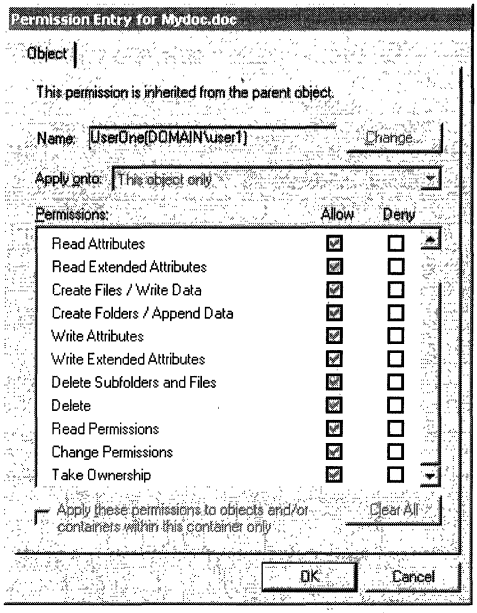


Figure 3.6 Advanced Permissions Dialog Box

Important To preserve permissions when you copy or move files between NTFS folders, use the Robocopy program on the *Microsoft® Windows® 2000 Resource Kit* companion CD.

You can back up and restore data on FAT and NTFS volumes. However, if you back up data from an NTFS volume and then restore it to a FAT volume, you lose security settings and other file information on the restored copies.

You can restore Remote Storage data only to an NTFS volume. For more information about Remote Storage, see "Data and Storage Management" in this book.

Although NTFS provides access controls to individual files and folders, users can perform certain actions on files or folders even if permissions are set on a file or folder to prevent access to users.

For example, you have a folder (Dir1) containing a file (File1), and you grant Full Control to a user for the folder Dir1. If you specify that the user has No Access to File1, the user can still delete File1. This is because the user's Full Control rights in the folder allow the user to delete the contents (files or subfolders) of the folder.

To prevent files from being deleted, you must set permissions on the file itself, and you must set permissions for the folder containing the file.

Anyone who has List, Read, or greater permissions in a folder can view file properties on any file in the folder, even if file permissions prevent them from seeing the contents of the file.

Note In the **Properties** dialog box, you can use the **Security** tab to deny **Full Control** while leaving **Modify**, **Read & Execute**, **Read**, and **Write** in place.

With FAT volumes, you cannot set any permissions on the individual files and folders. The only security available is the share permissions that are set on the entire share, that affect all files and folders on that share, and that only function over the network. Once a folder is shared, you can protect the shared folder by specifying one set of share permissions for all files and subfolders of the shared folder. Share permissions are set in much the same way file and folder permissions are set in NTFS. But because share permissions apply globally to all files and folders in the share, they are significantly less versatile than the file and folder permissions used for NTFS volumes.

Share permissions apply equally to NTFS and FAT volumes. They are enforced by Windows 2000, not the individual file system. However, when you move or copy a file from an NTFS to a FAT volume, permissions and other attributes unique to NTFS are lost.

POSIX Compliance

If you want POSIX compliance, you must use NTFS. POSIX compliance permits UNIX programs to be ported to Windows 2000. Windows 2000 is fully compliant with the Institute of Electrical and Electronic Engineers (IEEE) standard 1003.1, which is a standard for file naming and identification.

The following POSIX-compliant features are included in NTFS:

- **Case-sensitive naming.** For example, POSIX would interpret README.TXT, Readme.txt, and readme.txt as different files.
- **Hard links.** A file can be given more than one name. This allows two different file names, which can be located in different folders, to point to the same data.
- **Additional time stamps.** These show when the file was last accessed or modified.

Caution You must use POSIX-based programs to manage file names that differ only in case. POSIX-based programs allow you to create and manage case-sensitive file names.

You cannot use standard commands to manage file names that differ only in case. (Standard commands include those used at the command prompt, such as **copy**, **del**, and **move**, and their equivalents in My Computer.) For example, if you type **del AnnM.Doc** at the command prompt, both annm.doc and AnnM.Doc would be deleted.

File and Folder Compression

Windows 2000 supports compression on individual files and on folders for NTFS volumes. Files compressed on an NTFS volume can be read and written by any Windows-based application without first being decompressed by another program. Decompression occurs automatically when the file is read. The file is compressed again when it is closed or saved. Compressed files and folders have an attribute of **C** when viewed in My Computer.

Only NTFS can read the compressed form of the data. When an application such as Microsoft Word or an operating system command such as **Copy** requests access to the file, NTFS decompresses the file before making it available. For example, if you copy a compressed file from another Windows 2000–based computer to a compressed folder on your hard disk, the file is decompressed, copied, and recompressed.

This compression algorithm is similar to that used by MS-DOS 6.0 DoubleSpace® and MS-DOS 6.22 DriveSpace®, with one important difference—the MS-DOS functionality compresses the entire primary partition or logical drive, while NTFS enables the user to compress individual files and folders within the NTFS volume.

The compression algorithms in NTFS are designed to support cluster sizes of up to 4 KB. When the cluster size is greater than 4 KB on an NTFS volume, none of the NTFS compression functions are available.

Compressing and Decompressing Folders and Files

Files and folders on an NTFS volume are either compressed or decompressed. The compression state of a folder does not reflect the compression state of the files in that folder. For instance, a folder may be compressed, yet all or some of the files in that folder could be decompressed if they were moved from a compressed folder of if you selectively decompressed some of the files in the folder.

You can set the compression state of folders and compress or decompress files by using My Computer or a command-line program called Compact. When using My Computer, you can set the compression state of an NTFS folder without changing the compression state of existing files in that folder. If you have Read or Write permission, you can change the compression state locally or across a network. You have the option of selecting individual folders or files to compress or decompress.

▶ **To set the compression state of a folder**

1. Start **Windows Explorer**. In the left pane, select the folder you want to compress or decompress.
2. On the **File** menu, click **Properties** to display the **Properties** dialog box.
3. On the **General** tab, click **Advanced**.
4. In the **Advanced Attributes** dialog box, select or clear the **Compress contents to save disk space** check box, and then click **OK**.
5. In the **Properties** dialog box, click **OK**.

Windows 2000 then displays the **Confirm Attribute Changes** dialog box. This dialog box gives you the option of either compressing the folder only, or compressing the folder and its subfolders and files. To keep existing files or subfolders in the NTFS folders in their current compression state, click **Apply changes to this folder only**, and then click **OK**.

▶ **To compress or decompress individual files**

1. Start **Windows Explorer**. In the left pane, select the file you want to compress or decompress.
2. On the **File** menu, click **Properties** to display the **Properties** dialog box.
3. On the **General** tab, click **Advanced**.
4. In the **Advanced Attributes** dialog box, select or clear the **Compress contents to save disk space** check box, and then click **OK**.
5. In the **Properties** dialog box, click **OK.**

Note Windows 2000 allows closed page files to be compressed. However, when you restart Windows 2000, the page files automatically revert to an uncompressed state. For information about page files, see the topics about virtual memory in Windows 2000 Server Help.

You can set My Computer to display alternate colors for compressed files and folders with the following procedure:

▶ **To display alternate colors for compressed files and folders**

1. In **My Computer**, select the **Tools** menu.
2. On the **Tools** menu, click **Folder Options**.
3. On the **View** tab, select or clear the **Display compressed files and folders with alternate color** check box.
4. Click **OK** to return to My Computer.

Using the Compact Program

The Compact program is the command-line version of the compression functionality in My Computer. The **compact** command displays and alters the compression of folders and files on NTFS volumes. It also displays the compression state of folders.

There are two reasons why you might want to use Compact instead of My Computer:

- You can use Compact in a batch script.
- If the system fails during compression or decompression, the file or folder is marked as Compressed or Uncompressed. If the operation did not complete, Compact forces the operation to complete in the background.

Note Unlike My Computer, Compact does not prompt you to compress or uncompress files and subfolders when you set the compression state of a folder; it automatically compresses or decompresses any files that are not already in the compression state that you set for the folder.

For more information about the Compact program, at the command prompt, type:

compact /?

or see "File System Tools" later in this chapter.

Effects of Compression on Moving and Copying Files

Moving and copying files and folders in disk volumes can change their compression state. The compression state of these files and folders, and the file system in which they were created, can impact the way they are affected while being moved or copied. The compression state of an NTFS file or folder is controlled by its compression attribute.

Moving Files or Folders on NTFS Volumes

When you move an uncompressed file or folder to another folder, the file remains uncompressed after the move, regardless of the compression state of the folder it was moved to. For example, if you move an uncompressed file to a compressed folder, the file remains uncompressed after the move, as illustrated in Figure 3.7.

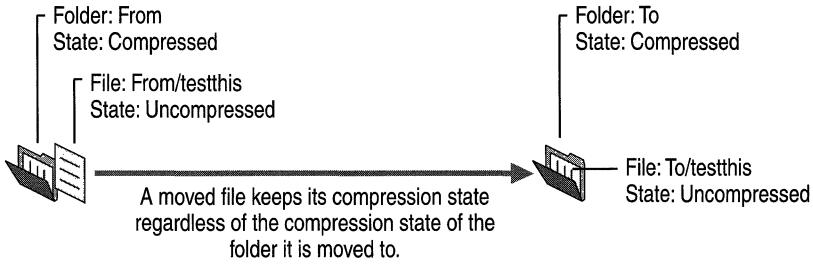


Figure 3.7 Moving an Uncompressed File to a Compressed Folder

When you move a compressed file or folder to another folder, the file remains compressed after the move, regardless of the compression state of the folder it was moved to, as illustrated in Figure 3.8.

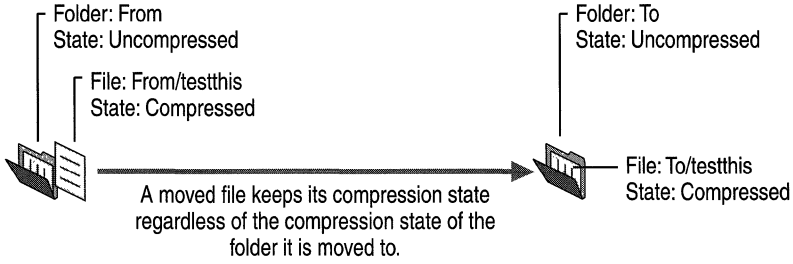


Figure 3.8 Moving a Compressed File to an Uncompressed Folder

Copying Files or Folders on NTFS Volumes

When you copy a file to a folder, the file takes on the compression attribute of the target folder. For example, if you copy a compressed file to an uncompressed folder, the file is automatically uncompressed when it is copied to the folder, as illustrated in Figure 3.9.

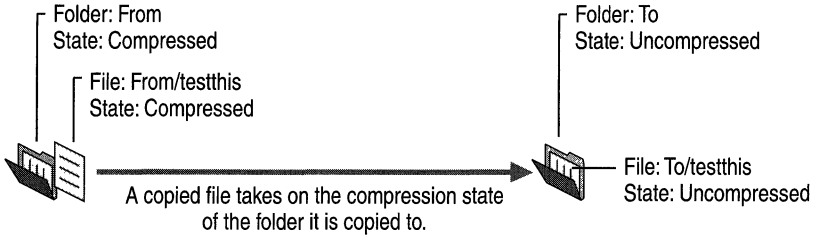


Figure 3.9 Copying a Compressed File to an Uncompressed Folder

When you copy a file to a folder that already contains a file of the same name, the file that is copied takes on the compression attribute of the target file, regardless of the compression state of the folder, as illustrated in Figure 3.10.

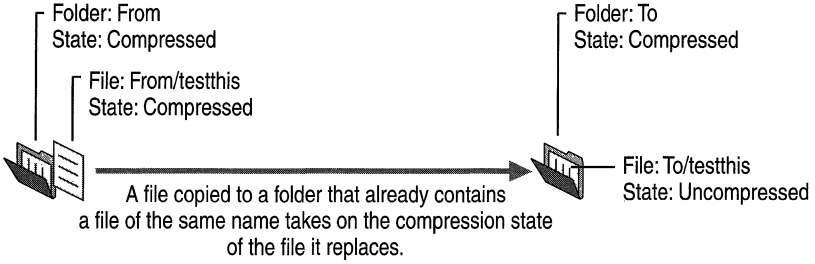


Figure 3.10 Copying a File to a Folder That Already Contains a File of the Same Name

Moving and Copying Files Between FAT16, FAT32, and NTFS Volumes

Like files copied between NTFS folders, files moved or copied from a FAT folder to an NTFS folder always assume the compression attribute of the target folder. Because Windows 2000 supports compression only on NTFS volumes, any compressed NTFS files moved or copied to a FAT volume are automatically decompressed. Similarly, compressed NTFS files copied or moved to a floppy disk are automatically decompressed.

Adding Files to an Almost Full NTFS Volume

When adding files to an NTFS volume that is almost full, you can get error messages that indicate there is not enough disk space to write the entire file if the file cannot be compressed, regardless of the degree of compression in the file when it is opened. For this reason, it is possible to get a read error when you are trying to open a compressed file.

If you copy files to a compressed NTFS folder that does not have enough room for all of the files in their uncompressed state, you will receive a message indicating that there is not enough space on the disk even though the files will all fit when compressed. Because NTFS allocates space based upon the uncompressed size of the file, you can get this error when the uncompressed size of the file exceeds the size of the volume. NTFS does not wait for the compression and writing of one file to complete before it begins work on subsequent files, and the system does not get the unused space back from compression until after the buffer is compressed.

When you are running a program and saving files to a compressed folder on a volume that is almost full, the success of the save depends on factors such as how much the file compresses and whether the beginning of the file compresses well.

If you cannot delete any files or do not have any files that you can compress, you can usually copy all of the files if you first copy the largest or the ones that compress best. You can also try copying them in smaller groups rather than all at once.

NTFS Compression Algorithm

NTFS compression uses a 3-byte minimum search rather than the 2-byte minimum used by DoubleSpace. This search enables much faster compressing and decompressing (roughly two times faster), while sacrificing only two percent compression for the average text file.

Each NTFS data stream contains information that indicates if any part of the stream is compressed. Individual compressed buffers are identified by "holes" following them in the information stored for that stream. If there is a hole, NTFS automatically decompresses the preceding buffer to fill the hole.

NTFS provides real-time access to a compressed file, decompressing the file when it is opened and compressing it when it is closed. When writing a compressed file, the system reserves disk space for the uncompressed size. The system gets back unused space as each individual compression buffer gets compressed.

Note Some programs do not allocate space before beginning a save, and only display an error message when they run out of disk space.

Compression Performance

NTFS compression might cause performance degradation because a compressed NTFS file is decompressed, copied, and then recompressed as a new file, even when copied inside the same computer. Similarly, on network transfers, the file is decompressed, which affects bandwidth as well as speed.

The current implementation of NTFS compression runs more efficiently on Windows 2000 Professional than on Windows 2000 Server. Compression on a computer running Windows 2000 Professional does not seem to produce a substantial performance degradation. Heavily loaded servers with considerable write traffic are poor candidates for data compression, while read-only, read-mostly servers, or lightly loaded servers might not see significant performance degradation.

The two ways to measure the performance of NTFS data compression are size and speed. You can tell how well compression works by comparing the uncompressed and compressed file and folder sizes. For more information about using the DirUse program to see the compressed size of folders see "File System Tools" later in this chapter.

Other Compression Methods

Compression utilities other than NTFS compression are available to compress files on computers running Windows 2000. These utilities differ from NTFS compression in the following ways:

- They usually run only from the command line.
- Files cannot be opened when they are in a compressed state—the file must first be decompressed by using the companion program to the one used to compress the file. When you close the file, it is saved in an uncompressed state, and you must use a program to compress it.

The *Windows 2000 Resource Kit* includes a compress utility, which can only be run from the command line, and two expand utilities: one runs from the command line; the other is a Windows 2000–based program. For more information about these programs, see "File System Tools" later in this chapter.

As mentioned earlier, the DoubleSpace and DriveSpace compression features in MS-DOS cannot be used when running Windows 2000.

NTFS Recoverability

NTFS is a recoverable file system that guarantees the consistency of the volume by using standard transaction logging and recovery techniques. In the event of a disk failure, NTFS restores consistency by running a recovery procedure that accesses information stored in a log file. The NTFS recovery procedure is exact, guaranteeing that the volume is restored to a consistent state. Transaction logging requires a very small amount of overhead.

NTFS ensures the integrity of all NTFS volumes by automatically performing disk recovery operations the first time a program accesses an NTFS volume after the computer is restarted following a failure.

NTFS also uses a technique called cluster remapping to minimize the effects of a bad sector on an NTFS volume. For more information, see "Cluster Remapping" later in this chapter.

Important If either the master boot record (MBR) or boot sector is corrupted, you might not be able to access data on the volume. Recovery from errors with the MBR or the boot sector is discussed in "Repair, Recovery, and Restore" in this book.

Recovering Data with NTFS

NTFS views each I/O operation that modifies a system file on the NTFS volume as a transaction, and manages each one as an integral unit. Once started, the transaction is either completed or, in the event of a disk failure, rolled back (such as when the NTFS volume is returned to the state it was in before the transaction was initiated).

To ensure that a transaction can be completed or rolled back, NTFS records the suboperations of a transaction in a log file before they are written to the disk. When a complete transaction is recorded in the log file, NTFS performs the suboperations of the transaction on the volume cache. After NTFS updates the cache, it commits the transaction by recording in the log file that the entire transaction is complete.

Once a transaction is committed, NTFS ensures that the entire transaction appears on the volume, even if the disk fails. During recovery operations, NTFS redoes each committed transaction found in the log file. Then NTFS locates the transactions in the log file that were not committed at the time of the system failure and undoes each transaction suboperation recorded in the log file. Incomplete modifications to the volume are prohibited.

NTFS uses the Log File service to log all redo and undo information for a transaction. NTFS uses the redo information to repeat the transaction. The undo information enables NTFS to undo transactions that are not complete or that have an error.

Important NTFS uses transaction logging and recovery to guarantee that the volume structure is not corrupted. For this reason, all system files remain accessible after a system failure. However, user data can be lost because of a system failure or a bad sector.

Caching and Data Recovery

The cache is the area of RAM that contains data. When you write data to disk, the lazy-write technique in Windows 2000 indicates that the data is written when, in fact, it is still in the cache. There can also be cache memory on the disk controller or on the disk itself. The following information will help you decide whether you want to enable the disk or controller cache:

- Turning on write caching improves disk performance, particularly if the disk is being heavily written to.
- Control of the write-back cache is a firmware function provided by the disk manufacturer. See the documentation supplied with the disk or disk controller. You cannot configure the write-back cache from Windows 2000.
- Write caching does not impact the reliability of the file system's own metadata. NTFS instructs the disk device driver to ensure that metadata writes get written regardless of whether write caching is enabled. Non-metadata is written to the disk normally, so such data can be cached.
- Read caching in the disk has no impact on file system reliability.

Cluster Remapping

In the event of a bad-sector error, NTFS implements a recovery technique called cluster remapping. When Windows 2000 detects a bad-sector, NTFS dynamically remaps the cluster containing the bad sector and allocates a new cluster for the data. If the error occurred during a read, NTFS returns a read error to the calling program, and the data is lost. If the error occurs during a write, NTFS writes the data to the new cluster, and no data is lost.

NTFS puts the address of the cluster containing the bad sector in its bad cluster file so the bad sector is not reused.

Important Cluster remapping is *not* a backup alternative. Once errors are detected, the disk should be monitored closely and replaced if the defect list grows. This type of error is displayed in the Event Log.

Features Built on Reparse Points

A reparse point is a file or a directory that has user-controlled data stored in the system-administered reparse attribute. The reparse attribute is used by file system filters to enhance the normal behavior of files or directories present in the underlying file system. Thus, a file or a directory that contains a reparse point acquires additional behavior not present in the underlying file system.

For more information about sparse files, see the Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Remote Storage

Remote Storage uses reparse points to mark files that have some of their state stored remotely. The user accesses a file that is in Remote Storage through a placeholder stored on the local computer. The corresponding file system filter uses the information stored in the reparse point to access the data from the storage location. The reparse point contains the allocation information necessary to identify and retrieve the state of the file that is stored remotely and is needed by NTFS to retrieve the file. For more information about Remote Storage, see "Data Storage and Management" in this book.

Volume Mount Points

Volume mount points are new system objects in the internal namespace of Windows 2000 that represent storage volumes in a persistent, robust manner. This feature allows multiple disk volumes to be linked into a single tree, similar to the way Dfs links remote network shares. You can have many disk volumes linked together, with only a single drive letter pointing to the root volume. The combination of an NTFS junction and a Windows 2000 volume mount point can be used to graft multiple volumes into the namespace of a host NTFS volume. Windows 2000 offers this new mounting feature as an alternative to drive letters so system administrators can transcend the 26-drive letter limit that exists in Windows NT. Volume mount points are robust against system changes that occur when devices are added or removed from a computer.

Important A volume is a self-contained unit of storage administered by a file system. The file system that administers the storage in a volume defines a namespace for the volume. A volume mount point is a directory name in an NTFS file system that denotes the root of an arbitrary volume. A volume mount point can be placed in any empty directory of the namespace of the containing NTFS volume. Because volumes can be denoted by arbitrary directory names, they are not required to have a traditional drive letter.

Placing a volume mount point on an NTFS directory causes the storage subsystem to resolve the directory to a specified local volume. This "mounting" is done transparently and does not require a drive letter to represent the volume. A Windows 2000 mount point always resolves to the root directory of the desired volume. Volume mount points require that the version of NTFS included with Windows 2000 be used because they are based on NTFS reparse points.

File System Tools

The utilities described in this section are available on either the Windows 2000 Setup CD or the *Windows 2000 Resource Kit* companion CD. Table 3.13 shows where to find the utilities.

Table 3.13 Location of File System Utilities

Tool	Location
Cacls	Windows 2000 Setup CD
Compact	Windows 2000 Setup CD
Compress	*Windows 2000 Resource Kit* companion CD
Convert	Windows 2000 Setup CD
DirUse	*Windows 2000 Resource Kit* companion CD
Expand	Windows 2000 Setup CD
Mountvol	Windows 2000 Setup CD

Cacls: Displays and Modifies NTFS Access Control Lists

You can use Cacls to display or modify access control lists (ACLs) of files or folders. Table 3.14 follows the command format and describes the command options. The format of the command is:

CACLS *filename* | *folder* [**/t**] [**/e**] [**/c**] [**/g** *user*:*perm*] [**/r** *user* [...]]

[**/p** *user*:*perm* [...]] [**/d** *user* [...]]

Table 3.14 Cacls Options

Option	Description
file name or *folder name*	Displays ACLs.
/t	Changes ACLs of specified files in the current folder and all subfolders.
/e	Edits an ACL instead of replacing it.
/c	Continues on access-denied errors.
/g *user*:*perm*	Grants a specified user access rights, where *perm* can be: R (Read) C (Change (write)) F (Full Control)
/r *user*	Revokes a specified user's access rights (only valid with **/e**).
/p *user*:*perm*	Replaces a specified user's access rights, where *perm* can be: N (None) R (Read) C (Change (write)) F (Full Control)
/d *user*	Denies access to a specified user.

Wildcard characters can be used to specify more than one file in a command. You can also specify more than one user in a command.

If you already have permissions set for multiple users on a folder or file and do not use the **/e** option, all user permissions are removed except for the user and permissions specified on the command line. Use the following syntax when modifying user permissions to include read, change, and full control:

cacls *filename* | *folder* **/e /r** *username*

cacls *filename* | *folder* **/e /g** *username*:*permission*

cacls *filename* | *folder* **/e /p** *username*:*permission*

The Cacls tool does not provide a **/y** option that answers automatically with **Y** to the **ARE YOU SURE? Y/N** prompt. However, you can use the **echo** command to pipe the character **Y** as input to the prompt when you are running Cacls in a batch file. Use the following syntax to automatically answer **Y**:

echo y| cacls *filename* | *folder* **/g** *username*:*permission*

Important Do not enter a space between the **Y** and the pipe symbol (|). If you do, Cacls will not make the permission change.

Compact: Compresses and Decompresses NTFS Files and Folders

Compact is the command-line version of the compression functionality in My Computer. Compact displays and alters the compression of folders and files on NTFS volumes. It also displays the compression state of folders. For more information about this program, at the command prompt type:

compact /?

Table 3.15 describes the options available with Compact. The syntax of the command is:

compact [**/c**] [**/u**] [**/s**[**:***folder*]] [**/a**] [**/i**] [**/f**] [**/q**] [*filename* [...]]

Table 3.15 Compact Options

Option	Description
none	Displays the compression state of the current folder.
/c	Compresses the specified folder or file.
/u	Decompresses the specified folder or file.
/s[**:***folder*]	Specifies that the requested action (compress or decompress) be applied to all subfolders of the specified folder, or to the current folder if none is specified.
/i	Ignores errors.
/f	Forces a specified folder or file to compress or decompress.
/a	Displays files with the hidden or system attribute.
/q	Reports only the most essential information.
filename	Specifies a pattern, file, or folder. You can use multiple file names and wildcard characters.

The following are reasons to use this utility rather than My Computer:

- You can use Compact in a batch script. Using the **/i** option enables you to skip files that cannot be opened when you are running in batch mode, such as files already in use by another program.
- If the system failed during compression or decompression, the file or folder is marked as Compressed or Uncompressed, even if the operation did not complete. You can force the operation to complete by using Compact with the **/f** option (with either the **/c** or **/u** option).

Note Compact automatically compresses or decompresses all of the files and subfolders when you change the compression state of a folder. It does not ask whether you want to change the compression state of the files or subfolders in it.

Volume Compression Requirements

When you attempt to compress a volume that is very low on free space, you might receive an error indicating that there was insufficient space to perform the action.

These errors indicate that the system needs additional free space to perform a compression. The system is not designed to manipulate the data in place on the disk. Additional space is needed to buffer the user data and to hold additional file system metadata. The amount of additional free space required depends on the cluster size, file size, and available space.

Compress: Compresses Files or Folders

Compress is a command-line utility that can be used to compress one or more files. You cannot open a file that has been compressed using this utility until you have expanded it with Expand. To use this program, at the command line type:

compress

with the appropriate options. Table 3.16 describes the options available with Compress. The syntax of the command is:

compress [**–r**] [**–d**] *source* [*destination*]

Table 3.16 Compress Options

Option	Description
–r	Renames compressed files.
–d	Updates compressed files only if out of date.
source	Specifies the source file. The asterisk (*) and question mark (?) wildcard characters can be used.
destination	Specifies the destination file or path. The destination can be a folder. If *source* specifies multiple files and the **–r** option is not specified, then *destination* must be a folder.

Note Do not use Compress to compress files or folders on NTFS volumes. Instead, compress NTFS files and folders with Compact or by setting or clearing the **Compressed** attribute in My Computer. For information about using My Computer, see "Compact: Compresses and Decompresses NTFS Files and Folders" earlier in this chapter.

Convert: Converts a Volume from FAT to NTFS

You can use Convert to convert a volume from FAT to NTFS. This utility performs the conversion within the existing volume. You do not need to back up and restore the files when you use this program.

You cannot convert the Windows 2000 boot partition while you are running Windows 2000, so Convert allows you to convert the partition the next time you start Windows 2000. When you convert the partition this way, Windows 2000 restarts twice to complete the conversion process. The syntax of the command is:

convert *drive*: **/FS:NTFS** [**/v**]

To use this utility, at the command line type:

convert

with the appropriate options. Table 3.17 describes the options available with Convert.

Table 3.17 Convert Options

Option	Description
drive	Logical drive that you want to convert.
/FS	Specifies that you want to convert to NTFS.
/v	Runs the tool in verbose mode.

Important Volumes that are converted from FAT to NTFS, (rather than initially formatted with NTFS) lack some performance benefits. Fragmentation of the MFT might occur and on boot partitions, NTFS permissions are not applied after the partition is converted.

Free Space Required to Convert FAT to NTFS

The conversion of a disk partition from FAT to NTFS requires a sufficient amount of available free disk space in order to build the NTFS disk structures. For information about the process Convert uses to convert FAT to NTFS and the space required for conversion, see the Knowledge Base link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Converting NTFS and FAT Volumes

FAT and NTFS use very different on-disk structures to represent the allocation of space for files. These structures are often referred to as metadata or file system overhead. Another kind of overhead associated with FAT and NTFS is related to the fact that both file systems allocate disk space in clusters of a fixed size. The exact size of these allocation units or clusters is determined at format time, and the defaults are dependent on the size of the volume.

Like FAT, NTFS has a certain amount of fixed-size overhead and a certain amount of per-file overhead. To support the advanced features of NTFS, such as recoverability, security, and support for very large volumes, the NTFS metadata overhead is somewhat larger than the FAT metadata overhead. However, because NTFS cluster overhead is smaller than FAT cluster overhead, it is often possible to store as much if not more data on an NTFS volume as on a FAT volume, even without using NTFS file compression.

Convert builds the NTFS metadata using space that is considered free space by FAT. Thus, if the conversion fails to complete, the FAT representation of the user files is still valid.

DirUse: Scans a Directory and Reports On Disk Space Usage

You can run DirUse to determine the actual usage of space for compressed files and folders in NTFS volumes. The syntax of the command is:

diruse [**/s** | **/v**] [**/q:***#*] [**/m** | **/k** | **/b**] [**/a**] [**/l**] [**/d**] [**/o**] [**/c**] [**/,**] [**/***] [*dirs*]

The important option for compressed folders and files is **/c**, which causes the display of compressed file or folder size instead of apparent size. For example, if your D drive is an NTFS volume, type **diruse /s /m /c d:** at the command prompt to get the disk space actually used (in megabytes) and the number of files in each of the folders. To see compression information for an individual file, open My Computer or Windows Explorer, select the file, and, on the **File** menu, select **Properties**.

For more information about DirUse, see the *Windows 2000 Resource Kit* Tools Help, or at the command prompt type:

diruse /?

Expand: Expands Compressed Files

The MS-DOS-based Expand utility runs from the command line. Type the **expand** command with the appropriate options as shown in Table 3.18. The syntax of the command is:

expand [**–r**] *source* [*destination*]

Table 3.18 Expand Options

Option	Description
–r	Renames expanded files.
source	Specifies the source file. The asterisk (*) and question mark (?) wildcard characters can be used.
destination	Specifies the destination file or path. The destination can be a folder. If *source* specifies multiple files and the **–r** option is not specified, then *destination* must be a folder.

Mountvol: Displays, Creates, and Deletes Volume Mount Points

Mountvol is a utility that enumerates the volumes in your system. Table 3.19 describes the operations Mountvol can perform on a volume mount point.

Table 3.19 Volume Mount Point Operations

Option	Description
Mountvol or Mountvol /?	Displays the name, globally unique identifier (GUID), and location of the volume.
Mountvol [drive:]path VolumeName	Creates a new volume mount point. Specify either a drive letter root directory or an existing empty NTFS directory as the source of the mount point and a volume name as the target.
Mountvol [drive:]path /D	Deletes an existing volume mount point.
Mountvol [drive:]path /L	Lists a volume name for a given volume mount point.

- 'Path' specifies the existing NTFS directory where the mount point will reside.
- 'VolumeName' specifies the name of the volume that is the mount point target.
- '/D' removes the volume mount point from the specified directory.
- '/L' Lists the mounted volume name for the specified directory.

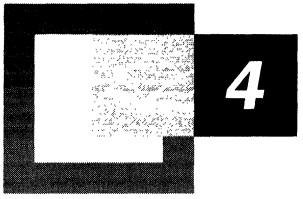
Important A volume can have only one drive letter. Using Mountvol to assign a drive letter fails if the volume already has a drive letter. To avoid this problem, delete the drive letter of the volume before assigning one using Mountvol.

Additional Resources

For more information about transaction logging and NTFS recovery operations, see *Inside Windows NT* by David Solomon (Microsoft Press 1998, ISBN 1572316772).

CHAPTER 4

Network Printing



By using a computer that is running Microsoft® Windows® 2000 Server as your network print server, you can print from any operating systems that your networked computers use. Computers that are running Microsoft® Windows® 2000 Professional can print easily, even across the Internet. Computers that are running Microsoft® Windows NT® version 4.0 and earlier, Microsoft® Windows® 95, and Microsoft® Windows® 98, do not require manual installation of a printer driver, because print resources are provided automatically from each application. Computers that are running UNIX, NetWare, and Macintosh operating systems can submit print jobs using Print Services for Unix, File and Print Services for NetWare, and Print Server for Macintosh, respectively.

Note This chapter concerns itself primarily with printing across a network and regards Windows 2000 Server as a dedicated print server. For general information about printing, such as creating print jobs and sending them to a printer, see the *Microsoft® Windows® 2000 Professional Resource Kit.*

In This Chapter

Introduction to Network Printing 161
Architecture 166
Sharing Printers 178
Working with Other Operating Systems 191
Printing and Active Directory 204
Printing and Clusters 210
Monitoring Printer Performance 213

Related Information in the Resource Kit

- For information about using a computer that is running Windows 2000 Professional as a peer-to-peer print server for a small LAN (10 or fewer connections), see "Connecting to Microsoft Peer-to-Peer Networks" in the *Microsoft® Windows® 2000 Professional Resource Kit.*
- For more information about using printing resources with Active Directory, see the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

Introduction to Network Printing

Windows 2000 Server is designed for network printing. Using a variety of platforms, applications send print jobs to printers that are attached to a Windows 2000 print server or are connected to the network by internal network adapters, external network adapters, or another server.

Print Devices, Print Clients, Printers, and Print Servers

A *print device* is a hardware device used for printing. Print device resolution is measured in dots per inch (DPI). The higher the DPI, the finer the resolution.

A print client is an application on a user's computer that submits print jobs to a print device. *Print jobs* are source code that contain both data and commands for print processing. Print jobs are classified into data types based on the modifications that the print server must make to the job to print correctly.

The *printer* is the software interface between a print device and the print clients. It is sometimes called a *logical printer* and typically resides on the print server for remote printing. For local printing or print jobs redirected to a remote print device, the printer software is located on the print client.

The printer software specifies the print device's interface with the operating system and is stored with a unique printer name in the Printers folder. It includes the printer driver. *Printer drivers* are programs that enable applications to communicate properly with specific print devices.

The printer software specifies the port by which a document reaches the print device. A printer port is the software interface through which a computer communicates with the print device by means of a locally attached interface, such as LPT, COM, or universal serial bus (USB) or by means of a network-attached device such as HP JetDirect, Intel NetPort, or Extended Systems Print Server using a network transport protocol like TCP/IP or Data Link Control (DLC).

Multiple printers can be defined for a single print device, allowing clients to use it in different ways. One user might want the device to be widely available; another might want to restrict it; one might want double-sided printing; another might want single-sided printing only. Conversely, as shown in Figure 4.1, a single set of printer software can access multiple print devices. This is called printer pooling.

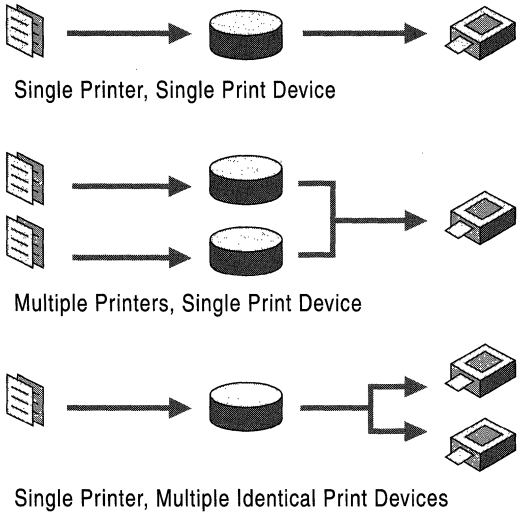


Figure 4.1 Printer Software-Hardware Relationships

A *print server* is a computer that manages a queue of print jobs for a shared print device. Computers running Windows 2000 Server or Windows 2000 Professional can function as a print server. However, Windows 2000 Professional limits concurrent connections to 10. If more than 10 clients need to connect to the print server simultaneously, Windows 2000 Server is the better solution.

A print server that is running on Windows 2000 can process remote printing over various network protocols, such as NWLink, TCP/IP, or AppleTalk. The Transmission Control Protocol/Internet Protocol (TCP/IP) is required for Internet communications and is installed with Windows 2000 Server.

The preferred port monitor in Windows 2000 is the Standard TCP/IP Port Monitor (SPM), which uses TCP/IP as the transport protocol. SNMP is used to configure and monitor the printer ports. In addition to SPM, Internet printing adds an hypertext transport protocol (HTTP) port monitor. All other port monitors that were included with Windows NT 4.0 are also present.

Local and Remote Printing

Several combinations of clients, servers, and printers are possible with Windows 2000, depending on whether the print device is *local* (receives data directly from the computer) or *remote* (accessed through a print server), and whether it is networked or directly attached to the computer. A directly attached print device is connected to the computer by a local interface, such as a parallel, serial RS-232/422/IRDA, or USB port. A networked printer is a node on the network: computers send print jobs to it through a network adapter, which might be built-in. A networked printer is also called a network-interface printer.

The following figures show four basic printing configurations. The thin lines represent physical connections, such as network or parallel cables, and the arrows represent the logical print data flow.

Figure 4.2 shows the simplest configuration, a local or stand-alone printer. The print device is plugged into the parallel port of the computer that runs the application. The printer driver and job queue are on that computer, and it sends print data directly to the print device.

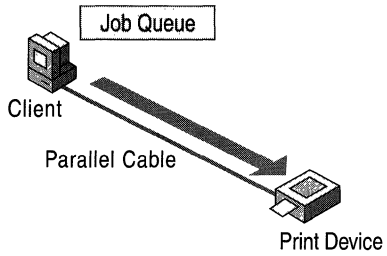


Figure 4.2 Local, Directly Attached Print Device

Figure 4.3 shows a small group of computers sharing a networked print device. This is a peer-to-peer network, where each computer has equal access to the printer, and there is no central control of printing or security. Each computer has its own job queue and cannot see the documents queued on the print device by other computers. If printing halts, the error message does not appear on every client. This is acceptable for small organizations where the users are in frequent contact, but becomes less manageable as traffic increases. Contention among computers submitting documents might cause the print device to time out or reject print jobs.

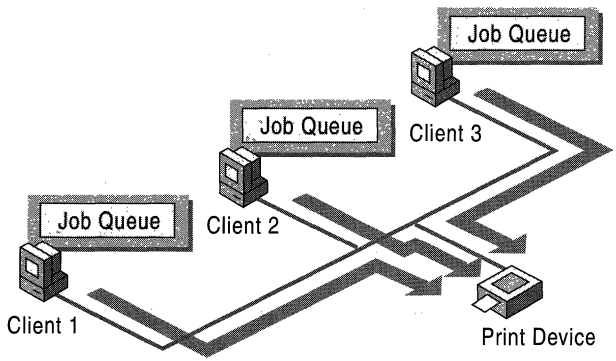


Figure 4.3 Local, Networked Print Device

Figure 4.4 illustrates a network configuration using a central print server. Many clients share access to the print device through the server, which is locally connected to the print device. The job queue resides on the server and is visible to each client.

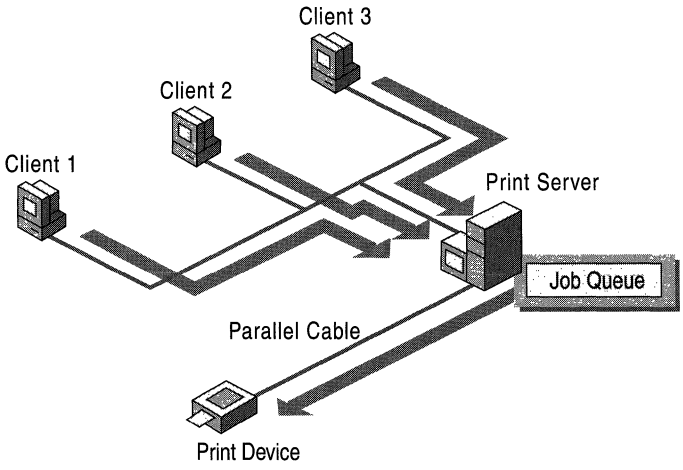


Figure 4.4 Remote, Directly Attached Print Device

Printing is controlled by the server administrator. The administrator defines and enforces a security plan for the network, maintains the printer software, and downloads it to clients when they connect to the printer share. When a client connects to a network print queue, the client checks for new print drivers on the print server and updates older print drivers on the client.

Clients might also be connected to other print devices, and the print server often has several attached print devices. However, the number of parallel ports on the print server limits the number of print devices that can be directly attached to it.

Figure 4.5 shows several clients sharing a print device in a domain managed by a computer that is running Windows 2000 Server; the print device is connected to the server over the network, allowing one print server to manage several print devices.

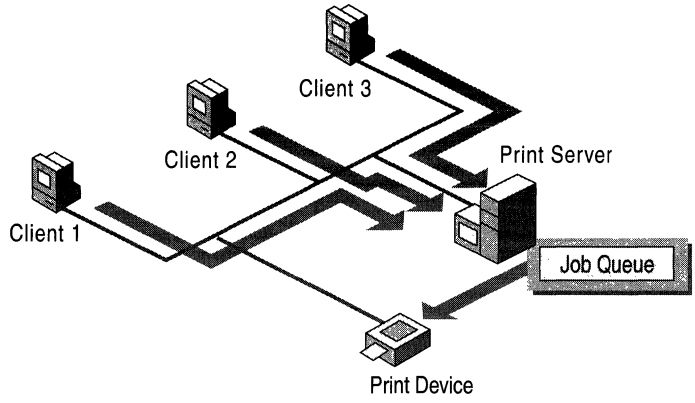


Figure 4.5 Remote, Networked Print Device

Use the Add Printer Wizard to create and share printers so that network clients can connect to them. Regardless of where the print devices are located, the printer software must be located on the print server:

- If the print device is attached locally, the wizard detects it and then attempts to configure the printer software.
- If the print device is attached elsewhere on the network, you must create a port for it when you configure the printer software.

You can also use the Add Printer Wizard to connect to remote print devices. It is important to remember the following:

- *Creating a printer* means installing the print device either directly on a print server or on the network itself, and then configuring the printer software that controls the print device on the print server. Run the Add Printer Wizard and click the **Local computer** option. You must name the printer, install the printer driver, and specify a port.
- C*onnecting to a printer* means connecting to the share on the computer that created the printer. To connect to a printer, run the Add Printer Wizard, and click the **Network printer** option. If the print driver for the client platform exists on the print server, it is not necessary to install the printer driver because Windows 2000 downloads it automatically. Otherwise, you will be prompted to install the printer software.

Web Printing

Windows 2000 Server printing is now integrated with the Internet. To install a printer from the Internet, use the printer's Uniform Resource Locator (URL) as the name of the printer. You can also choose to use the URL format within an intranet. For Windows 2000 Server to process print jobs that contain URLs, it must be running Microsoft® Internet Information Services (IIS).

You can view and manage printers from any browser, but you must use Microsoft Internet Explorer version 4.0 or later to be able to connect to a printer using a browser.

For more information about printing over the Internet, see "Internet Printing" later in this chapter.

Architecture

Figure 4.6 shows the processing flow of a document submitted for printing. Some processes, or the software components that perform them, are slightly different for print clients that are not running Windows 2000. For more information about different print clients, see "Working with Other Operating Systems" later in this chapter.

For more information about how Windows 2000 processes print jobs using the Internet or a corporate intranet, see "Internet Printing" later in this chapter.

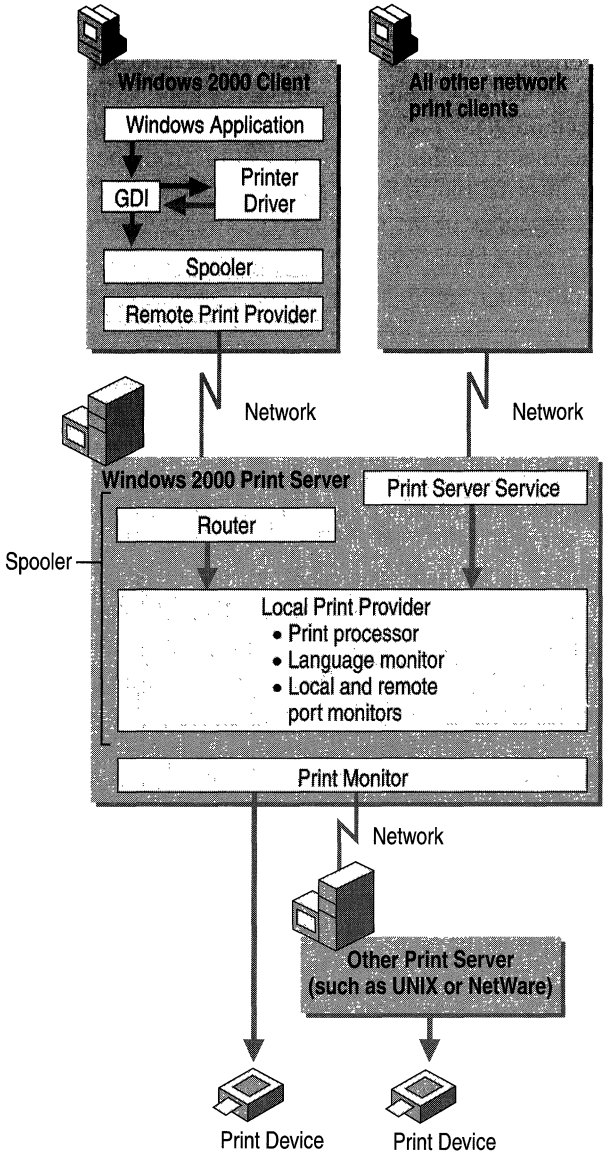


Figure 4.6 Windows 2000 Remote Printing Process

1. A user at a client chooses to print a document.

   If the client is running any Windows-based operating system and is printing from a Windows-based application, the application calls the graphical device interface (GDI), which calls the printer driver associated with the target print device. Using the document information from the application and the print device information from the printer driver, the GDI renders the print job in the printer language of the print device. The application then calls the client side of the spooler (Winspool.drv).

   If the client is running a non-Windows-based operating system or is using a non-Windows-based application on a Windows operating system, another software component replaces the GDI to perform a similar task.

2. The client delivers the print job to the print server.

   If the client is running Windows 2000, the client side of the spooler makes a remote procedure call (RPC) to the server side of the spooler (Spoolsv.exe), which makes a direct application programming interface (API) call to the print router (Spoolss.dll). The print router polls the remote print providers, and the remote print provider (Win32spl.dll) makes an RPC to Spoolsv.exe on the print server, which receives the print job over the network.

   If the client is running another Windows-based operating system or is a Windows 2000 client that created a local printer port and redirected output to a network server (\\Server\Printer), the print job goes to the server message block (SMB) redirector on the client. The redirector delivers the job to the print server service on the print server.

   UNIX or other line printer remote (LPR) clients can print to the Windows 2000 line printer daemon (LPD) service. LPR clients must comply with Request for Comments (RFC) 1179.

   Macintosh clients can print to the Windows 2000 server in two ways:

   - Over AppleTalk if Windows 2000 Server is running Print Server for Macintosh
   - Over TCP/IP if the Macintosh client has an LPR client and Windows 2000 LPD service is running

3. The print router or print server service receives the print job.

   On the print server, print jobs from Windows 2000 clients using Windows applications are enhanced metafiles (EMFs). Some print server services for clients that are not running Windows 2000 assign a data type and others leave the data type blank. Print jobs with no data type assume that the default data type in the **Print Processor** dialog box on the print server. For more information about printer server services, see "Print Processor" later in this chapter.

4. The print router or print server service passes the print job to the local print provider on the server (a component of the spooler), which spools the print job (writes it to disk) in an SPL file.

5. The local print provider polls the print processors. When a print processor recognizes the job's data type, that print processor receives the print job and alters it (or not) according to its data type to ensure that the job prints correctly.
6. The job is despooled to the print monitor.

   If the print device is bidirectional, the job first goes to a language monitor, which handles bidirectional communication with the printer and then passes the job to the port monitor.

   If the print device is not bidirectional, the job goes directly to the port monitor, which transmits the print job to the print device or to another server over a specific communications channel.
7. The print device receives the print job and reproduces the data in the form of the physical medium.

Note For a list of currently supported printers, see the Microsoft Windows 2000 Hardware Compatibility List (HCL) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

Graphical Device Interface

The GDI calls the printer driver and provides information about the type of printer needed and the data type used. The printer driver responds by sending the GDI the fully qualified path name for the printer and printer escape sequences or Printer Job Language (PJL) which the GDI passes to the spooler.

Windows 3.*x*-based (16-bit) applications running on Windows 2000 use the Win16-on-Win32 (WOW) layer, which interacts with the applications through the GDI and with printer drivers through the device driver interface (DDI). The WOW layer translates 16-bit print and display APIs to 32-bit Microsoft® Win32® services. The GDI also provides services to the printer driver, including caching, client/server communications, and ANSI-to-Unicode conversion.

Printer Driver

A *printer driver* is a software program that understands how to communicate with printers and plotters. Printer drivers translate the information a user sends from the computer into commands that the printer understands. Various drivers must be installed on the print server to support different hardware and operating systems. For example, an administrator running Windows 2000 Server who shares a printer with clients running Windows 95 and Windows 98 might want to install the appropriate drivers so the users won't be prompted to install the missing drivers.

The printer driver sends the printer-setting information, including the specifications needed to produce each character of the document, to the GDI. It also transmits helper services or utilities required to make the output print correctly.

Windows 2000 provides three generic printer drivers: Universal (Unidriver), PostScript, and HP-GL/2 plotter.

Universal Printer Driver (Unidriver)

The Unidriver is also called the raster driver because it supports raster graphics printing and is compatible with most types of printers. Each printer vendor provides a device-specific data file.

This driver supports color printing at various depths (4 bits per pixel (bpp), 8 bpp, 24 bpp), scalable TrueType and OpenType fonts, device fonts (including double-byte), grayscale printing, font substitution, run length encoding (RLE), Tag Image File Format (TIFF) version 4.0, and Delta Row Compression (DRC). It also has an extension interface that allows printer manufacturers to customize the driver for specific models.

The Unidriver contains the following component files:

- Unidrv.dll is the printer graphics driver file for printer languages based on raster (bitmap) images, including Printer Control Language (PCL), and most inkjet and dot-matrix printer languages.
- Unidrvui.dll is the configuration file. It displays the user interface for Unidrv.dll.
- Raster minidriver (x.gpd) is the data file, also called characterization file. The file name depends on the printer or printer family. For more information, see the documentation included with your printer.

PostScript Printer Driver

The Windows 2000 PostScript driver uses Adobe version 4.3–compatible PostScript printer description (PPD) files. This driver automatically supports key features, including binary transfer compression, resolution, and paper source, for printers reporting as PostScript levels 1, 2, and 3. Its output is device independent because it is fully compliant with Document Structuring Convention (DSC) 3.0.

This driver contains the following component files:

- Pscript5.dll is the driver file.
- Ps5ui.dll is the configuration file. It displays the user interface for the driver file, reports device capabilities to applications, and handles DevMode settings. Ps5ui.dll also allows you to enable TrueType and OpenType font substitution and image color matching, and to determine whether color matching needs to be done on the host or on the printer.

- X.ppd is the data file, also called the characterization file. PPD files are the only printer driver files that are generally binary-compatible across processors and platforms.

HP-GL/2 Plotter Driver

The Windows 2000 plotter driver supports diverse plotters that use the HP-GL/2 language but does not support HP-GL. This driver's output requires a plotting device that can process all of the enhancements built into the HP-GL/2 language.

This driver contains the following component files:

- Plotter.dll is the driver file.
- Plotui.dll is the configuration file. It displays the user interface for Plotter.dll.
- X.pcd is the data file.

If Microsoft does not supply a driver for your printer...

... ask the device manufacturer for compatibility settings. To ensure quality, use drivers that have passed Windows Hardware Quality Labs (WHQL) tests. Microsoft Product Support Services does not support systems with unapproved drivers. For information, see the WHQL link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.

You can also e-mail ntwish@microsoft.com to suggest support for a driver or feature. Please include the following information in your request: requester's name, business name, requester's contact information (phone number or e-mail address), printer manufacturer, printer model, and nature of request (feature request or driver request). This does not guarantee that Microsoft will write a driver for your printer.

Client Spooler

The spooler on the client side is not used for network printing. It functions the same as the spooler on the server side when printing locally.

Remote Print Provider

A Windows 2000-based client has three remote print providers. When the server receives a print job from a client, the print router polls the remote print providers and passes the job to the first one that recognizes the printer name. On a Windows-based network, this provider is Win32spl.dll. Win32spl.dll sends jobs to print servers, such as those running Microsoft® Windows NT or Microsoft® Windows for Workgroups.

Win32spl.dll performs specific processing based on the type of target print server:

- For a Windows 2000 print server, Win32spl.dll makes an RPC connection to the spooler on the server (Spoolsv.exe) which makes a call to the print router (Spoolss.dll). The print router receives the job over the network and passes it to the local print provider as if a local client had submitted it.
- For a print server with Windows for Workgroups or OS/2, Win32spl.dll sends a message to the Windows 2000 redirector, which then forwards the job over the network to the server.

Print Router

To submit a document for printing, the application communicates with the client side of the spooler (Winspool.drv). Winspool.drv makes an RPC connection to the server side of the spooler (Spoolsv.exe), which makes a direct API call to the print router (Spoolss.dll), which is also on the server side of the spooler. The print router passes the print job to the local print provider.

Print Server Service

Remote print jobs are intercepted by a print server service before passing to the server spooler. Different services are provided for different clients. A Microsoft® MS-DOS® client running Microsoft Client for Networks, for instance, cannot use the same service as a UNIX client.

The service determines whether the spooler alters the document and how to alter it, and assigns the corresponding data type value to the job. Alternatively, the service can omit the data type and let the print processor component of the spooler apply the default value. Each service supplied with Windows 2000 uses different logic to determine how the document prints. The print server service assigns the RAW (ready to print) data type to all print jobs from clients that are not running Windows 2000 or Windows NT. Print server services, such as Print Services for Unix and Print Server for Macintosh, can also assign the RAW data type to incoming jobs, based on information from the client about the type of printer.

Note This component is used only if the print client has added the printer as a local printer and redirected to a network printer. If the print client has connected to the printer, the job goes directly to the spooler.

The default print server service for Windows 2000 is Windows 2000 Server Service, or Srv.sys. The service receives jobs from print clients that use SMB redirectors. Srv.sys does not set the data type. For more information about the data types used with Windows clients, see "Print Processor" later in this chapter.

Other print server services available with Windows 2000 are listed in Table 4.1.

Table 4.1 Special Print Server Services

Name of Service	Purpose	Availability
Print Services for Unix	Nearly all remote printing using the TCP/IP protocol.	Installed with Windows 2000.
Peer Web Services	Printing to the Internet from Windows 2000 Professional.	*Microsoft® Windows® 2000 Professional Resource Kit* companion CD.
Internet Information Services (IIS)	Printing to and from the Internet using Windows 2000 Server.	Installed with Windows 2000 Server.
File and Print Services for NetWare	Printing from NetWare computers using Windows 2000 Server.	Sold separately as Microsoft Services for NetWare.
Client Service for NetWare	Printing to NetWare servers from Windows 2000 Professional.	Optional component of Windows 2000 Professional.
Gateway Service for NetWare	Printing to and from NetWare print queues using Windows 2000 Server. It also provides a gateway for SMB clients to print to NetWare print queues.	Optional component of Windows 2000 Server.
Print Server for Macintosh	Printing from Macintosh clients.	Optional component of Windows 2000 Server.

Server Spooler

The components below the print router and above the print device are collectively called the spooler. Each component of the spooler uses the services of the component directly below it. The spooler is a series of dynamic-link libraries (DLLs) consolidated in a single architecture, providing smooth background printing by using background thread processing. This means that the spooler passes data to the printer only when the printer is ready to receive more data.

Local Print Provider

The local print provider, Localspl.dll, writes its contents to a spool (SPL) file. It also tracks administrative information, such as user name, document name, and data type, in a shadow (SHD) file. Spooling protects a print job by saving it on disk. If a power failure or other disaster occurs before all jobs in the queue are printed, the SPL and SHD files preserve the documents and prevent loss of data after processing resumes.

Next, Localspl.dll polls the print processors for recognition of the document data type. If no data type has been set, Localspl.dll receives the job and uses the default data type from the **Print Processor** dialog box.

By default, SPL and SHD files are written to the folder %SystemRoot%\System32\Spool\Printers. If the hard disk partition containing Windows 2000 does not have enough space for these files, you can change the location of the folder.

▶ **To change the location of the default spool folder for all printers on a server**

1. In the **Printers** dialog box, on the **File** menu, click **Server Properties**.
2. In the **Print Server Properties** dialog box, click the **Advanced** tab.
3. Type the path and name for the new default spool folder, and then click **OK**.

The change is effective immediately. You do not need to restart your system.

Important Do not attempt to spool to a root (such as D:\). This causes the files to revert to the old default.

▶ **To change the location of the spool folder for a specific printer**

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

---

1. Create a new spool folder.
2. Start a registry editor (Regedt32.exe or Regedit.exe).
3. Add the following entry to the registry:

   Entry name: **SpoolDirectory**

   Path: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Print\Printers\<Printer-name>

   Data type: REG_SZ

   Value: <path to the new spool folder>

4. Restart the computer to make the change effective.

Important Do not omit step 1. Adding the **SpoolDirectory** entry to the registry does not create a spool folder. If you do not create a folder, the files spool to the root, which is the default spool folder. For the same reason, never specify a root as the new folder.

By default, SPL and SHD files are deleted after the job prints. However, by enabling spooler event logging, you can use the data they contain to get valuable data about printer traffic, hard disk space, and other printing maintenance issues. See "Auditing Printing Events" later in this chapter.

Print Processor

The print processor tells the spooler to alter a job according to the document data type. It works together with the printer driver to move the spooled print jobs from the hard disk drive to the printer. Localspl.dll is the print processor for all Windows-compatible printing; Sfmpsprt.dll is used to print to Apple devices.

Software vendors can develop their own print processors for custom data types. A printer vendor can also develop a custom print processor if the vendor has its own printer driver or supports a data type other than the five that Localspl.dll supports. Usually, the print processor is installed when the printer driver is installed.

RAW Data Type

For print jobs on computers that are not running Windows 2000 or Windows NT, the spool file's data type is RAW by default. These files are device dependent. The spooled data is destined and formatted for a particular device and does not need to be printed on a different device.

NT EMF 1.00x Data Type

With an EMF file, the GDI releases console control after generating it. The data is then interpreted in the background on a 32-bit spooler thread and sent to the printer driver. Splitting the rendering of a print job in this way is especially useful for very large documents, because the application is not tied up for the whole rendering time. The time gain is greatest if you have connected to rather than added the printer, because the background processing is done on the print server rather than the local computer.

EMF spool files are encoded to provide greater printer independence. For example, a graphic measuring 2 inches by 4 inches on a video graphics adapter (VGA) display and stored in an EMF maintains those dimensions whether it is printed on a 300-dpi laser printer or a 75-dpi dot-matrix printer. The EMF data type also ensures that the print server uses the fonts you specified.

TEXT Data Type

The TEXT data type allows you to send a simple-text print job to a print device (such as a PostScript device) that cannot interpret simple text. The spooler creates a new print job, embedding the text in print instructions derived from the print device's defaults for font, form, and orientation. The spooler uses the resolution setting specified using the **Printing Defaults** button on the **Advanced** tab in the printer's **Properties** dialog box.

Text files consist of ASCII characters. Several character sets are in common use, and text files do not indicate which one to use. The TEXT data type uses the American National Standards Institute (ANSI) character set, so it might print some characters wrong if the application uses a different set. Most character sets are identical for the values 0 through 127, so this problem usually affects only extended characters (values 128 through 255).

Windows operating systems use the ANSI character set. The default PCL character set is called Roman-8. PC-437 and PC-850 are commonly used by MS-DOS-based applications in the United States and Europe, respectively.

Print Monitors

Windows 2000 supports two kinds of print monitors: language monitors and port monitors. Port monitors are further subdivided into local and remote.

Language Monitor

The language monitor provides the common language needed for the client and printer to understand each other in bidirectional communication, so you can configure the printer and monitor printer status. You can request configuration and status from the printer, and the printer can send unsolicited status (such as "Paper tray empty") to the client.

Windows 2000 includes Pjlmon.dll, a language monitor that uses Printer Job Language (PJL) as the language. Any bidirectional printer that uses PJL can use Pjlmon.dll. For example, PJL is the language that implements the bidirectional communication between a Hewlett-Packard LaserJet 5Si and its print server.

If a printer uses a different printer language, the vendor can develop a language monitor for it. A vendor might also develop a language monitor to add data, such as printer-specific control information, to the print stream that is going to a unidirectional printer.

To take advantage of bidirectional printing, you need a bidirectional printer, an IEEE 1284–compliant cable, and a correctly configured port. Some parallel ports are set by default to IBM AT-compatible mode; you need to change the setting to bidirectional.

Local Port Monitor

The local port monitor, Localspl.dll, controls parallel and serial I/O ports where a printer might be attached. It sends print jobs to local devices, including those on familiar ports like LPT1 and COM1.

The FILE port is listed by default on the **Ports** tab of the printer's **Properties** dialog box. When you send a print job to a printer that uses this port, the spooler prompts you for the name of a file where the document needs to be stored.

You can select other local ports by clicking **Add Port** on the **Ports** tab of an existing printer property, selecting **Local Port**, and then clicking **New Port**. The **Port Name** dialog box prompts you for the name of a port to be governed by Localspl.dll. Your entry will be listed as a **Local Port** on the **Ports** tab. Possible entries include:

- A file name, such as C:\dir\file_name. All jobs sent to this port are written to the named file, and each new job overwrites the last one.
- The name of a remote print share, such as \\Server\Printer (URLs are not accepted). Jobs sent to this port are transferred over the network to the named share by the network redirector. This can be useful if you are printing remotely but you want the job to spool locally too (as in network printing on Windows 95 or Windows 98). If your printer is not connected directly to the computer, select **Standard TCP/IP Port** or a remote port monitor other than **Local Port**.
- IrDA. Use this port to connect to infrared-enabled printers meeting Infrared Data Association (IrDA) specifications. If your hardware does not support IrDA, it is not be listed on the Ports tab.
- USB. This specifies the universal serial bus (USB) port, used to connect to cameras, modems, and audio.
- 1394. This specifies the 1394 port designed for Institute of Electrical and Electronics Engineers (IEEE) 1394 bus class drivers.
- NUL. This specifies the null port, which you can use to test whether network clients can send jobs. Jobs sent to NUL are deleted without wasting paper or delaying real print jobs.

▶ **To test client connectivity by using the NUL port**

1. Pause the printer assigned to this port.
2. Send a job from a network client.
3. Look in the queue to confirm that the job arrived.
4. Resume the printer.

A second local port monitor, Usbmon.dll, initially does not appear in the Add Printer Wizard because USB printers are hot-plugging devices. Usbmon.dll is installed automatically whenever you plug in a USB printer to the correct physical port on your computer. If USB has been enabled in BIOS, Windows 2000 detects the device and displays its settings on the screen. You might be required to insert a CD-ROM containing driver files.

Installation opens the USB Root Hub and a generic USB parallel printer port (for example, USB001, 002), and copies the parallel printer point-and-print driver. Settings for the port can be modified using Device Manager.

Remote Port Monitors

All other port monitors that are supplied with Windows 2000 are remote monitors and enable printing to remote printers. An example is Lanman Print Services Port.

Sharing Printers

To install a printer for use by print clients, you must perform two major tasks:

1. Install and configure the printer on the print server.
2. Share the printer and, optionally, install the drivers for any clients that might be connecting as Plug and Play clients.

By default, the local administrator for a computer can install a printer on that computer. However, as a print server administrator, you can restrict some or all of the local choices for a user through security permissions or, in a Windows 2000 domain, user profiles.

> **Administering Remote Ports**
> Windows 2000 Server lets administrators remotely configure and manage ports from any computer running Windows 2000. This feature is applicable to local, Standard TCP/IP, and LPR ports. HP Network and AppleTalk ports must be configured on the Server console.

Local Printing

Local printers can be added from the Printers folder by starting the Add Printer Wizard. However, Plug and Play compatible printers start the Add Printer Wizard automatically when they are connected. Refer to the print device documentation for instructions on connecting the print device to your computer.

Add Printer Wizard

Printing is managed through the Printers folder which contains the Add Printer Wizard, shown in Figure 4.7.

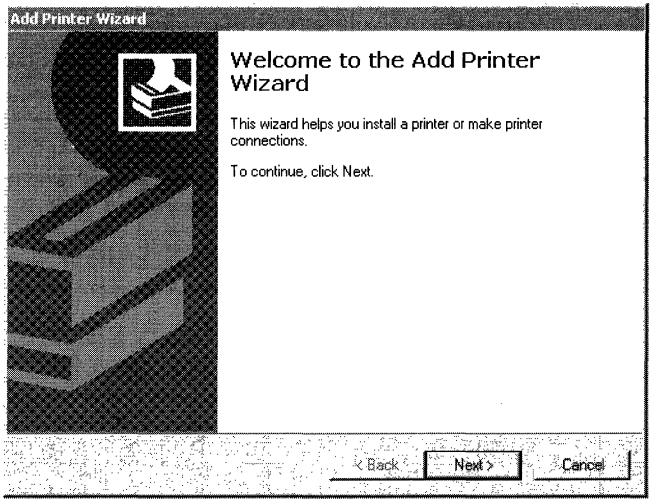


Figure 4.7 Add Printer Wizard

To install a printer under local control, click **Local printer** on the second screen of the wizard. Printer ports can be added and configured in the wizard or at another time. The wizard prompts you to install a printer driver if one is needed; or you can choose to replace the existing driver.

To connect to a printer share controlled by another computer, click **Network printer** in the wizard.

You can also use the wizard to connect to a remote printer share and install its software interface on your computer, assuming that you want local control and have the correct permissions. This bypasses the print server for the remote printer by processing print jobs locally and redirecting the output to a remote printer.

▶ **To install a printer share as a local printer**

1. In the Printers folder, double-click **Add Printer**, and then click **Next**.
2. Click **Local Printer**, and then click **Next**.
3. Click **Create a new port**, select **Local Port**, and then click **Next**.
4. In the **Port name** field, type the address of the network printer share as *\\server_name\share_name*.

5. On the **Name Your Printer** screen the print device name appears in the input field. Be sure to change the name so that your copy of the printer software cannot be confused with the printer share.
6. Finish the wizard.

   If you choose to share the printer with other network users, the wizard uses the server name and print device name to build a printer share address.

Note Clients on Microsoft 16-bit and UNIX operating systems must always create a printer share as a local printer using the installation commands specific to their systems.

Plug and Play

Plug and Play is a combination of hardware and software that allows Windows 2000 to automate the installation and configuration of new hardware. When a Plug and Play printer is connected to the computer, the printer signals the computer with a Plug and Play identification that Windows 2000 uses to determine the device type and device characteristics (for example, a printer; model 123). The Add Printer Wizard is started and the printer is either configured automatically or the operator is prompted for additional information if all the device characteristics are not known.

Print devices using a USB port or an IEEE 1394 port are detected upon physical connection and are installed automatically. A print device that connects to a parallel port (LPT) cannot be detected when its cable is attached. In this case, you must add the printer using the Add Printer Wizard with **Automatically detect my printer** selected to trigger Plug and Play.

Note When connecting a Plug and Play printer to a Centronix parallel port, the cable must meet the IEEE 1284 standard and the port must be configured as bidirectional in the basic input/output system (BIOS).

If your computer has an IrDA (infrared) port and you have an IrDA-enabled printer, turn them on, and then point the infrared port on the computer at the infrared port on the printer. The appropriate printer driver is installed and the IrDA printer appears in your **Printers** folder. You can also use the Add Printer Wizard to install an IrDA printer by clicking **IRDA** under **Available ports**.

Note Plug and Play print devices are not shared automatically. To share a print device, you must run the Add Printer Wizard.

Forwarding Print Jobs

A null session forwards a print job from one print server to another. This allows job processing to be moved to more protected sites. For example, if you create local printer B that connects to printer share A, you must share local printer B. If a user tries to connect to B and print, access will be denied unless job forwarding is enabled on the computer where printer share A exists.

By default, the null session is disabled, preventing job forwarding, but administrators can edit the registry to enable job forwarding and allow the print server that receives the print job to support null sessions.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

▶ **To enable job forwarding**

1. Start a registry editor (Regedt32.exe or Regedit.exe) on the receiving print server, and then find subkey HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Services\lanmanserver\parameters
2. Change the value of the **NullSessionShares** entry to the share name for the printer.
3. Restart the computer to make the change effective.

Remote Printing

Standard TCP/IP Port Monitor (SPM) is designed for Windows 2000 print servers that communicate with shared printers using TCP/IP. This includes network-ready printers, network adapters like the Hewlett-Packard JetDirect, and external network boxes like the Intel NetPort. SPM can support many printers on one server and is faster and easier to configure than the LPR Port Monitor.

SPM is also compatible with RFC 1759, the standard for the Simple Network Management Protocol (SNMP). As a result, SPM provides much more detailed status than LPR.

For more information about the HPmon print monitor, see "Legacy Network-Interface Printing" later in this chapter. For more information about remote printing to print devices on other platforms, see "Working with Other Operating Systems" later in this chapter.

Prerequisites

To use SPM, TCP/IP must be installed on the print server so it can talk to the print device. Clients do not need TCP/IP; any common network protocol such as SMB, NetWare Core Protocol (NCP), LPR, AppleTalk, or NetBEUI can be used. The transport protocol is not important because only the Windows 2000 print server communicates with the print device.

For example, if both the client and the server have the Internetwork Packet Exchange (IPX) protocol, the client can send the document using IPX. The server can send the document to the printer over TCP/IP. Figure 4.8 illustrates these connections.

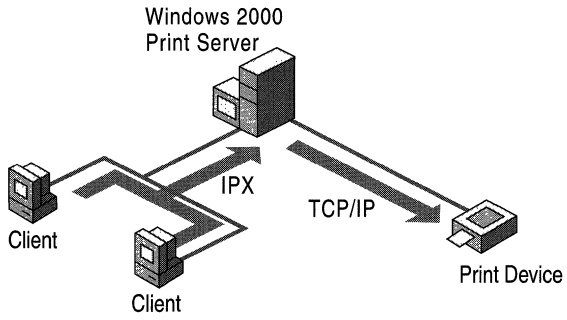


Figure 4.8 Protocol Links for Remote Printing

Print Server Protocols Used

SPM sends documents to a printer using either of two print server protocols, Raw or LPR. Together, these protocols support most current TCP/IP printers. Do not confuse these protocols with the transport protocols such as TCP/IP or DLC.

The Raw protocol is the default for most print devices:

- The print server opens a TCP stream to the printer's port 9100, or another port number, to select connections to multiport external devices. For example, on some devices port 9101 goes to the first parallel port, 9102 goes to the second parallel port, and so on.
- The print server writes to the stream, using normal TCP flow control. The data is raw page description language (PDL), and no protocol is imposed.
- The print server disconnects when it is finished sending data.

SPM uses the LPR protocol if you specify it during port installation or reconfiguration. SPM deviates from the LPR standard in one way: it does not conform to the RFC 1179 requirement that the source TCP port lie between port 721 and port 731. This range is insufficient when a server sends data to more than 11 printers. SPM therefore uses ports from the general, unreserved pool of ports (ports 1024 and above).

Port Installation Procedure

To configure a standard TCP/IP port using SPM, use the following procedure.

▶ **To configure a standard TCP/IP port using SPM**

1. Open the Add Standard TCP/IP Printer Port Wizard by clicking **Standard TCP/IP Port** from the **Add Port** button in either the **Print Server Properties** dialog box or the printer's **Properties** dialog box.



Figure 4.9 Add Standard TCP/IP Printer Port Wizard

2. Type a name or the IP address of a print device in the **Printer Name or IP Address** text box.
3. Type a port name, which can be any character string, in the **Port Name** text box, or use the default name that the wizard supplies, and then click **Next**.

   The system sends an SNMP trap to the device. Using the SNMP values returned from the trap, the device details are determined and the appropriate device options are displayed for further selection (for example, which port on a multi-port print device).

4. If prompted by the **Additional Port Information Required** screen, click **Standard**, and then select one of the listed devices.

   –Or–

   Click **Custom**, and then configure the port by using the **Configure Standard TCP/IP Port Monitor** screen that appears. If you do not know details of the port, try using **Generic Network Card**.

   If the wizard cannot determine the protocol, it prompts you for the information. If you are not prompted, skip to step 6.

5. When prompted for the protocol, select either **Raw** or **LPR**. RAW is preferred.

   If the wizard detects that the device supports multiple ports (indicated in the Tcpmon.ini file), it prompts you to select a port.

6. Select a port from the list and finish the wizard.

   The new port is listed on the **Ports** tab of the **Properties** dialog box.

Reconfiguration

The SPM port can be reconfigured in the printer's **Properties** dialog box. Click the **Configure Port** button on the **Ports** tab. The SPM has its own **Configure** dialog box that appears, as shown in Figure 4.10.

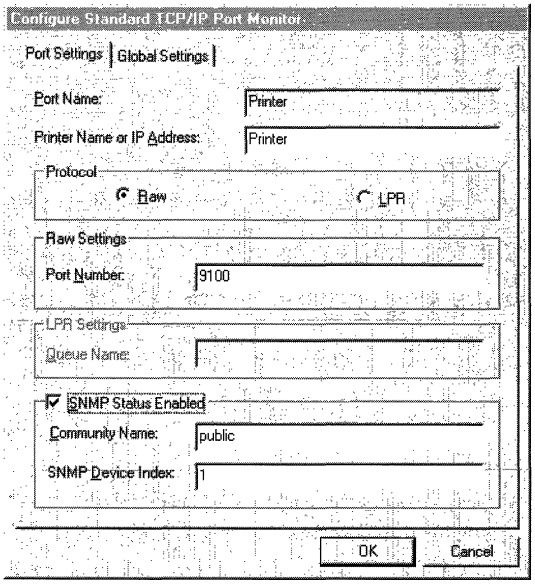


Figure 4.10 SPM Configuration Dialog Box

Warning The dialog box does not verify the settings created in the following procedure. If they are incorrect, the port no longer works. Check with the printer manufacturer to see if the device supports SNMP.

▶ **To reconfigure the Standard TCP/IP Port**

1. In the Add Standard TCP/IP Printer Port Wizard, click either **Raw** (for port 9100) or **LPR** protocol.
2. Depending on which protocol was clicked, select **Raw Settings** or **LPR Settings**.

   For **Raw Settings**, type a port number. For **LPR Settings**, type the LPR queue name specified by the printer vendor.
3. If the print device supports SNMP and RFC 1759, select the **SNMP Status Enabled** check box.
4. If **SNMP Status Enabled** is selected, you can change both the community name and the host device index.

   The device index is used mainly for multiport devices that support several printers; each port on a multiport device has a different device index, specified by the device vendor.
5. When the configuration is correct, click **OK**.

Status Reporting

Printers return status over SNMP. Since SPM is compatible with SNMP, it allows detailed status reporting if the printer provides it. Printers that are not compliant with the SNMP standard do not return status information. When there is an error during printing, the spooler displays a general printing error or does not detect any error at all.

Printer status is returned on a separate thread asynchronously from the print transmission. SPM polls the printers at intervals and sends status updates to the client.

Internet Printing

Installing and configuring a printer from the Internet is like printing on a local port except that the address is a URL. The URL can also be used within an intranet. It makes no difference to the application, which is typically unaware that the printer is accessed through a URL.

If the print server and print client are not within the same intranet, the URL is listed on the client as the printer address.

Figure 4.11 shows the path that print data takes from a client application to a print server spooler when the client prints to a URL-identified job queue.

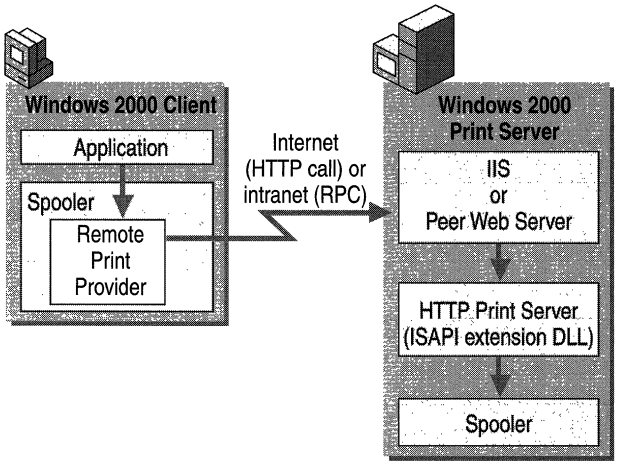


Figure 4.11 Processing Flow when Printing to a URL

Prerequisites

Before a computer running Windows 2000 Professional can print to a printer using a URL address, it must be running Microsoft® Peer Web Services (PWS). A Windows 2000 Server must have Internet Information Services (IIS). The low-level protocol used for job submission is Internet Printing Protocol (IPP) 1.0, which uses the HTTP protocol as a carrier.

Installing a Printer from a Web Page

You can view Web pages to find a URL-identified printer. You can also connect to a printer share via a Web page from clients running Windows 2000, Windows 98, and Windows 95.

Automatic installation begins on the printer's installation page (*share_name* on *server_name*) on the Web. The installation page displays the options your permissions allow you to choose. Click the **Install** option. Windows 2000 downloads the printer software to the client, and the printer is displayed in the Printers folder on the client.

The installation route depends on whether the client and print server are both running Windows 2000 or Windows NT and are within the same intranet. If they are, they communicate via RPCs, and the installed printer will continue to use RPCs to link the client and the server even if HTTP is not specified in the address.

The installation uses HTTP, not RPCs, as the communication medium in the following instances:

- The client and server are not on the same intranet.
- The client is not running Windows 2000, Windows 98, or Windows 95.
- The printer contains an internal network adapter and supports IPP 1.0, and is not connected to a server.

With HTTP, the print server generates and sends the client a cabinet (CAB) file containing the required INF and installation files. On the client computer, the CAB file starts the Add Printer Wizard to complete the installation. A progress report is displayed in HTML while the wizard is working.

Important Installation is not automatic for Web-based printers with internal network adapters. You must start the Add Printer Wizard, enter the printer's URL instead of a universal naming convention (UNC), and finish the wizard manually. This method can be used to install any URL-identified printer by means of HTTP.

Information Displays in HTML

Regardless of the underlying protocol used, the job queue appears in the standard Win32 format when you open the queue window from the Printers folder. When communication is by means of HTTP, however, the job queue is displayed in HTML from http://*server_name*/*share_name*.

If the print server is running IIS or PWS, the Printers folder on the client displays an HTML link to the Web-based printer information page, generated by the print server.

Security for Internet Printing

Print server security is provided by IIS or PWS, which runs on the print server. Basic authentication, which all browsers support, is allowed. To support all browsers and all Internet clients, the administrator must select basic authentication. IIS and PWS also allow the use of Windows NT challenge/response authentication and Kerberos authentication, both of which are supported by Internet Explorer.

The administrator selects the authentication method in IIS or PWS by clicking the **File Security** tab in the **Properties** dialog box for that server. This displays the **File Security** page as shown in Figure 4.12.

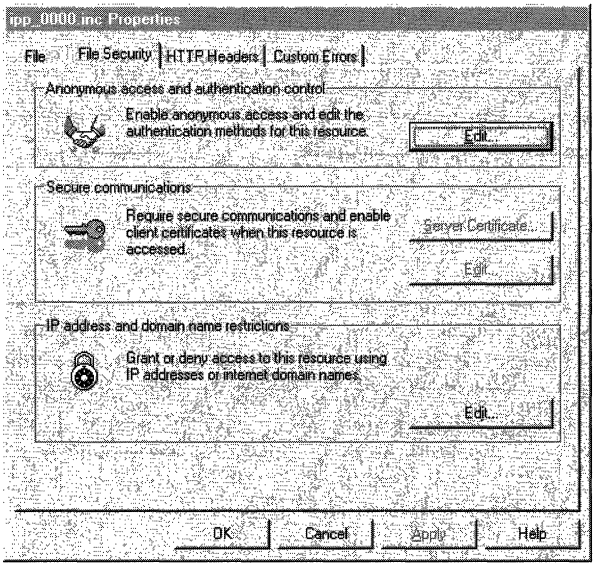


Figure 4.12 File Security Page

▶ To select an authentication method

1. In the console tree of the IIS console, expand the node for the server, and then expand the **Default Web Site** node.
2. Click the icon at the Printers node. This node represents a virtual directory that is used to set all security for Internet printing. A list of Web server proxy pages appears in the details pane.
3. Right-click a page icon, click **Properties**, and then click the **File Security** tab. The tab displays three choices for security control: Anonymous access and authentication control, Secure communications, and IP address and domain name restrictions.
4. Click the respective **Edit** button to enable either Anonymous access and authentication control or Secure communications, and then fill in details.
5. Click the **Edit** button under IP address and domain name restrictions to list exceptions to access rights and to toggle the default to either:
   - Grant access to all computers other than the listed exceptions (the initial setting).

     –Or–
   - Deny access to all computers other than the listed exceptions.

6. Click **OK** to save your settings.

Typically, administrators select **Anonymous access**. After clicking the **Edit** button, anonymous access is enabled on the **Authentication Methods** dialog box (Figure 4.13) by clicking the **Anonymous access** check box. Then click the **Basic authentication** check box to select basic authentication.

Note **Integrated Windows authentication** is checked by default and takes precedence over basic authentication if the user's Web browser supports it. To ensure that users are authenticated only with basic authentication, clear all check boxes except **Basic authentication**.

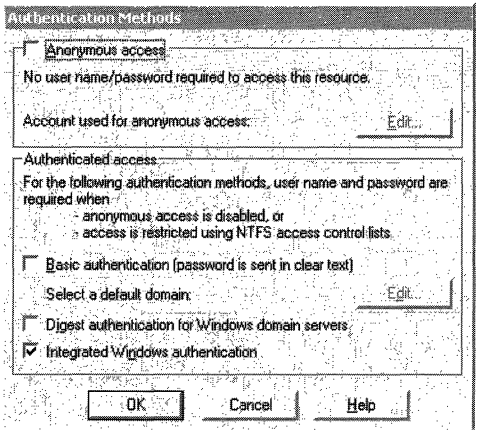


Figure 4.13 Authentication Methods Dialog Box

Checking **Basic authentication** allows all clients to access each server resource by impersonating the Anonymous account IUSR_*computername*. No user action is required. However, if a user attempts to go to another domain or proxy server that does not allow anonymous access, a dialog box appears that asks for the user name and password. If the user closes the box, it does not appear again in the same logon session.

For basic authentication, user names and passwords are not encrypted; they are base-64 encoded and can potentially be intercepted. Integrated Windows authentication is more secure because it does not send the password. IIS applies either challenge/response or Kerberos encryption technology, depending on the capability of the client, when the **Integrated Windows authentication** check box is selected. For more information about IIS security, see the *Microsoft® Windows® 2000 Server Resource Kit Internet Information Services Resource Guide*.

Earlier Network-Interface Printing

The HP Network Print Monitor, Hpmon, is used to send documents to HP network adapters that support only the DLC protocol. Such adapters include both the JetDirect device, which connects a parallel printer to the network, and the network adapters installed in some printers to connect them directly to the network.

Prerequisites

To print to a network adapter by using Hpmon, you must meet the following requirements:

- DLC must be available on the network on which the print device resides.
- The DLC protocol must be installed on both the print server and the network adapter and enabled on the print server.
- Any intervening subnet junctions must be bridged rather than routed. DLC is not routable.
- The hardware address of the network adapter is needed for the print device.
- Install Hpmon on the print server from the Windows 2000 product CD.

Creating a Port

Hpmon is installed when the DLC protocol is installed. You can create a port using the **Port** tab in either the **Print Server Properties** dialog box or the printer's **Properties** dialog box. Double-click **Hewlett-Packard Network Port**.

In the **Name** field, type a logical name for the port. It is used to identify the printer and is associated with the card address (or media access control (MAC) address). In the **Card Address** field, click the network adapter address. If the adapter is not on the network or you are working offline, the address is not listed and you must type it in. If it must be listed but is not, see if the adapter is correctly attached.

Hpmon does not allow more than one print server at a time to access the same network adapter, although each server can access it through more than one port. Configure the Hpmon ports as either a **Job-based connection** or a **Continuous connection**. The default is **Continuous**.

With **Job-based connection**, the print server connects to the adapter, sends a document, and disconnects when the document is printed. The connection is restored when you send another document. This enables other print servers to connect and print.

With **Continuous connection**, the print server stays connected until either it or the adapter is restarted. Security is more centralized because others cannot print to the device unless they manually restart it.

Note If you configure two Windows 2000 print servers to send documents to the same network adapter, set both servers to **Job-based connection**, or one blocks the other from connecting.

Working with Other Operating Systems

Windows 2000 print servers can accommodate a wide variety of computing platforms, operating systems, and network protocols.

Printing from Windows 95 and Windows 98 Clients

Windows 95 and Windows 98 print clients can use Windows 2000 print services without adding software. However, to use the Active Directory™ directory service for a printer search, these clients must have Active Directory Client Upgrade installed. This utility is included with Windows 2000 Server.

Because of the cross-platform point and print functionality in Windows 2000, drivers for clients running Windows 95 and Windows 98 can be installed on the print server; when a client connects, it downloads the drivers from the print server and begins printing. Select the **Sharing** tab from the printer properties on the print server and select the **Additional Drivers** button. From this window, you can select the **Windows 95 or 98** check box. You are then prompted for the location of the Windows 95 or Windows 98 driver files. These files are located on the Windows 2000 Server Setup CD in the directory Printers\win9x. The Add Printer Wizard looks for the printer name in any INF files in the specified location and attempts to install the drivers identified in the INF file.

Printing from Other Microsoft Print Clients

The Windows 3.*x*, MS-DOS, and OS/2 operating systems use their own printer drivers. Clients with these systems can get a list of all available printers in a domain but cannot use Active Directory. These clients can print to a Windows 2000 print server by installing a local printer and redirecting the local port to the network printer share on the Windows 2000-based server. For example, if a printer is installed on LPT1 and you type the following on the command line:

net use lpt1: \\Servername\Sharename

The output to the LPT1 port is redirected by the client network redirector.

Required Redirectors

Network redirectors are required because 16-bit clients print to ports rather than printers and are unaware of the printers in the Printers folder. This limits most 16-bit applications to local ports.

To use remote printers, install Microsoft® LAN Manager Client Version 2.2c (for OS/2) or Microsoft Network Client version 3.0 for MS-DOS (for MS-DOS or Windows 3.*x*). Redirectors are not needed for Microsoft® Windows® for Workgroups (Windows 3.11), which has a built-in redirector.

Note Some 16-bit applications do not print to addresses that are longer than 31 characters; some do not print to any printer if the default printer has a long address. Also some 16-bit applications do not support spaces in addresses.

Output Destination Rules

The rules for directing output differ between 16-bit applications and Windows-based 32-bit applications. Windows 2000 accommodates 16-bit applications in the following ways:

- If the redirector controls the port (as when a **net use** command redirects output to a shared resource), the redirector determines where the document goes.
- If the port is not controlled by a redirector but has a printer assigned to it, the document goes to that printer. That printer's spooling options take effect.
- If the port is not controlled by a redirector and no printer is assigned, the document is sent directly to the port device driver and prints unaltered.

For example, print server \\Server1 has two shared printers defined in its Printers folder. Printer HPV prints to an LPT2 port supplied by a separate I/O card, and printer HPIIISi prints to the FILE port. LPT1 and LPT2 each have one print device attached. Type the **net use** command:

net use lpt3 \\server1\hpiiisi

Copy a file to the parallel ports. In the following results, **/b** is the switch for a binary transfer:

copy test.txt lpt1:/b

The client checks the redirector and finds that it does not manage LPT1. It checks the spooler and finds that neither printer prints to LPT1. The document is sent to the parallel port device driver. To do this, type:

copy test.txt lpt2:/b

The client finds that the redirector does not manage LPT2. It finds that printer HPV prints to LPT2. The job goes to HPV, whose spooling options take effect as the document prints on LPT2. Type:

copy test.txt lpt3:/b

The client finds that the redirector is managing LPT3, so the redirector takes control. The **net use** command has redirected LPT3 output to the print share \\Server1\HPIIISi, where the spooling options of HPIIISi take effect.

Note Although **net use** is typically used to assign a local port to a remote shared resource, it can also assign a local port to a local shared resource. This is useful in testing and troubleshooting.

Printing with UNIX Clients and Servers Using LPR/LPD

A line printer remote (LPR) utility lets an application on one computer print to a spooler on a remote computer. The receiving utility is called a line printer daemon (LPD). The LPR/LPD combination was developed for UNIX computers but is widely used for many operating systems. Both utilities are included in Print Services for Unix.

▶ **To install Print Services for Unix**

1. From the **Start** menu, point to **Settings**, click **Control Panel**, double-click **Add/Remove Programs**, and then click **Add/Remove Windows Components**.
2. Click **Next**.
3. In **Components**, click **Other Network File and Print Services** (but do not select or clear its check box), and then click **Details**.
4. Select the **Print Services for Unix** check box.
5. Click **OK**, and then click **Next**.

Printing Documents Received Through the LPD Utility

The LPD utility, Lpdsvc, receives print jobs from computers that have LPR utilities. Any client that is fully RFC 1179 compliant can use LPR to send printing to Lpdsvc on a printer server that is running Windows 2000. Partially compliant systems might fail.

Note Windows 2000 also has an emulation subsystem in which you can run and print from UNIX applications written to the POSIX.1 standard. The control commands and processing are the same as in native UNIX systems.

Prerequisites

LPR print clients can print to a print server running Windows 2000 by sending straight text and using the **f** command so that Windows 2000 formats the print jobs properly. The following connections must be in place:

- Both client and server have TCP/IP, and the client has an RFC-compliant version of LPR.
- Print Services for Unix is installed on the print server.
- The printer share name is identified to the LPR software on the client. The share name may differ from the printer name in the Printers folder and in the printer's **Properties** dialog box.

In Windows 2000, the LPD utility is included in Print Services For Unix. When installed, it starts automatically for all shared printers. LPR clients know the printer names as the Windows 2000 share names.

Directing Output

Use the LPR utility on the client to direct the output to a printer on the print server. For example, on a Windows 2000 client, type the following on the command line:

lpr -S servername -P queuename filename

Other LPR clients might use a different command or syntax.

Default Data Types

Print Services for Unix sets the print data type when it sends the document to the spooler. This is derived from the control command included in a print job from an LPR client. It might be necessary to change the default data type at the client to avoid processing PCL or PostScript print jobs as TEXT when they are actually RAW. For more information about the data types, see "Print Processor" earlier in this chapter.

If the control command is **f** or **p**, the data type is TEXT, and the spooler edits the document file to print correctly. If the command is **l**, the data type is RAW and no editing is done. If the command is **o**, the document is already formatted in PostScript code and is assigned the RAW data type.

Some UNIX systems usually send the **f** command by default, resulting in the following symptoms:

- The output includes PCL or PostScript code.
- Extended characters are printed incorrectly.
- The printer's default font is used.
- An extra page is printed at the end.

You can make a registry entry to force Windows 2000 to process the data as RAW. For more information, see the see the Knowledge Base link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources. Search for the keyword SimulatePassThrough.

Sending Documents Through the LPR Utility

The LPR Monitor (Lprmon) provided with Print Services for Unix acts as the client for printing according to RFC 1179 guidelines and spools to an LPD-enabled target host. The target is usually a UNIX, MVS, or VAX/VMS-based computer but can also be a network adapter, an external network box, a Windows NT–based computer, or a print server running Print Services for Unix or another version of LPD.

Prerequisites

To use Lprmon, the client must have TCP/IP and Print Services for Unix installed. The target server must be running a Berkeley Style Daemons (BSD)–compatible LPD utility according to RFC 1179. If the server is running Windows 2000 or Windows NT, check that Print Services for Unix is started.

Entering the Printer Address

While installing this monitor, the Add Port Wizard asks you to enter the **Name Or Address Of Host Providing LPD** and **Name Of Printer On The Target LPD**. The printer name is defined by the LPD and varies based on the manufacturer. Please refer to the LPD product documentation for the printer names.

Host Name or IP Address

For the host name, enter the DNS name or the IP address. The DNS name of the target computer needs to be defined in the host file on the Windows 2000 print server **hosts** file. You can use the **ping** command to verify the name and address of the target computer. If the connector is a network adapter instead of a computer host, see the documentation provided with it.

Printer Name - Host

If the target host is a UNIX computer and you do not know its name, log on to the UNIX computer that is running a BSD-compatible LPD daemon. To view the target's /etc/printcap file, at the command prompt, type:

cat /etc/printcap

Each entry in a UNIX /etc/printcap file corresponds to a UNIX print queue. The first field of an entry lists names for the print queue (for example, LP, Lablaser, or The_Lab_Printer), separated by "|" characters and ending with a colon. For example, a 9600-baud TTY printer might show a listing such as:

```
lp|lablaser|the_lab_printer:\
:lp=/dev/ttya:br#9600:\
:lf=/usr/spool/lpd/lablaser-err:\
:sd=/usr/spool/lpd/lablaser:
```

The second line tells what kind of printer is used. You can enter any of the names for the print queue in the Add LPR Port Wizard. Alternatively, to display a list of all defined print queue names on the host, type:

lpc stat all

Note This example is for illustration only. Refer to UNIX documentation for information about the /etc/printcap file.

Printer Name - Network Adapter

Because the printer name varies depending on the manufacturer, see the adapter's documentation for details. Typical examples include TEXT, PASSTHROUGH, RAW, P1, P2, and so on.

Changing the Default Control Command

The LPR sends a processing instruction in each print job in the form of a control command: **f**, **l**, **o**, or **p**. The default is **l** for Lprmon and **f** for Lpr.exe, but it can be changed by modifying a registry entry.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

▶ **To change the default command for a particular printer**

1. Start a registry editor (Regedt32.exe or Regedit.exe).
2. Add the following entry to the registry:

   Entry name: **PrintSwitch**

   Path: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors\LPRPort\Ports\<Port-name>\<IP Address or Host name:Printer-name>

   Data type: REG_SZ

   Value: <control command>

Normally, you would set the default value to **l**, but this can cause a problem when sending an ASCII text file to a PostScript printer on a UNIX computer. The **l** value sets the RAW data type, and the PostScript instructions are ignored. Many UNIX systems add software that scans arriving documents for PostScript code with the **l** value. If it is found, the document goes directly to the printer; otherwise, the software adds PostScript code.

Not all scanners recognize the output of the Windows 2000 PostScript driver as valid PostScript code, and the printer prints PostScript code instead of interpreting it. To correct this, set **o** as the default value for this registry entry.

Respooling

LPR must include an accurate byte count in the control file, but it cannot get it from the local print provider. After Lprmon receives a document from the local print provider, it spools it a second time as a temporary file in the System32 subfolder, finds the size of that file, and sends the size to the LPD print server.

Status Reporting

The LPR protocol does not return a detailed error status report. If there is a problem, the message is always ERROR. To find the specific problem, see Windows 2000 Server Help.

Printing with NetWare Clients and Servers

Microsoft has three products that provide printing compatibility with NetWare systems: File and Print Services for NetWare, Client Service for NetWare, and Gateway Service for NetWare.

Printing Documents from NetWare Clients

A print server that is running Windows 2000 Server can process print jobs from NetWare print clients if the File and Print Services utility is installed. File and Print Services does not run on Windows 2000 Professional. The NetWare clients can be connected to the print server's network directly or through a NetWare server.

Windows 2000 File and Print Services also allows NetWare clients to print to NetWare-compatible printers that are attached directly to the Windows 2000 network.

Prerequisites

You must ensure that communication with NetWare is enabled by means of the NWLink protocol (in full, NWLink IPX/SPX/NetBIOS Compatible Transport). File and Print Services installs NWLink software automatically if it is not already present on the Windows 2000 print server. NetWare connectivity over TCP/IP is not supported.

Printer Search Facility

With File and Print Services, the Windows 2000 print server appears to the NetWare client like a NetWare 3.*x*-compatible file and print server. File and Print Services presents the same dialog boxes to the client as a NetWare server uses to process a print job from a client. The printers on the Windows 2000 print server are displayed and searched for like NetWare print queues.

Directing Output

To print by means of File and Print Services, NetWare clients must download the printer driver, duplicating the shared printer as a local printer. They cannot use the RPRINTER printing method, although they can use both PSERVER and LDP. Ports must be specified as follows:

- For a printer attached to the Windows 2000 print server, select port LPT1.
- For a networked printer, select either the NetWare_Pserver_0 or NetWare_Pserver_1 port.

Sending Documents to a NetWare Network

Gateway Service for NetWare allows Windows 2000 Server to send print jobs to printers using Novell PSERVER technology on a NetWare network.

When using Gateway Service for NetWare, requests from Microsoft networking clients are processed through the gateway so access is slower than direct access from the client to the NetWare network. Clients that require frequent access to NetWare resources must run Client Service for NetWare to bypass the Windows 2000 print server.

Prerequisites

To use this capability, do the following on the server that is making the connection from the Windows 2000 network:

- Ensure that the NWLink protocol is installed on the computer. Gateway Service for NetWare installs it automatically if it is not already present.
- Install Gateway Service for NetWare on a computer running Windows 2000 Server. To install these utilities, open **My Computer, Control Panel, Network and Dial-up Connections,** and then right-click **Local Area Connection**. Click **Properties**, **Install**, **Client**, **Gateway (and Client) Services for NetWare**, and then click **OK**. Click **Close**.

Note Client Service for NetWare is also installed automatically on the server so you can test your installation by sending a print job from the Windows 2000 server to the NetWare printer.

- Add a NetWare port. This requires you to install the NetWare remote port monitor, Nwmon, which manages communication between the Windows 2000 print server and the print device.

Printer Search Facility

With Gateway Service for NetWare, you can view available NetWare printers in the Add Printer Wizard or by using **net view**.

Connecting to the NetWare Printer Share

Gateway Service lets clients connect to a NetWare printer share the same way they connect to a printer share on the Windows 2000 network. Windows 2000 downloads the printer driver. Printer settings, such as paper size, are not retrievable on a NetWare print server, so you need to adjust them on the client.

Unlike Windows 2000, the NetWare printer driver is not automatically kept current on the client. Use the Add Printer Wizard to update it.

Print Processing

In Novell PSERVER processing, the server grabs the document from a passive print queue, rather than the print server driving the transport of the document.

The Microsoft NetWare print provider, Nwprovau, performs the document transfer. If Nwprovau recognizes the printer name when the print router on the Windows 2000-based print server polls it, it takes control of the print job and sends a message to the NetWare Workstation Service (Nwwks), which in turn passes control to the NetWare redirector. The NetWare redirector sends the print job to the NetWare print server, as shown in Figure 4.14.



Figure 4.14 NetWare Print Provider

Printing with Macintosh Clients and AppleTalk Printers

Print Server for Macintosh allows Macintosh clients to connect to printers shared on computers that are running Windows 2000 Server. The AppleTalk protocol also installs Macintosh Port Monitor (Sfmmon) which allows computers that are running Windows 2000 Professional or Server to print to AppleTalk print devices. Windows 2000 can only send print jobs to a Macintosh computer using AppleTalk when there is third-party software running on the Macintosh.

You only need install the AppleTalk protocol to print to AppleTalk printers on a network. If there are Macintosh clients on the network that need to spool print jobs to the Windows 2000 server, install Print Server for Macintosh on the server.

Printing Documents from Macintosh Clients

Macintosh clients can print to a print server that is running Windows 2000 Server with AppleTalk and Print Server for Macintosh. To the Macintosh client, the Windows 2000–based computer looks like an AppleTalk print device on the network.

Printer Search Facility

Macintosh clients can browse all printers available on the network by using the **Chooser** dialog box. These printers can be distributed over several Windows 2000 print servers. Macintosh users do not have access to Active Directory.

Prerequisites

No reconfiguration of the Macintosh client is required. You must install Print Server for Macintosh to receive the print jobs on the print server. Installing Print Server for Macintosh also installs the AppleTalk protocol.

Print Server for Macintosh is installed from Control Panel. Select **Add/Remove Programs**, **Add/Remove Windows Components**, **Other Network File and Print Services**, and then click **Details**. Click **Print Server for Macintosh**, click the check box, click **OK**, and then click **Next**.

Data Types

Setting up Print Server for Macintosh installs a print processor, Sfmpsprt, that assigns either of two data types, listed in Table 4.2, to a document.

Table 4.2 Data Types for Print Server for Macintosh

Data Type	Instructions to Spooler	Use
RAW	Print the document with no changes.	For all documents targeted to PostScript printers
PSCRIPT1	Convert the document to rasterized images, or bitmaps.	For all documents targeted to non-PostScript print devices

The PSCRIPT1 data type means that the document file is Level 1 PostScript code from a Macintosh client but the target printer is not a PostScript printer. The spooler sends the PostScript code through a Microsoft® TrueImage® raster image processor, which creates a series of one-page, monochrome bitmaps at 300 dots per inch (dpi) maximum. The printer driver returns a print job that prints the bitmaps on the page.

Because the monochrome, 300-dpi limitation is in the raster image processor software, it applies even if the driver supports color or higher resolution. (PostScript printers are not affected.) For those who need a higher-end raster image processor, original equipment manufacturer (OEM) Win32 raster image processor packages are commercially available for Windows NT 3.1 and later versions.

Note Do not send binary PostScript jobs to a server running Windows 2000. Because Windows 2000 is not designed for this type of input, all output is garbled.

User-Level Security

Native Macintosh networking supports security for files but not for printers. AppleTalk has no mechanism supporting client user name or password. Thus Macintosh print clients cannot identify themselves on a Windows 2000 network, and the print server cannot impose user-level security on them. If a Macintosh client is physically able to send a job to the printer, it has implicit permission. However, you can set user-level permissions for all Macintosh print clients as a group.

▶ **To limit user permissions for Macintosh print clients**

1. Create a new user account for the group, to take the place of the System account, giving this account the print permissions that you want the group to have.

   By default, the Macintosh MacPrint service logs on as the System account. That account has print permission on all local printers and thus gives Macintosh clients access to any local printer on the Windows 2000 server.

2. Set the MacPrint service to log on using the new account.

Tip This procedure can be used to bypass a restriction of the System account, that is, the inability of a System account on one computer to access resources on other computers. Thus MacPrint, logging on as System, cannot send print jobs to printers that forward jobs to other print servers. The solution is to give the new user account permission to print on all print servers to which print jobs are forwarded.

Sending Documents over AppleTalk

Windows 2000 has a Macintosh port monitor, Sfmmon, that sends print jobs to remote printers such as the Apple LaserWriter family using the AppleTalk protocol. It also lets you send jobs to AppleTalk spoolers regardless of the printer to which the spooler is attached. This monitor also enables any computer running Windows 2000 to send local print jobs to AppleTalk printers. Macintosh clients can also send local print jobs to AppleTalk printers but might not share these printers with other clients.

Note Some printers misprint non-PostScript documents if these documents are received over AppleTalk; others misprint PostScript documents containing binary data if they arrive over any protocol but AppleTalk. These problems result from restrictions in those printers and do not mean that Windows 2000 is transmitting documents incorrectly.

Prerequisite

For a computer running Windows 2000 to print to a remote Macintosh device, the AppleTalk protocol must be installed on both the sending computer and the receiving print device. If the sending computer running on a Windows 2000 Server configuration has Print Server for Macintosh installed to provide access to Macintosh clients, AppleTalk was installed automatically.

Printer Capture

A captured printer accepts print jobs only from the Windows 2000 print server, thus giving you complete control over the printer. In general, it is best to capture a printer unless another source (such as a Macintosh server) prints to it. If a printer is used only by Windows 2000, it is recommended that you capture it.

Capture ensures that users do not accidentally bypass the print server and send print jobs directly to the printer or reset the printer, which might cause spooler problems. It also prevents job contention caused by the client preparing the printer to accept the same level of PostScript printing that the driver on the client supports. This can cause excessive cycling of the printer, taking up excessive time and stressing the print device.

If a printer is not captured, both Windows 2000 Server and Macintosh users can send print jobs to it. You can enable or disable printer capture by rerunning the Add Printer Wizard and answering **No** or **Yes** to **Do you want to capture this AppleTalk printing device?**

Printing and Active Directory

A directory service needs to make it easy for users to find printers. In Windows 2000, the print subsystem is tightly integrated with Active Directory, making it possible to search across a domain for printers at different locations.

Printing and Directory Service Overview

Active Directory is a distributed database shared by the domain controllers in a network. Information about printer queues, sites, names, and addresses is kept in Active Directory. This information must be sent by individual print servers, as shown in Figure 4.15. It is important to keep the printer information that is stored in Active Directory up-to-date.

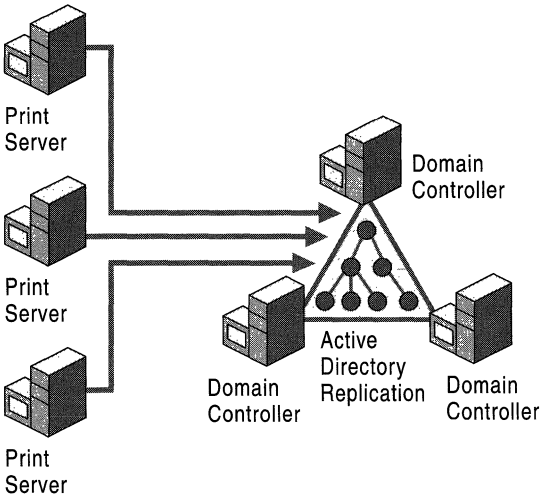


Figure 4.15 Print Servers and Active Directory

Pertinent characteristics of the relationship between print servers and Active Directory include the following:

- Each print server is responsible for publishing its own printers in Active Directory.
- The print server does not have an affinity to any specific domain controller—it dynamically finds a domain controller in the appropriate domain.
- When a printer is updated on the print server, the changes are automatically propagated through Active Directory.

- Printers are published in Active Directory as **printQueue** objects. The published **printQueue** object contains a subset of the information stored on the print server for a printer.

By default, the integration of printing with Active Directory is configured to work without administrative intervention. You only need to make changes if the default behavior is not acceptable. The default behavior includes the following:

- Any printer shared by a print server is published in Active Directory. This still requires administrative access to the host computer to install and share a printer.
- The **printQueue** object is placed in the print server's **Computer** object in Active Directory. To learn more about the information that is propagated into the Active Directory **printQueue** object, see the Platform Software Development Kit (SDK) link on the Web Resources page at http://windows.microsoft.com/windows2000/reskit/webresources.
- When any change occurs in the printer's configuration, the Active Directory object is updated. All the configuration information is resent to Active Directory even if not all of it has not changed.
- If a print server disappears from the network, its printers are removed from Active Directory.

Publishing Windows 2000 Printers

You can only publish printers that are shared. Printer publishing is controlled by the **List in the Directory** check box on the printer's **Sharing** tab.

The Add Printer Wizard does not let you change this setting when you create a printer. Printers that are added using the Add Printer Wizard are published by default. If you do not want a printer published in Active Directory, on the **Sharing** tab of the printer properties, clear the **List in the Directory** check box.

Note For a USB printer that is detected and installed automatically, you must manually share and publish it using the **Sharing** tab.

The printer is placed in the print server's **Computer** object in Active Directory. Once it has been placed in Active Directory, the object can be moved or renamed using the Active Directory Users and Computers console.

Publishing a Printer Manually

To publish a printer in Active Directory manually, you must first set the pruning interval to "never." If you do not do this, the printer you manually add to Active Directory is removed the next time the pruner runs. Then publish the printer using the Add Printer Wizard to permanently create the **printQueue** object.

Publishing Mechanism

The print server sends data asynchronously to Active Directory. Initially it sends the data after a one-second delay. If this fails, the server retries after a longer period and continues until it reaches a delay of two hours. At this point the server retries at this interval until it is successful. During this pending phase, the message "The Directory operation is still in progress" is displayed on the **Sharing** tab of the printer properties.

The printer is published to a random domain controller, so a query might not show the printer until it has been replicated to all the domain controllers. For local domain controllers on the same site, the maximum delay is approximately 30 minutes, but is typically 5 – 10 minutes. For inter-site searches, the delay depends on the replication strategy of your organization.

Pruning Orphans

When a printer is deleted from a print server, the corresponding Active Directory object is removed. However, there can be circumstances where the printer is no longer available. For example, when the print server is being rebuilt or is powered off.

Under these circumstances the orphaned entries must be removed because Active Directory must reflect only the print devices that are currently available. A program called an orphan pruner accomplishes this by running on each domain controller to periodically check for orphaned printers. If a printer does not exist, it is deleted. The pruner only looks at print servers that are in the same site as the domain controller on which it is running.

The orphan pruner is controlled by several policy settings. By default, if the pruner cannot see a printer three times in a row at 8-hour intervals, it assumes that the entry is no longer valid and deletes it.

In some cases, the pruner might remove a printer that still needs to be published. For example, if a print server is down for a few days, its printers need to be removed. Once the server comes back up, its printers need to be republished. To cover this situation, a print server verifies that its printers are published when it restarts and the spooler starts up. You can force a restart by issuing the commands:

```
net stop spooler
```

and

```
net start spooler
```

Alternatively, you can use the "check published state" policy.

Sites Without Domain Controllers

It is unusual but possible to create an Active Directory site that has no domain controllers. If this case, the print servers in the site are never pruned. To resolve this, you can make another domain controller responsible for managing this site. This is also required for users to log on to the network.

▶ To add a site to a domain controller in another site

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

1. Start a registry editor (Regedt32.exe or Regedit.exe).
2. Add the **SiteCoverage** entry to the registry. For its value, list each additional site that the domain controller must cover:

   Path: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters\SiteCoverage

   Data type: REG MULTI_SZ

   Value: *<site name>*

Supporting Non-Windows 2000 Printers

Printers that are on print servers running Windows NT 4.0 or Windows NT 3.51 can be published in Active Directory by using the Users and Computers console. Alternately, you can use the Pubprn.vbs script, which is provided in the System32 folder. You can publish all printers on a server or specify single printers to be published.

To run Pubprn.vbs, type:

Cscript c:\winnt\system32\pubprn.vbs <params>

If you don't enter any parameters, the following help message is displayed:

```
Usage: [cscript] pubprn.vbs server "LDAP://OU=..,DC=..."
    server is a Windows NT server name (e.g.: Server) or UNC printer
name (\\Server\Printer)
    "LDAP://CN=..,DC=..." is the DS path of the target container
Example 1: pubprn.vbs MyServer
"LDAP://CN=MyContainer,DC=MyDomain,DC=Company,DC=Com"
Example 2: pubprn.vbs \\MyServer\Printer
LDAP://CN=MyContainer,DC=MyDomain,DC=Company,DC=Com"
```

For instructions about publishing printers on other network operating systems, see the *Microsoft® Windows® 2000 Server Resource Kit Internetworking Guide.*

Group Policy Settings

Table 4.3 lists group policies related to printing. The Computer column indicates whether this policy must be set for the domain controller or for the print server. It is recommended that you have one or more computer groups for the domain controllers and one or more computer groups for the main printer servers to set the policy on those groups.

Table 4.3 Important Group Policy Settings for Printing

Name	Description	Computer	Default
Pruning Interval	The length of time that the pruner waits before starting another Active Directory scan.	Domain controller	8 hours
Pruning Retries	The number of scans after which the pruner deletes the printer from Active Directory if it is missing.	Domain controller	3

(continued)

Table 4.3 Important Group Policy Settings for Printing *(continued)*

Name	Description	Computer	Default
Backward Compatibility Printer Pruning	You might want a different pruning policy for non-Windows 2000 printers. The default is to never prune. Alternately, you can choose to prune only if the print server confirms that the printer is absent.	Domain controller	Never
Allow Pruning of Published Printers	If true, the printers published by this computer are candidates for pruning. If false, the pruner never attempts to prune the printer on this server.	Server	True
Allow Printers to be Published	Disable this option to prevent printers from being published	Server	Yes
Automatically Publish New Printers	Disable this option to prevent the Add Printer Wizard from automatically publishing a new printer when it is added	Server	Yes
Check Publish State	If this is set to any other value than never, the print server checks if its printers are published at regular intervals. (The check is the same as what is performed at system startup).	Server	Never

For more information about Group Policy, see "Group Policy" in the *Microsoft® Windows® 2000 Server Resource Kit Distributed Systems Guide*.

Printer Location Tracking

Printer location tracking in Windows 2000 allows users to search for and find printers at their location or another specified location, according to attributes assigned to printers.

For more information about printer location tracking and procedures for setting it up, see Windows 2000 Server Help.

Group Policy lets you configure printer location tracking for a group of computers. See the "Computer location" and "Pre-populate printer search location text" policies in Computer Configuration\Administrative Templates\Printers. For information about these policies, see the **Explain** tab for each policy or the Windows 2000 Resource Kit Group Policy Reference (Gp.chm) on the *Windows 2000 Resource Kit* companion CD.

Printing and Clusters

The Cluster service is a virtual server that supports mission-critical applications and data by allowing a group of independent computers to work together as a single system. The cluster appears to network clients as a single server. All computers in the cluster are known by a common name—the cluster name. Each computer is a node, and if one node fails, the cluster's resources "fail over"—move over to a node that is still operating.

Cluster service runs only on Windows 2000 Advanced Server. However, after it is set up on Windows 2000 Server computers, it can be administered, including installing and configuring printers, from any computer running Windows 2000 in the network.

Creating the Spooler

Adding printers requires the creation of a spooler for the cluster so the cluster can be used as a print server. Without a spooler, you cannot add printers.

Prerequisites

To create a spooler for a cluster, you must be the administrator of the cluster as well as of each node within the cluster.

You must also have the administration software installed on your computer. Cluster Administrator is the graphical application supplied with the Cluster service package to manage clusters. Alternatively you can use Cluster.exe, a command-line tool, or custom administration tools developed using the Cluster service command interfaces.

Specifying the Resource

The network applications, data files, and other tools installed on the nodes are the cluster resources, which provide services to network clients. The spooler is specified by selecting the **Print Spooler** resource type in the **Cluster Administrator** window or using a Cluster.exe command. Resources are organized into groups, consisting of resources that are interdependent and therefore must fail over together. There can be no more than one Print Spooler resource per cluster group.

The procedure to specify a Print Spooler resource is as follows. It is like the procedure for creating any resource in Cluster service, and more details can be found in other chapters in this book.

▶ To create a Print Spooler resource

1. In the **Cluster Administrator** window, start the New Group Wizard, and then create a group for the print spooler.

   This group, called the virtual server, is used to contain all resources necessary to create a printer share.

2. Add an IP Address resource to the group.

   This resource is a prerequisite for the Network Name resource, created next. If the nodes have Print Services for Unix installed and started, clients with the LPR utility can also use this IP address to access the printer.

3. Add a Network Name resource to the group.

   This is the server name (cluster name) that clients will use when connecting to the printer share.

4. Add a Physical Disk resource to the group.

   The Physical Disk resource is where the spooler files for the printer shares will be stored. This resource must be on the SCSI bus that is common to the nodes.

5. Add the Print Spooler resource to the group.

   Ensure that the spool directory is placed on a Physical Disk resource that is part of the group.

6. Configure the possible owners.
7. Add the Physical Disk and Network Name resources as dependencies.
8. On the **Parameters** property sheet for the Print Spooler resource type, configure the following options:
   - **Spool folder**: type a fully qualified path for the spool folder (for example, G:\Spool). The spooler creates the folder if it does not already exist.
   - **Job completion time out**: specify how long the document can take to get from the client to the printer before the printer stops trying to print it.
9. Use Cluster Administrator to bring the Print Spooler resource online.

After you have created and configured the Print Spooler resource, you need to install printer drivers on each node and add printers to the clustered spooler.

Installing Printing Components

This process does not use Cluster Administrator. Instead, perform the following actions. Only TCP/IP and simple LPR ports are supported.

- Go to the Printers folder for each node and ensure that each has the appropriate protocols, port monitors, language monitors, print processors, and forms installed. You can do this remotely, but you must address each node by its node name, not the cluster name. A specification made on one node does not carry over to the other nodes, so ensure that the settings are identical on all nodes.
- On each node install the printer drivers you want to use for the cluster. Each node of the cluster must have the correct printer drivers installed for all clients that connect to the printer share. Otherwise, Point and Print does not work and clients are prompted to install drivers when they connect to the printer the first time.
- Add printer ports from the Cluster virtual computer Printers folder in addition to the cluster node computers.

Adding a Printer to a Cluster

After creating the group, resources, ports, and printer drivers, you can add a printer to the cluster. Each node must have connectivity to the remote print server device (port). A locally connected computer cannot be used in a cluster configuration because the printer is connected directly to the node and does not fail over if that node goes down.

Adding a printer to a cluster is the same as adding it to any other computer, with the following exceptions:

- Never start from the local Printers folder. The cluster always appears remote, even if you are working on the active cluster node. Instead, type the cluster name (for example, \\Cluster-prn) in the **Run** menu. Then click the remote Printers folder that is displayed.
- If the Add Printer Wizard does not appear when the remote Printers folder opens, you cannot continue. One of three things is wrong:
  - You do not have administrator permissions on each cluster node.
  - The spooler service is not started on the current computer (unlikely).
  - The cluster does not have at least one Print Spooler resource online.

What Happens at Failover

When a group containing a Print Spooler resource fails over to another node, the document that is currently being spooled to the printer is restarted from the other node after the failure. When you move a Print Spooler resource or take it offline, Cluster service waits until all documents are finished spooling or until the configured wait time has elapsed. Documents that are spooling from an application to a Print Spooler resource are discarded and must be respooled to the resource (or reprinted) if the group containing the resource fails over before the application has finished spooling.

Monitoring Printer Performance

Windows 2000 provides several features that allow you to assess the performance of your network printing resources.

Using System Monitor

System Monitor allows you to observe counters for printing performance. It also supports setting alert levels for the counters being monitored. Individual counters and objects are present for each instance of a printer that is installed.

The two most critical performance counters for monitoring printing are the following:

- Bytes Printed/sec. The number of RAW bytes per second that are sent to the printer. Low values for this counter can indicate that the printers are underutilized, either because there are no jobs, print queues are not evenly loaded, or the server is too busy. This value varies according to the type of printer. Consult your printer documentation for acceptable values for printer throughput.
- Job Errors. Number of job errors. Job errors are typically caused by improper port configuration. Check your port configuration for invalid settings. A printing job instance will increment this count only once, even if it happens multiple times. Also, some monitors do not support this kind of error, causing the counter to remain at 0.

Collecting baseline data is an important step in monitoring all performance counters. For more information about System Monitor, see "Overview of Performance Monitoring" in this book.

Auditing Printing Events

Auditing is a means of tracking a printer's usage. It's possible to specify which groups or users and which actions to audit for a particular printer, as well as audit both successful and failed actions. Windows 2000 stores the data that is generated from auditing a file, which can be viewed and published in various formats using Event Viewer.

To change audit entries, you use the **Audit Entry for *printer_name*** page of the Printer Properties dialog box (Figure 4.16).

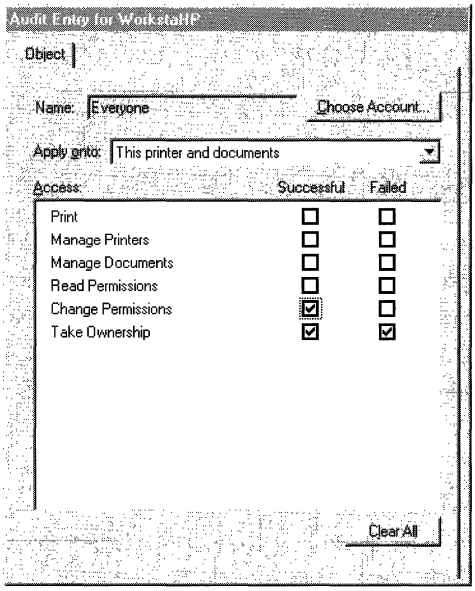


Figure 4.16 Example of an Audit Entry Page

▶ **To add, remove, view, or edit audit entries for a printer**

1. In the **Properties** dialog box for the printer, click the **Security** tab, and then click **Advanced**.
2. Click the **Auditing** tab.

   If the tab is not visible, this means you do not have administrator permissions (Manage Printers, Manage Documents) for the server and you cannot continue.
3. Use the **Add** and **Remove** buttons to specify user names and/or groups to be audited, or the **View/Edit** button to change settings.

   The **Add** and **View/Edit** buttons take you to the **Audit Entry for *printer_name*** page as shown in Figure 4.16.
4. On the **Audit Entry** page, under **Apply onto**, specify whether the auditing should be done by printer, by document, or both.

5. Under **Access**, select check boxes as appropriate to tailor the auditing for the users or groups appearing in the **Name** box:
   - **Successful** means "Audit all successful attempts to perform this action."
   - **Failed** means "Audit all failed attempts to perform this action."
   - **Print**, **Manage Printers**, and **Manage Documents** are the printing permissions. **Read Permissions**, **Change Permissions**, and **Take Ownership** are permissions to control permissions. Table 4.4 shows the associated events that are audited by selecting each permission.
6. To configure another user or group for auditing, click **Choose Account**.
7. When finished, click **OK** to save all your settings.

Note Most printers should not be audited: the Event Log service would fill up with useless information. It is best to limit auditing to select, high-security printers.

Table 4.4 Audit Events Matrix for Printers

	Permission Selected for Auditing					
Event	**Print**	**Manage Documents**	**Manage Printers**	**Read Permissions**	**Change Permissions**	**Take Ownership**
Printing documents	Audited	Not audited	Not audited	Not audited	Not audited	Not audited
Changing document printing preferences	Audited	Not audited	Not audited	Not audited	Not audited	Not audited
Changing document job properties	Not audited	Audited	Not audited	Not audited	Not audited	Not audited
Pausing, restarting, moving, and deleting documents	Not audited	Audited	Not audited	Not audited	Not audited	Not audited
Changing document printing defaults	Not audited	Audited	Audited	Not audited	Not audited	Not audited
Creating a printer share	Not audited	Not audited	Audited	Not audited	Not audited	Not audited

(continued)

Table 4.4 Audit Events Matrix for Printers *(continued)*

Event	Permission Selected for Auditing					
	Print	Manage Documents	Manage Printers	Read Permissions	Change Permissions	Take Ownership
Changing printer properties	Not audited	Not audited	Audited	Not audited	Not audited	Not audited
Deleting a printer	Not audited	Not audited	Audited	Not audited	Not audited	Not audited
Reading printer permissions	Not audited	Not audited	Not audited	Audited	Not audited	Not audited
Changing printer permissions	Not audited	Not audited	Not audited	Not audited	Audited	Not audited
Taking ownership	Not audited	Not audited	Not audited	Not audited	Not audited	Audited

Important For this procedure to work, the **Audit Object Access** option in Group Policy must be set to audit successful attempts, failed attempts, or both. To access this option, click **Computer Configuration**, **Windows Settings**, **Security Settings**, **Local Policies**, and then click **Audit Policy**.

Logging Spooler Events

Another source of valuable information is the spooler, which can provide data about printer traffic, hard disk space, spooler errors, and other printing maintenance issues. Use the following procedure to enable spooler event logging.

▶ **To enable spooler event logging**

1. In the **Printers** dialog box, click the appropriate printer.
2. On the **File** menu, click **Server Properties**.
3. Click the **Advanced** tab.
4. Verify that the entry in the **Spool folder** field is the path you want; if not, type in the correct path.
5. Select any combination of the three check boxes related to event logging to enable the information that you want, and then click **OK**.

Note These check boxes also appear in Group Policy under User Configuration.

For more information about the spooler, see "Server Spooler" earlier in this chapter.

Utilities

Windows 2000 provides several utilities to help you administer your network printing resources.

rundll32

Some enterprise configurations require clients to connect to network printers during an unattended setup. Microsoft facilitates this with the **rundll32** command, which allows printers to be installed from the command line. This is particularly useful when you need to add or remove printers from a group of users by means of a logon script.

The **rundll32** command provides greater control than older supplemental utilities such as Con2prt.exe, which allowed only for the addition and deletion of networked printers. For example, the **rundll32** command allows local printers to be modified and deleted. It also allows you to add printer connections by computer rather than by user as the Add Printer Wizard does. The result is that you can control all aspects of the end user's printing experience by running a local or remote batch file or logon script containing **rundll32** commands.

The case-sensitive syntax for the **rundll32** command is:

rundll32 printui.dll,PrintUIEntry *<options>*

Examples of useful options include:

rundll32 printui.dll,PrintUIEntry /il /c\\computer

This installs a printer for the user logged on a remote computer named *computer* by running the Add Printer Wizard. You can also type:

rundll32 printui.dll,PrintUIEntry /ga /c\\computer /n\\print_server\printer

This adds a connection to *print_server/printer* for all users on a remote computer named *computer*.

The available options for **rundll32** are shown in Figure 4.17

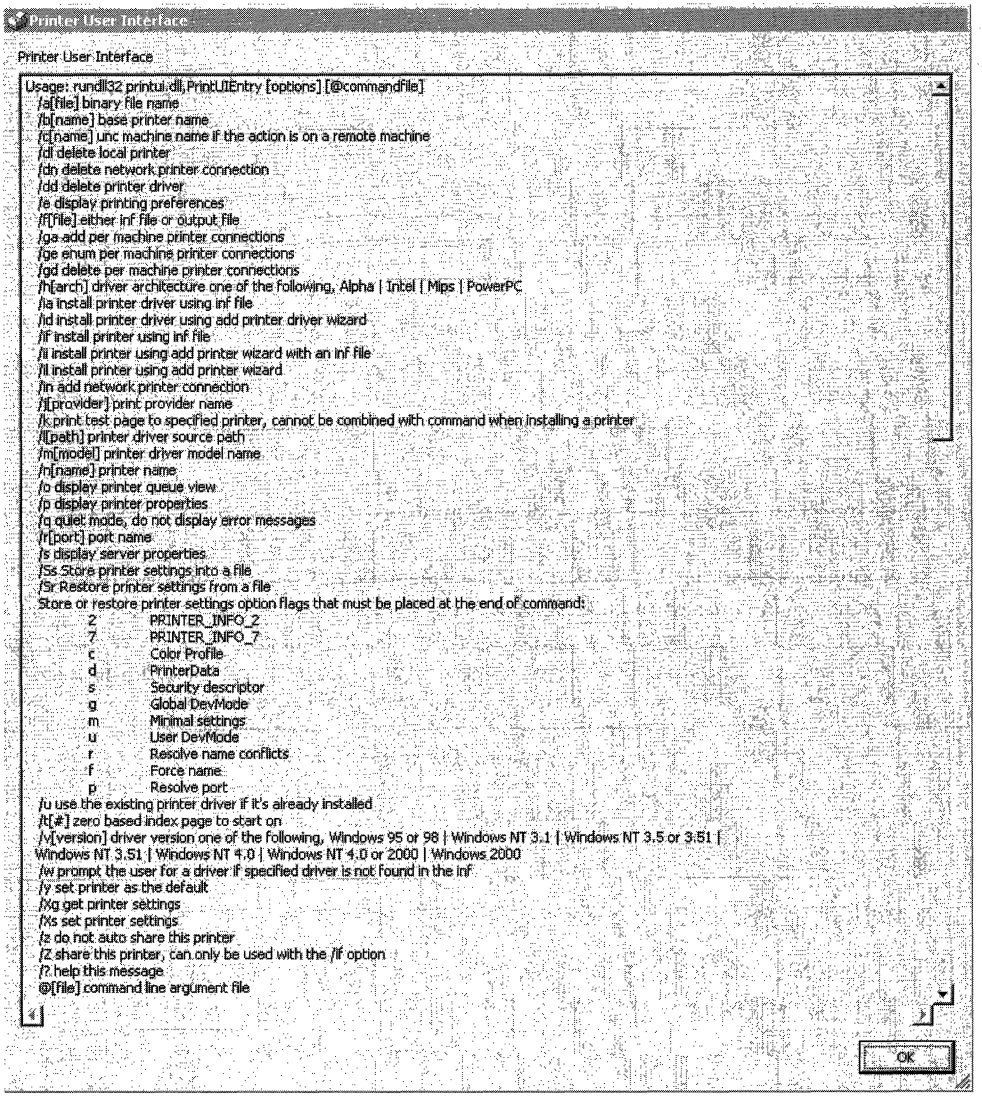


Figure 4.17 Rundll32 Options

PrnAdmin

The PrnAdmin utility lets you use scripts that perform administrative printer functions. You can create your own scripts or modify one of the sample scripts that is provided with PrnAdmin. Either way you can manage both local and remote printers using PrnAdmin.

This tool can perform the following functions:

- Add or delete local or remote printers.
- Add or delete printer connections.
- Add or delete local or remote forms.
- Add or delete ports on local or remote printers.
- Add or delete printer drivers on local or remote printers.
- Enumerate ports, drivers, printers, or forms on local or remote printers.
- Pause, resume, or purge a local or remote printer.
- Configure a local or remote printer.

Note To create scripts, you need to know a scripting language such as Microsoft® Visual Basic or JavaScript or be able to make use of the objects provided by PrnAdmin in any language that supports COM programming.

To install PrnAdmin, type the following command:

regsvr32 [Path]PrnAdmin.dll

To use the PrnAdmin sample scripts shown in Figure 4.18, type:

Cscript prnmgr.vbs

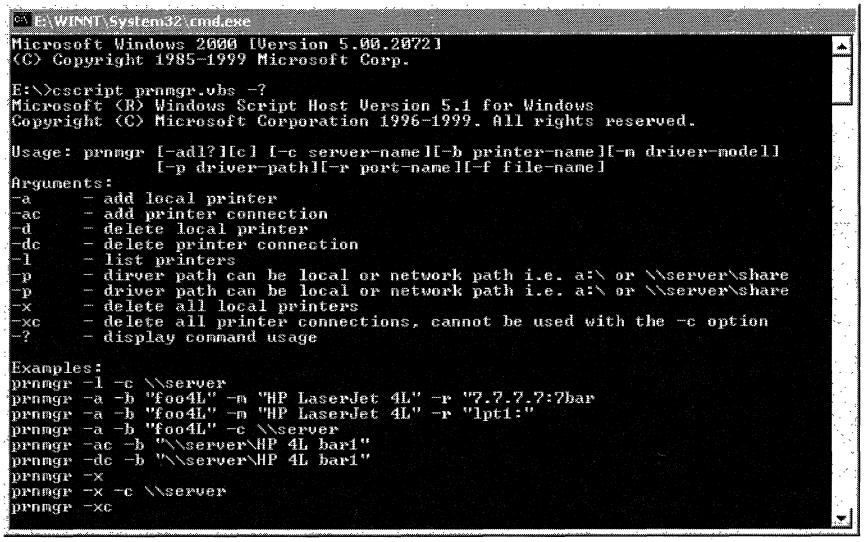
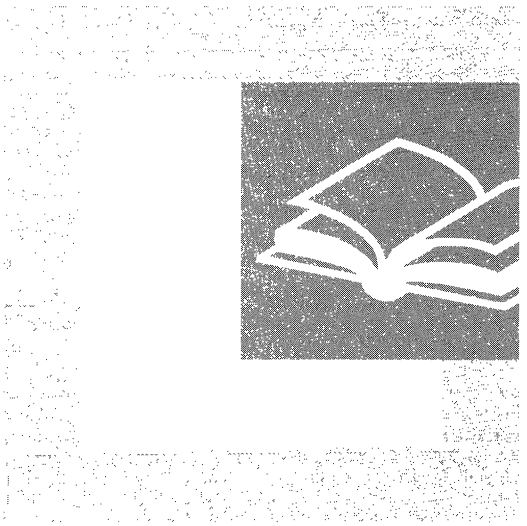


Figure 4.18 PrnAdmin Scripts

PART 2

Performance Monitoring



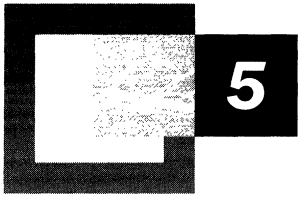
Monitoring server performance involves observing a system both as a whole and as a collection of its individual components. Part 2 describes the tools Windows 2000 provides for assessing system performance, explains how your system uses its hardware resources, and offers strategies for correcting performance problems.

In This Part

Overview of Performance Monitoring 223
Evaluating Memory and Cache Usage 289
Analyzing Processor Activity 325
Examining and Tuning Disk Performance 365
Monitoring Network Performance 393
Measuring Multiprocessor System Activity 429

CHAPTER 5

Overview of Performance Monitoring



Monitoring performance is a necessary part of preventive maintenance for your computer system. Through monitoring, you obtain performance data that is useful in diagnosing system problems and in planning for the growth in demand for system resources.

In This Chapter

Performance Monitoring Concepts 225
Monitoring Tools 232
Starting Your Monitoring Routine 258
Analyzing Monitoring Results 261
Investigating Bottlenecks 265
Troubleshooting Problems with Performance Tools 267
Specific Monitoring Scenarios 272
Monitoring Legacy Applications 278
Integrating the System Monitor Control into Office and Other Applications 283

Related Information in the Resource Kit

- For more information about diagnosing system problems, see "Troubleshooting Strategies" in this book.
- For more information about tools for monitoring performance, see *Microsoft® Windows® 2000 Resource Kit* Tools Help.

Performance Monitoring Concepts

Regular performance monitoring ensures that you always have up-to-date information about how your computer is operating. When you have performance data for your system over a range of activities and loads, you can define a baseline—a range of measurements that represent acceptable performance under typical operating conditions. This baseline provides a reference point that makes it easier to spot problems when they occur. In addition, when you are troubleshooting system problems, performance data gives you information about the behavior of system resources at the time the problem occurs, which is useful in pinpointing the cause. Finally, monitoring system performance provides you with data to project future growth and to plan for how changes in your system configurations might affect future operation. Figure 5.1 shows the sequence for monitoring different system resources.

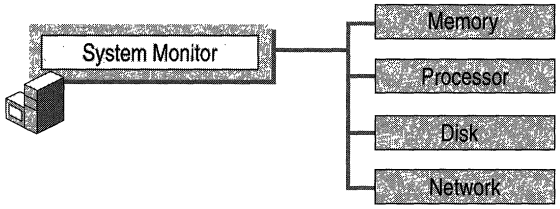


Figure 5.1 Overall Monitoring Sequence

The following sections describe the scope and type of performance data collected, the design of the performance data architecture, and methods of data collection used by the performance tools.

Scope of Performance Data

In general, performance monitoring concentrates on how the operating system and any applications or services that are instrumented for performance-data collection use the resources of the system, such as the disks, memory, processors, and network components. *Throughput*, *queue*, and *response time* are terms that describe resource usage.

Throughput Defined

Throughput is a measure of the work done in a unit of time, typically evaluated from the server side in a client/server environment. Throughput tends to increase as the load increases up to a peak level. It then begins to fall, and a queue might develop. Throughput in an end-to-end system, such as client/server, depends on how each component performs. The slowest point in the system sets the throughput rate for the system as a whole. Often this slow point is referred to as a bottleneck. Performance monitoring tells you where bottlenecks occur in your system. The resource that shows the highest use is often the bottleneck, but not always—it can also mean a resource is successfully handling a lot of activity. As long as no queues develop, there is no bottleneck. Microsoft® Windows® 2000 reports throughput data on resources such as disks and network components.

Queue Defined

A queue can form under a few different circumstances. For example, a queue can develop when requests come in for service by the resource at a faster rate than the resource's throughput, or if requests demand differing, particularly longer, amounts of time from the resource. A queue can also form if requests occur at random intervals—for example, in large batches for a time and then none at all. When a queue becomes long, work is not being handled efficiently, and you might experience delays in response time. Windows 2000 reports queue development on disks, processors, and server work queues, tracking server message block (SMB) calls of the Server service.

Response Time Defined

Response time is the measure of time required to do work from start to finish. In a client/server environment, you typically measure response time on the client side. Response time generally increases as the load increases. You can measure response time by dividing the queue length for the resource over the resource throughput. As an alternative, the new trace log feature in the Windows 2000 performance tools allows you to track units of work from start to finish in order to determine response times.

The following sections describe how performance monitoring tools enable users to collect data about the throughput, queue formation, and response time of different system resources.

Data Collection Architecture

Windows 2000 collects data about system resources, such as disks, memory, processors, and network components. In addition, applications and services that you might be running on your system can also perform data collection. By default, the operating system obtains performance data for system resources using the registry.

When you use performance tools to access registry functions for performance data, the system collects the data from the appropriate system object managers, such as the Memory Manager, the input/output (I/O) subsystem, and so forth.

As an option, Windows 2000 supports collecting data using the Windows Management Infrastructure (WMI) interface by means of the following command syntax typed at the Windows command prompt:

perfmon / WMI

In addition to several of the system performance counter DLLs, the operating system installs managed object files (MOFs) for data collection using WMI instead of the registry. These files reside in System32\Wbem\Mof. The Windows Management service must be running on the monitoring and monitored computer (if different) in order to obtain data using WMI.

Windows 2000 defines the performance data it collects in terms of objects, counters, and instances. Think of a performance object as any resource, application, or service that you can measure. The following sections describe these entities in more detail.

Performance Objects

By default, Windows 2000 installs numerous performance objects corresponding to hardware or other resources in the system. The following table shows the default performance objects installed by the operating system on a Microsoft® Windows® 2000 Professional installation.

Table 5.1 Windows 2000 Performance Objects

Object name	Description
ACS/RSVP Service	Reports activity of the Quality of Service (QoS) Admission Control Service used to manage the priority use of network resources (bandwidth) at the subnet level.
Browser	Reports activity of the Browser service in Microsoft® Windows® 2000 Server that lists computers sharing resources in a domain and other domain and workgroup names across the wide area network (WAN). Windows 2000 provides the Browser service for backward compatibility with clients that are running Microsoft® Windows® 95, Microsoft® Windows® 98, Microsoft® Windows® 3.*x*, and Microsoft® Windows NT®.
Cache	Reports activity for the file system cache, an area of physical memory that holds recently used data.

(continued)

Table 5.1 Windows 2000 Performance Objects *(continued)*

Object name	Description
Distributed Transaction Coordinator	Reports statistics about activity of the Microsoft Distributed Transaction Coordinator, a part of Component Services (formerly known as Transaction Server) used to coordinate two-phase transactions by Message Queuing.
HTTP Indexing Service	Reports statistics regarding queries run by the Indexing service, a service that builds and maintains catalogs of the contents of local and remote disk drives to support powerful document-search capabilities.
IAS Accounting Clients	Reports activity of the Internet Authentication Service (IAS) as it centrally manages remote client accounting (usage).
IAS Accounting Servers	Reports activity of the Internet Authentication Service (IAS) as it centrally manages remote server accounting (usage).
IAS Authentication Clients	Reports activity of the Internet Authentication Service (IAS) as it centrally manages remote client authentication.
IAS Authentication Servers	Reports activity of the Internet Authentication Service (IAS) as it centrally manages remote server authentication.
ICMP	Reports the rates at which Internet Control Message Protocol (ICMP) messages are sent and received by using the ICMP protocol, which provides error correction and other packet information.
Indexing Service	Reports statistics pertaining to the creation of indexes and the merging of indexes by the Indexing service. Indexing service indexes documents and document properties on your disks and stores the information in a catalog. You can use Indexing service to search for documents, either with the Search command on the Start menu or with a Web browser.
Indexing Service Filter	Reports filtering activity of the Indexing service. Indexing service indexes documents and document properties on your disks and stores the information in a catalog. You can use Indexing service to search for documents, either with the Search command on the Start menu or with a Web browser.
IP	Reports activity at the Internet Protocol (IP) layer of Transmission Control Protocol/Internet Protocol (TCP/IP).
Job Object	Reports the accounting and processor usage data collected by each active, named job object.

(continued)

Table 5.1 Windows 2000 Performance Objects *(continued)*

Object name	Description
Job Object Details	Reports detailed performance information about the active processes that make up a job object.
Logical Disk	Reports activity and usage of disk partitions and volumes. Use **diskperf -y** to enable disk counters and **diskperf -n** to disable them. To specify the type of counters you want to activate, include **d** for physical disk drives and **v** for logical disk drives or storage volumes. When the operating system starts up, it automatically sets the **diskperf** command with the **-yd** switch to activate physical disk counters. Type **diskperf -yv** to activate logical disk counters. For more information about using the **diskperf** command, type **diskperf -?** at the command prompt.
Memory	Reports usage of random access memory (RAM) used to store code and data.
NBT Connection	Reports the rate at which bytes are sent and received over connections that use the NetBT protocol, which provides NetBIOS support for the TCP/IP protocol between the local computer and a remote computer.
Network Interface	Reports rates at which bytes and packets are sent and received over a TCP/IP connection by means of the network adapters. Typically the first instance of the Network Interface object (Instance 1) that you see in System Monitor represents the loopback address; however, sometimes the loopback address does not appear. The loopback address is a local path through the protocol driver and the network adapter. All other instances represent installed network adapters (WAN interfaces, remote access modems, and so forth).
Objects	Reports data about system software objects, such as events, and so on.
Paging File	Reports usage of the paging file, used to back up virtual memory allocations.
Physical Disk	Reports usage of hard disks and redundant array of independent disks (RAID) devices.
Print Queue	Reports statistics for print jobs in the queue of the print server. New for Windows 2000.
Process	Reports activity of the process, which is a software object that represents a running program.
Processor	Reports activity of the processor (also called the CPU), the part of your computer hardware that carries out program instructions.
Redirector	Reports activity for the Redirector file system, which diverts file requests to network servers.

(continued)

Table 5.1 Windows 2000 Performance Objects *(continued)*

Object name	Description
Server	Reports activity for the Server file system, which responds to file requests from network clients.
Server Work Queues	Reports the length of queues and objects in the queues for the Server service.
System	Reports statistics for systemwide counters that track file operations, processor time, and so on.
TCP	Reports the rates at which TCP segments are sent and received using the Transmission Control Protocol (TCP).
Telephony	Reports activity for telephony devices and connections.
Thread	Reports activity for a thread (the part of a process that uses the processor).
UDP	Reports the rates at which User Datagram Protocol (UDP) datagrams are sent and received using UDP.

If you are running Windows 2000 Server, Setup automatically installs the Active Server Pages, FTP Service, Internet Information Services Global, and Web Service objects for use with Internet Information Service. In addition, Windows 2000 Server Setup installs the SMTP Server and SMTP NTFS Store Driver objects. Depending on the services you have configured, your system might provide several additional objects, such as the NTDS object, which reports activity of the Active Directory™ directory service, and the DNS object, which reports performance statistics for the Domain Name System (DNS) service. For detailed information about these and other performance objects, see the Windows 2000 Performance Counters Reference on the *Microsoft® Windows® 2000 Resource Kit* companion CD.

For information about writing applications that install performance objects that can be integrated with the performance tools, see the Software Development Kit (SDK) documentation in the MSDN™ Library at http://windows.microsoft.com/windows2000/reskit/webresources.

Performance Counters and Instances

Each object has counters that are used to measure various aspects of performance, such as transfer rates for disks or the amount of processor time consumed for processors. Each object has at least one instance, which is a unique copy of a particular object type, though not all object types support multiple instances. This chapter and the following chapters describe objects, counters, and instances using the following syntax:

\\Computer_name\Object(ParentInstance/ObjectInstance#InstanceIndex)\Counter

The *Computer_name* portion is optional; if you do not include a computer name, the default is the local computer.

Note that the syntax includes a parent instance, object instance, and an instance index. This applies, for example, if the object has multiple instances and these instances might be identifiable by name or number, as defined by the counter developer. (Typically, internal system counters use numeric instance indexes.)

For example, if you are monitoring threads of the Microsoft Windows Explorer process, track the Windows Explorer instance of the Thread object (Windows Explorer would be the parent instance), and then each thread running Windows Explorer (these threads are child instances). The instance index allows you to track these child instances. The instance index for the thread you want might be 0, 1, and so on, for each thread, preceded by the number sign (#). The operating system configures System Monitor properties to display duplicate instances by default. Instance index 0 is hidden; numbering of additional instances starts with 1. You cannot monitor multiple instances of the same process unless you display instance indexes.

An instance called _Total is available on most objects, and represents a sum of the values for all instances of the object for a specific counter.

Data Collection and Reporting

Depending on the tools used, you can configure data collection to occur almost immediately or according to a predefined schedule. Performance data reported is sampled, meaning that data is collected periodically rather than traced, whereby data is obtained as events occur. This collection method has the advantage of keeping overhead low, but it might occasionally overestimate or underestimate values when activity falls outside the sampling interval.

If you want more precise performance data, use event tracing, a new capability in Windows 2000. Event tracing can measure activity as it happens, eliminating the inaccuracies of sampling and making it possible to correlate resource usage such as page faults, disk input/output (I/O), and processor time with workload that can include threads, processes, or transactions. This capability supplements counter-based monitoring methods. You can configure trace logs for providers you have or for the built-in system provider that runs traces for the Windows kernel provider using trace logs in **Performance Logs and Alerts**. Because running trace logs of page faults and file I/O data incurs some performance overhead, log this data only for brief periods. Note that an additional program is required to parse the log output into readable form. Developers can create such a tool using APIs provided in the Platform Software Development Kit.

For information about writing a trace provider, see the Platform Software Development Kit (SDK) documentation in the MSDN Library at http://windows.microsoft.com/windows2000/reskit/webresources.

Depending on how a counter is defined, its values might be reported in one of the following ways:

- Instantaneous counters, which might have names containing the word "current," display the most recent measurement. Be aware that instantaneous counters might not provide meaningful data unless you have a steady workload.
- Averaging counters, which typically have names that include "per second" or "percent," measure a value over time and display the average of the last two measurements over the period between samples. (Because counters are never cleared, this is actually an average of the difference between the measurements.) When you start these counters, you must wait for the second measurement to be taken before any values are displayed. For example, Memory\Pages/sec shows the number of pages read over the sample interval, divided by the number of seconds in the interval.

  For averaging counters, the sampling method can result in a slight delay in displaying values as data is collected and computed. In addition, after a single large value is reported, causing spikes in a performance graph, averaging counter values can be artificially high for a while until the average starts to reflect more recent steady-state activity.

Windows 2000 supports other types of counters, such as percentage, difference, and text. Difference counters display the change in value between the last two measurements. By default, counters that display their values as percentages cannot exceed 100 percent.

For information about the preceding generic counter types and their specific subtypes, see the Windows 2000 Performance Counter Reference (Counters.chm) on the *Windows 2000 Resource Kit* companion CD.

Monitoring Tools

The primary monitoring tools in Windows 2000 are the Performance console and Task Manager. Task Manager offers an immediate overview of system activity and performance, and the Performance console provides detailed information that can be used for troubleshooting and bottleneck analysis. The Performance console hosts two tools: System Monitor, and Performance Logs and Alerts. The chapters in this part of the *Server Operations Guide* concentrate on using the Performance console. The following sections describe the utilities that are installed with the operating system.

You can start the Performance console from the **Administrative Tools** menu. To use the **Administrative Tools** menu in Windows 2000 Professional:

Add the **Administrative Tools** menu to the **Programs** menu.

Or, use the **Administrative Tools** menu in Control Panel

Administrative Tools are provided on the **Start** menu by default in Windows 2000 Server.

▶ **To add the Administrative Tools menu to the Programs menu on a computer running Windows 2000 Professional**

1. On the **Start** menu, point to **Settings**, and then click **Taskbar & Start Menu**.
2. Click the **Start Menu Options** tab. Under **Start Menu Settings**, select the **Display Administrative Tools** check box and then click **OK**.
3. Click the **Start** button again, point to **Programs**, and then click **Administrative Tools**.

▶ **To use the Administrative Tools menu in Control Panel**

1. Double-click **My Computer** on the Windows 2000 desktop.
2. Under **My Computer**, double-click **Control Panel**.
3. In Control Panel, double-click **Administrative Tools**.

System Monitor

System Monitor in Windows 2000 extends the functionality provided by Performance Monitor, which shipped in Microsoft® Windows® NT versions 4.0 and Windows 3.51. Features of System Monitor include the following:

- The graph display is much more flexible and configurable. You can modify many attributes of the display, including changing font and color, adding borders, and so on.
- Counter configuration is simplified. You can now copy counter paths and settings from the System Monitor display to the Clipboard and paste counter paths from Web pages or other sources into the System Monitor display.
- Graphs can be printed when performance displays are saved as HTML files using the **Save As** command on the shortcut menu. In addition, you can save reports as tab-separated files (for use with Microsoft Excel) by means of the **Save As** command. To use the shortcut menu, right-click the details pane of System Monitor.

- System Monitor is portable. Because System Monitor is hosted in the Microsoft Management Console (MMC), you can save a console file containing a group of counters that you want to monitor—you can install it on any other computer and be able to monitor the same types of data on that computer. This is useful in monitoring other systems that you administer.
- The functionality of System Monitor chart, histogram, and report views is provided by an ActiveX® control (Sysmon.ocx). This design gives a user the flexibility of including the control in an HTML page (PerfManager on the *Windows 2000 Resource Kit* companion CD is an example) or of programming the control into a Microsoft® Office or Microsoft® Visual Basic® application, as described in "Integrating the System Monitor Control into Office and Other Applications" later in this chapter. In most cases, you work with the control's functionality in the form it is presented in Perfmon.msc, the Microsoft Management Console component that hosts the performance tools.

For information about logging and alert capabilities of the Performance console, see "Performance Logs and Alerts" later in this chapter. Windows NT 4.0 Performance Monitor is provided under the name Perfmon4.exe on the Windows 2000 Resource Kit companion CD. Typing **perfmon.exe** at the command prompt causes the system to launch System Monitor, not Performance Monitor.

Starting System Monitor

If you are running Windows 2000 Professional, you can start the Performance console as follows:

- On the **Administrative Tools** menu, click **Performance**.

  Or, on the **Start** menu, click **Run**, type **perfmon.msc**, and then click **OK**.

  Or, type **perfmon.msc** at the Windows command prompt.

When you start the Performance console, a blank System Monitor graph appears. **Performance Logs and Alerts** appears beneath System Monitor in the console tree, as shown in Figure 5.2.

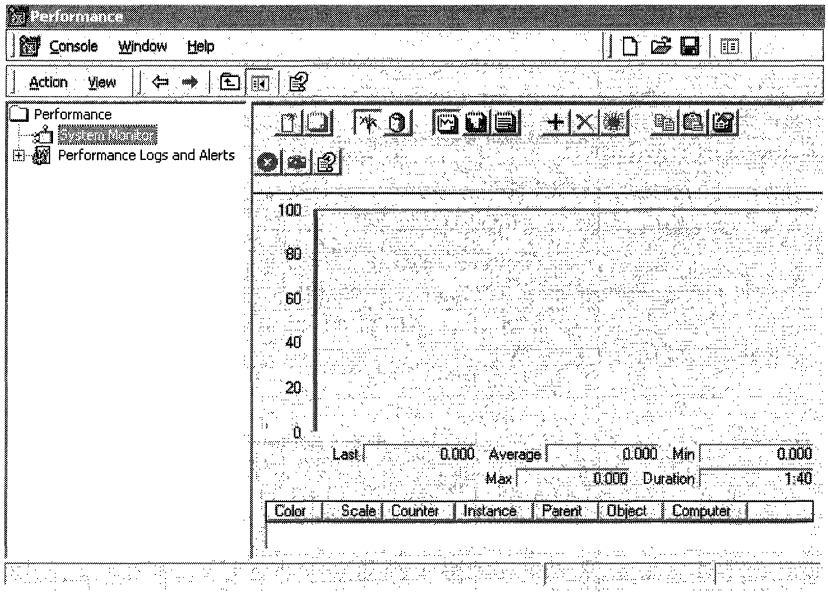


Figure 5.2 Performance Console

The following section describes the user interface for System Monitor and provides tips on how to use it.

Working with System Monitor

With System Monitor you can create graphs, bar charts (histograms), and text reports of performance counter data. System Monitor is designed for short-term viewing of data, troubleshooting, and diagnosis.

The System Monitor display consists of the following elements:

- An optional toolbar with capabilities such as copying and pasting counters, clearing counters, adding counters, and so on. The toolbar buttons provide the quickest way of configuring the monitoring display, but you can also use a shortcut menu to add counters and configure properties.
- The area where counter values are displayed. You can vary the line style, width, and color of these lines. You can also change the color of the window and of the chart within the window.
- A legend showing the selected counters and associated data such as the computer name, parent object, and instances.

- A value bar, where you see the last, minimum, maximum, and average values for the counter that is currently selected. The value bar also shows a **Duration** value that indicates the total elapsed time displayed in the graph (based on the update interval).
- A timer bar that moves across the graph indicates the passing of each update interval. Regardless of the update interval, the view shows up to 100 samples. System Monitor compresses log data as necessary to fit it in the display. For example, if there are 1,000 samples, the display might show every tenth sample.

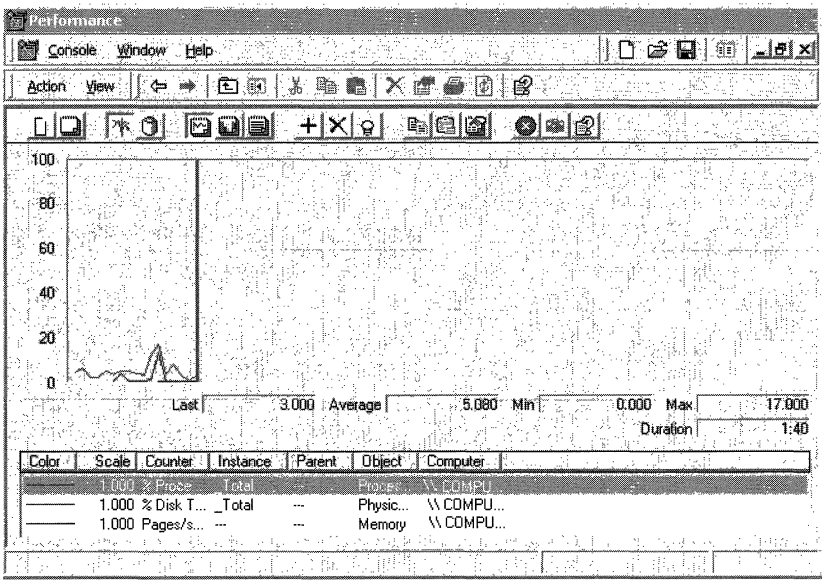
You can configure System Monitor using either the toolbar or a shortcut menu. Using the shortcut menu offers more control and flexibility in configuring the display. The following sections describe these different configuration methods. To see procedures and a brief overview of System Monitor, click **Help** on the System Monitor toolbar.

Using the Toolbar

The toolbar is displayed by default. Using the toolbar, you can configure the following options:

- **Type of display.** Use the **View Chart**, **View Histogram**, or **View Report** button.

  Figure 5.3 shows the different display options.



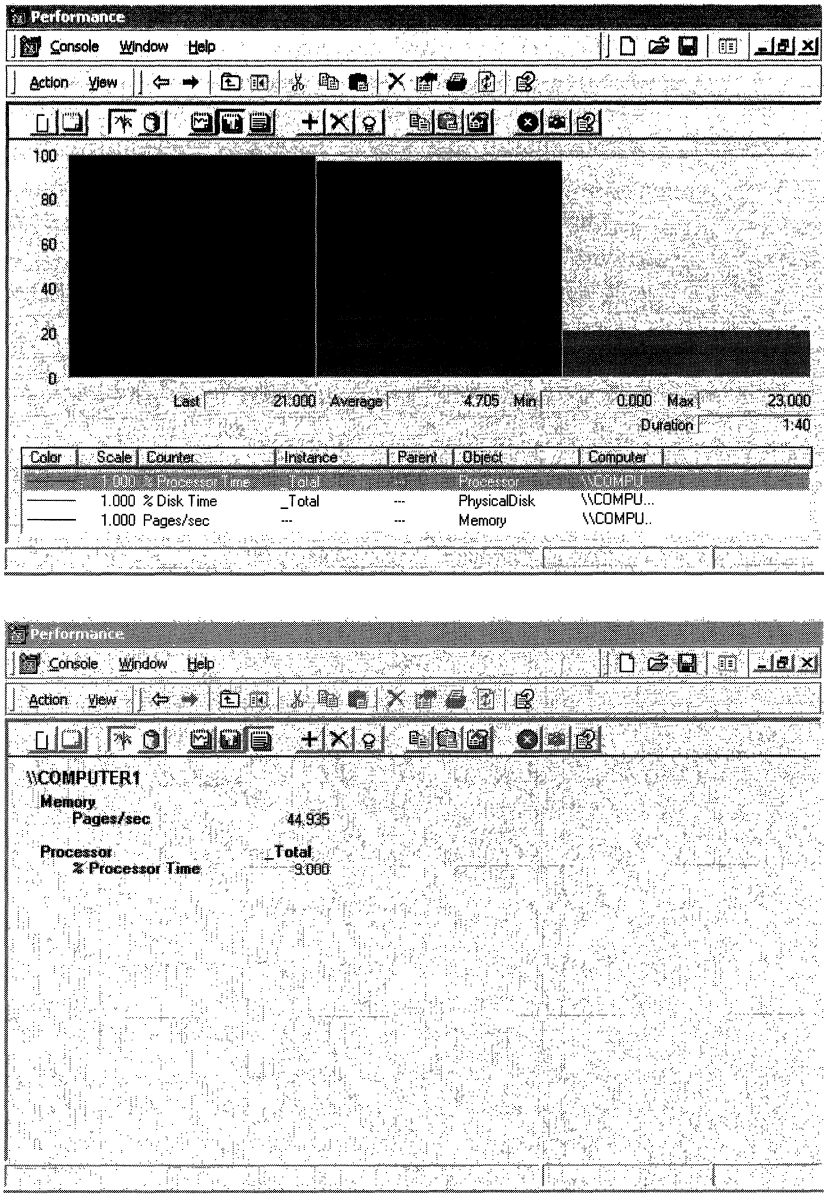


Figure 5.3 Display Type Options for System Monitor

Histograms and reports are useful for simplifying graphs with multiple counters. However, they display only a single value, so they are recommended only when you are charting current activity and watching the graphs as they change. When you are reviewing data logged over time, line graphs are much more informative so that trends can be identified.

- **Data source.** Click the **View Current Activity** button for real-time data or the **View Log File Data** button for data from either a completed or a currently running log.
- **Counters.** Use the **Add** or **Delete** buttons as needed. You can also use the **New Counter Set** button to reset the display and select new counters. Clicking the **Add** button displays the **Add Counters** dialog box, as shown in Figure 5.4. You can also press the DEL key to delete a counter that is selected in the legend.

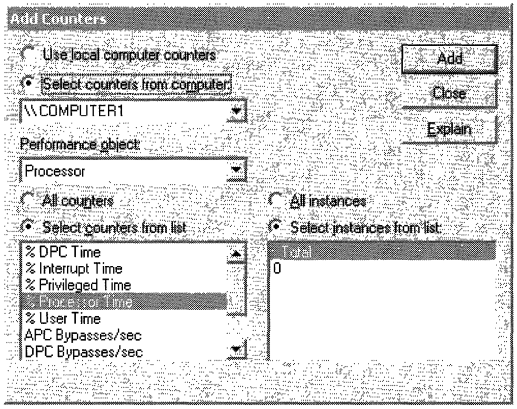


Figure 5.4 Add Counters Dialog Box

- **Data updates.** Click **Clear Display** to clear the displayed data and obtain a fresh data sample for existing counters. To suspend data collection, click **Freeze Display**. Use the **Update Data** button to resume collection.
- **Highlighting chart or histogram data.** To accentuate the line or bar for a selected counter with white (default) or black (for light backgrounds), click **Highlight** on the toolbar.
- **Importing or exporting counter settings.** To save the displayed configuration to the Clipboard for insertion into a Web page, click **Copy Properties**. To import counter settings from the Clipboard into the current System Monitor display, click **Paste Counter List**.
- **Configuring other System Monitor properties.** To access colors, fonts, or other settings that have no corresponding button on the toolbar, click **Properties**.

Using the Shortcut Menu

When you right-click the System Monitor display, a shortcut menu appears with the following options:

- **Add Counters.** Use this option in the same way you use the **Add** button in the toolbar.
- **Save As.** Use this if you want to save the current display configuration under a new name. If you click **Save** on the **Console** menu, the current settings are stored, overwriting the blank version of Perfmon.msc installed by Windows 2000 Setup and altering the default appearance of the tool.
- **Properties.** Click this button to access the five properties tabs that provide options for controlling all aspects of System Monitor data collection and display. The **General** properties tab appears by default, as shown in Figure 5.5.

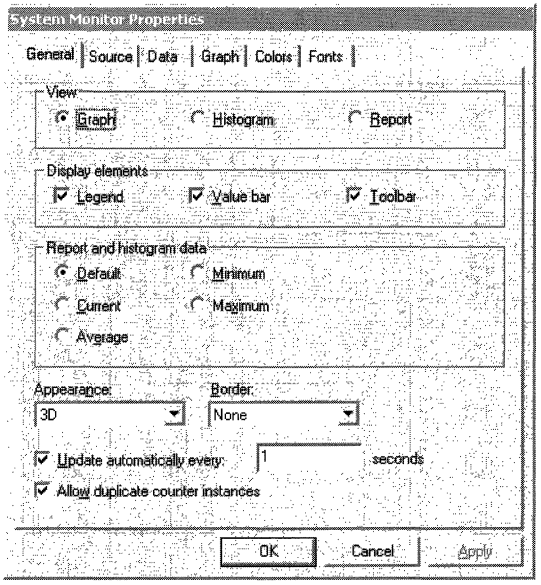


Figure 5.5 General Tab

Many properties can be configured from the toolbar, but some are only configurable using **System Monitor Properties**. The following table lists property tabs alphabetically by name, along with the attributes they control.

Table 5.2 System Monitor Properties

Use this tab	To add or change this
Colors	Background color of results pane surrounding the chart area, color of chart data-display area.
	You can choose each color from a palette (in the **Property Name** list box) or you can base the colors on system colors (screen elements) defined using the **Display** icon in Control Panel. When using the palette, note the following: ▪ **BackColorCtl** refers to the area surrounding the chart. ▪ **BackColor** refers to the chart data-display area. ▪ **ForeColor** refers to the color of the text in the display and legend.
	Grid color, timer bar color.
Data	Color, width, style, or chart line.
	Notice that defining a nondefault line width limits the line styles that are available. Styles can be selected only when you are using the default line width.
	Scale of counter data values.
	Counter values can be scaled exponentially from .0000001 to 1000000. You might want to adjust the counter scale settings to enhance the visibility of counter data in the chart. Changing the scale does not affect the statistics displayed in the value bar.
	Objects, counters, and instances.
Fonts	Font type, size, and style.
General	View type: chart, histogram, or report.
	Update frequency and manual or periodic sampling.
	Histogram or report value type (choose between minimum, maximum, average values for the one displayed in a report view).
	Using report value types other than **Current** when monitoring real-time data incurs substantial overhead because of the need to make calculations across all samples for each value displayed.
	Display of counter legend.
	Display of last, minimum, and maximum values for a selected counter (the value bar).
	Border style, appearance of the entire control. You can include or omit a border, or configure three-dimensional or flat effects for the window.
	Display of toolbar.
	Display of instance indexes (for monitoring multiple instances of a counter).
	The first instance (instance number 0) displays no index; System Monitor numbers subsequent instances starting with 1.

(continued)

Table 5.2 System Monitor Properties *(continued)*

Use this tab	To add or change this
Graph	Title of graph.
	Label on value axis, vertical or horizontal grid lines, and upper and lower limits of graph axes.
Source	Source of data displayed: current data input to the graph, current or archived data input from a log.
	Time range for a log and view time range.

Getting the Most from System Monitor

Windows 2000 Server online Help for System Monitor explains how to perform common tasks. The following list supplements the information provided in online Help to enable you to use System Monitor more effectively.

- **Print data.** You can print performance data in several ways:
  - Copy the current view to the Clipboard (by pressing ALT+PRINT SCREEN), start a paint program, paste in the image from the Clipboard, and then print it.
  - Add the System Monitor control to a Microsoft Office application such as Microsoft® Word or Microsoft® Excel, configure it to display data, and then print from that program. For information about this process, see "Integrating the System Monitor Control into Office and Other Applications" later in this chapter.
  - Save the System Monitor control as an HTML file by right-clicking the details pane of System Monitor and typing a file name for the HTML file to be created. You can then open the HTML file and print it from Microsoft® Internet Explorer or another program.
  - Import a log file in comma-separated (.csv) or tab-separated (.tsv) format into an Excel spreadsheet and print from that application.
- **Learn about individual counters.** When adding counters, if you click **Explain** in the **Add Counters** dialog box for System Monitor or **Performance Logs and Alerts**, you can view counter descriptions.
- **Vary the data displayed in a report.** By default, reports display only one value for each counter. This is current data if the data source is real-time activity, or averaged data if the source is a log. However, using the General properties tab, you can configure the report display to show different values, such as the maximum, minimum, and so on. Notice that monitoring the nondefault value for a report can increase performance-monitoring overhead.
- **Arrange items in the legend.** To sort entries in ascending or descending order for that category, click **Object**, **Counter**, **Instance**, or **Computer** in the counter legend. For example, to sort all counters by name, click **Counter**.

- **Select a group of counters or counter instances to monitor.**
  - To select all counters or instances, click **All counters** or **All instances**.
  - To select specific counters or instances, click **Select counters from the list** or **Select instances from the list**.
  - To monitor a group of consecutive counters or instances in a list box, hold down the SHIFT key and scroll down through the items in the list box.
  - To select multiple, nonconsecutive counters or instances, select the item and press CTRL.

Important Monitoring large numbers of counters can incur a high amount of overhead, even to the point of making the system unresponsive to keyboard or mouse input. To reduce this burden, display data in report view when collecting from large numbers of counters or direct data to a binary log, and view the data in System Monitor as it is being written to the log.

- **Simplify detailed graphs.** You can maintain two separate instances of System Monitor if you want to monitor a large number of counters while keeping each graph relatively simple and uncluttered. It is also a good way to compare data from different sources.
- **Track totals for all instances of a counter.** Instead of monitoring individual instances for a selected counter, you can instead use the _Total instance, which sums all instances' values and reports them in System Monitor.
- **Pinpoint a specific counter from lines in a graph.** To match a line in a graph with the counter for which it is charting values, double-click a position in the line. If chart lines are close together, try to find a point in the graph where they diverge.
- **Accentuate a specific counter's data.** To draw attention to a particular counter's data, use the highlighting feature. To do so, press CTRL+H or click **Highlight** on the toolbar. For the counter selected, a thick line replaces the colored chart line. For white or light-colored backgrounds (defined by the BackColor property), this line is black; for other backgrounds, this line is white.
- **View data from a running log.** If you are working with a log file that is currently collecting data, you need to click the **Select Time Range** button and keep moving the **Time Range** bar to the right to update the display with new samples.
- **Use Windows NT 4.0 settings files.** You can display legacy alert, report, chart, and log settings files in System Monitor by using the following command at the command prompt:

  perfmon.exe *settings_file_name*

When you open one of these settings files, the system temporarily converts the file for use with System Monitor but discards the converted version after the console starts. If you want to save the settings file for permanent use with System Monitor, type the following command:

perfmon.exe /HTMLFILE:*new_file_name settings_file_name*

Performance Logs and Alerts

Performance Logs and Alerts, a service in Windows 2000, improves the logging and alert capabilities that were provided in Windows NT 4.0. Logging is used for detailed analysis and record-keeping purposes. Retaining and analyzing log data collected over a period of several months can be helpful for capacity and upgrade planning.

Windows 2000 provides two types of performance-related logs—counter logs and trace logs—and an alerting function. The following list describes these new or enhanced tools:

- Performance Logs and Alerts replaces Performance Data Log in the Windows NT Server 4.0 Resource Kit. As a result, data collection occurs regardless of whether any user is logged on to the computer.
- In Windows 2000, counter logs record sampled data about hardware resources and system services based on performance objects and counters in the same manner as System Monitor. When a counter log has been started, the Performance Logs and Alerts service obtains data from the system when the update interval has elapsed.
- Trace logs collect event traces that measure performance statistics associated with events such as disk and file I/O, page faults, or thread activity. When the event occurs, a data provider designed to track these events sends the data to the Performance Logs and Alerts service. The data is measured from start to finish, rather than sampled in the manner of System Monitor. The built-in Windows 2000 kernel trace data provider supports tracing system data; if other data providers are available, developers can configure logs with those providers as appropriate. A parsing tool is required to interpret the trace log output. Developers can create such a tool using APIs provided in the Platform Software Development Kit.
- With the alerting function, you can define a counter value that will trigger actions such as sending a network message, running a program, or starting a log. Alerts are useful if you are not actively monitoring a particular counter threshold value but want to be notified when it exceeds or falls below a specified value so that you can investigate and determine the cause of the change. You might want to set alerts based on established performance baseline values for your system. For information about establishing a baseline, see "Starting Your Monitoring Routine" later in this chapter.

- Viewing logged data is easier and more convenient. Counter logs can be viewed in System Monitor as they are collecting data as well as after data collection has stopped. Data in counter logs can be saved as comma-separated or tab-separated files that are easily viewed with Excel.
- Logs can be circular—that is, recording data until they achieve a user-defined size limit and then starting over. Alternatively, linear logs collect data according to user-defined parameters such as: run for a specified length of time, stop when that parameter is met, and start a new log. A binary file format can also be defined for logging intermittent data (such as for a process that is not running when you start the log but that begins and ends during the logged interval).
- You can save log settings to an HTML file or you can import settings from an HTML page to create new logs. When exported, the resulting HTML page hosts the System Monitor control, an ActiveX control that provides the performance monitoring user interface. If you open this page, you can dynamically observe, from a System Monitor view, the same counters you configured in the log. When imported, a new log or alert is created, based on the settings in the HTML page. This is a convenient way to insert the same settings into both a log and an alert, if appropriate.
- Configuring logs and alerts is flexible and easy to manage. Users can manage multiple logging sessions from a single console window. For each log, users can start and stop logging either manually, on demand, or automatically, at scheduled times or based on the elapsed time or the current file size. Users can also specify automatic naming schemes and stipulate that a program be run when a log is stopped.

Starting Performance Logs and Alerts

In Windows 2000 Professional, the Performance Logs and Alerts component is available in the Performance console and in the Computer Management console. The following procedures describe how to open the component from these locations.

Note This procedure assumes that you have added the **Administrative Tools** option to your **Programs** menu as described in "System Monitor" earlier in this chapter.

▶ **To start Performance Logs and Alerts from the Performance console**

1. Click **Start**, point to **Programs**, and then click **Administrative Tools**.
2. Click **Performance**.

3. Double-click **Performance Logs and Alerts** to display the available tools.

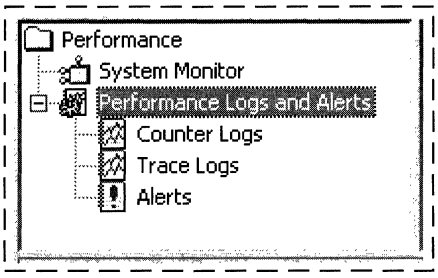


Figure 5.6 Performance Logs and Alerts Console Tree

Working with Logs and Alerts

To begin configuring logs and alerts, click the name of the tool to select it. If any logs or alerts have previously been defined, they will appear in the appropriate node of the details pane. A sample settings file for a counter log named System Overview is included with Windows 2000. You can use this file to see some basic system data such as memory, disk, and processor activity. For information about the types of data to monitor in your own configuration, see "Starting Your Monitoring Routine" later in this chapter.

Right-click in the details pane to create a new log or alert. You can do this in a new file or you can use settings from an existing HTML file as a template.

Note You must have Full Control access to a subkey in the registry in order to create or modify a log configuration. (The subkey is HKEY_CURRENT_MACHINE\SYSTEM \CurrentControlSet\Services\SysmonLog\Log_Queries.) In general, administrators have this access by default. Administrators can grant access to users by using the Security menu in Regedt32.exe. To run the Performance Logs and Alerts service, you must have the right to start or otherwise configure services on the system. Administrators have this right by default and can grant it to users by using Group Policy. For information about starting and using Group Policy, see Windows 2000 Server Help.

You are prompted to name your log or alert and then to define properties. Figure 5.7 is an illustration of the **General** properties tab for a counter log.

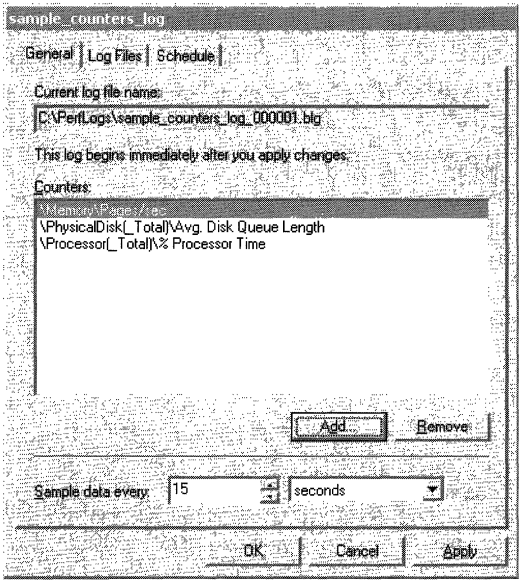


Figure 5.7 General Properties Tab for a Counter Log

If you are configuring a counter log or an alert, use the **Add Counters** dialog box to specify objects, counters, instances, and updating. If you are configuring a trace log, use the **General** property tab shown in Figure 5.8.

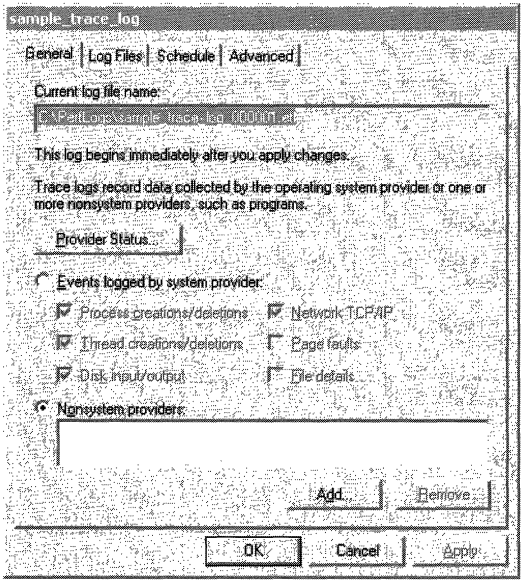


Figure 5.8 General Properties Tab for Trace Log

Each tool offers some unique properties. The ability to configure scheduling is common to logs and alerts, but some options might not be available for all tools. Table 5.3 describes the options available in each tool and the property tab to use to configure it.

Table 5.3 Summary of Log and Alert Properties

For this feature	Use this tab	To configure these settings	Notes
Alerts	General	Counters, sample interval, alert threshold, and alert comment	
	Action	Actions to take when an event occurs	Examples of actions for an alert include running a program, sending a message, starting a counter log, and updating the event log.
	Schedule	Start and stop parameters for alerts	Automated restart is not available if you configure the alert to stop manually. You might need to update the Performance Logs and Alerts service properties if you opt to run a program that displays to the screen after the system triggers an alert. Use **Services** under **Services and Applications** in **Computer Management** for this purpose.
Counter Logs	General	Counter log counters and sample interval	
	Log Files	File type, file size limits, path and name, and automatic naming parameters	Counter logs can be defined as comma-separated or tab-separated text files, or as binary linear or circular files.

(continued)

Table 5.3 Summary of Log and Alert Properties ***(continued)***

For this feature	Use this tab	To configure these settings	Notes
Counter Logs	Schedule	Manual or automated start and stop methods and schedule	Counter logs can be defined as comma-separated or tab-separated text files, or as binary linear or circular files. You can specify that the log stop when the log file is full. You cannot configure the service to automatically restart or to run a program if a log is configured to stop manually. You cannot configure a log to stop when full if the file is configured on the **Log Files** tab to grow to a maximum size limit.
Trace Logs	General	Trace log providers and events to log	You cannot configure the service to automatically restart if a log is configured to stop manually. You can have only one system trace log running at a time. You cannot enable multiple providers simultaneously. To obtain disk input/output data from the system provider, you must also select **File details**.
	Log Files	Trace log comment, file type, path and name, and automatic naming parameters	Only two types of trace logs are available: circular and sequential.
	Schedule	Start and stop parameters for a trace log	You cannot configure the service to automatically restart or to run a program if a log is configured to stop manually.
	Advanced	Trace log buffer size, limits, and transfer interval (periodic flushing)	

To start or stop a log or alert, right-click the name in the Performance Logs and Alerts window, point to **All Tasks**, and then click **Start** or **Stop**.

Getting the Most from Performance Logs and Alerts

Windows 2000 Server online Help for Performance Logs and Alerts describes performing the most common tasks with logs and alerts. The following list provides some additional hints about using the tools effectively:

- Export log data to a spreadsheet for reporting purposes. Importing log data into a spreadsheet program such as Excel offers benefits, such as easy sorting and filtering of data. To format the data for easy export, configure the log file type as Text File-CSV or Text File-TSV on the **Log Files** properties tab.
- Record transient data in a log. Not all counter log file formats can accommodate data that is not persistent throughout the duration of the log. If you want to record intermittent data such as a process that starts after you start the log, select the binary linear or circular file format on the **Log Files** tab.
- Limit log file size to avoid disk-space problems. If you choose automated counter logging with no scheduled stop time, the file will grow to the maximum size allowed based on available space on your disk up to 2GB (the largest log file that System Monitor can read). Trace logs have no file-size limit. When setting this option, take into consideration your available disk space and any disk quotas that are in place. Change the file path from the default (the Perflogs folder on the local computer) to a location with adequate space if appropriate. An error might occur if your disk runs out of disk space due to logging.
- Name files for easy identification. Use **File name** and **End file names with** on the **Files** properties tab to make it easy to find specific log files. For example, if you set up periodic logging, such as a log for every day of the week, you can develop different naming schemes with the base name being the computer where the log was run, or the type of data being logged, followed by the date as the suffix. For example, you could have a scheme that generates a file named ServerRed1_050212.blg, meaning it was created on a computer named ServerRed1 at noon, assuming the **End file name with** entry was set at **mmddhh**.
- Determine what trace data providers are available for trace logging. On the **General** properties tab, click **Provider Status** to see all data providers that have been installed. To see only enabled (running) data providers, click the **Show only enabled providers** check box in the **Provider Status** dialog box. For more information about WMI data providers, see the WMI SDK documentation in the MSDN Library at http://windows.microsoft.com/windows2000/reskit/webresources. You can only have one instance of each provider running at the same time.

Task Manager

Task Manager provides information about applications currently running on your system, the processes and memory usage or other data about those processes, and statistics about memory and processor performance.

Comparison with System Monitor

Although useful as a quick reference to system operation and performance, Task Manager lacks the logging and alert capabilities of the Performance console. In addition, although the data displayed by Task Manager comes from the same source as some performance counters, Task Manager does not have access to the breadth of information available from all installed counters. However, Task Manager provides capabilities not available with the Performance console, as described in Table 5.4. For information about these capabilities, see online Help for Task Manager and the chapters identified in Table 5.4.

Table 5.4 Other Chapters on Task Manager

Capability	Server Operations Guide chapter
Stop running processes.	"Analyzing Processor Activity"
Change the base priority of a process.	"Analyzing Processor Activity"
Set affinity for a process to a particular processor (on multiprocessor systems).	"Measuring Multiprocessor System Activity"

Starting Task Manager

To start Task Manager, use any of these methods:

- Press CTRL+SHIFT+ESC.
- Right-click the taskbar, and then click **Task Manager**.
- Press CTRL+ALT+DEL, and then click **Task Manager**.

You can also start Task Manager from the command prompt or the **Run** dialog box.

Working with Task Manager

Task Manager has three tabs: **Applications**, **Processes**, and **Performance**. While Task Manager is running, the status bar always displays the total number of processes, CPU use, and virtual memory use for the system. Note the following display possibilities:

- All Task Manager columns can be resized.
- Clicking a column sorts its entries in ascending or descending order.

- Select **Always on Top** from the **Options** menu to keep the window in view as you switch between applications.
- Press CTRL+TAB to toggle between tabs, or click the tab.

When Task Manager is running, an accurate miniature CPU usage gauge appears on the taskbar on the end opposite the **Start** button. When you place the mouse pointer over this icon, it displays the percentage of processor use in text format. The miniature gauge always matches the CPU Usage History chart on the **Performance** tab, as shown in Figure 5.9.

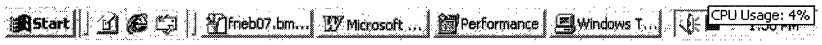


Figure 5.9 Task Manager CPU Gauge Shown on the Taskbar

To make Task Manager the top window, double-click the gauge, or right-click the gauge and then select **Task Manager** from the menu that appears.

If you run Task Manager frequently and do not want to see its button on the taskbar, click **Hide When Minimized** on the **Options** menu. To open an instance of Task Manager when it is hidden, click the Task Manager CPU gauge on the taskbar.

You can control the rate at which Task Manager updates its counts by setting the **Update Speed** option on the **View** menu.

- **High.** Updates every half-second.
- **Normal.** Updates once per second.
- **Low.** Updates every 4 seconds.
- **Paused.** Does not update automatically. Press F5 to update.

This will reduce Task Manager overhead, but might cause you to miss some data. You can force an update at any time by clicking **Refresh Now** on the **View** menu or by pressing F5.

Monitoring Processes

In **Task Manager**, click the **Processes** tab to see a list of running processes and measures of their performance. The Task Manager process table includes all processes that run in their own address space, including all applications and system services.

To include those in the display, on the **Options** menu, click **Show 16-bit Tasks**. Figure 5.10 is an example of how Task Manager displays process information.

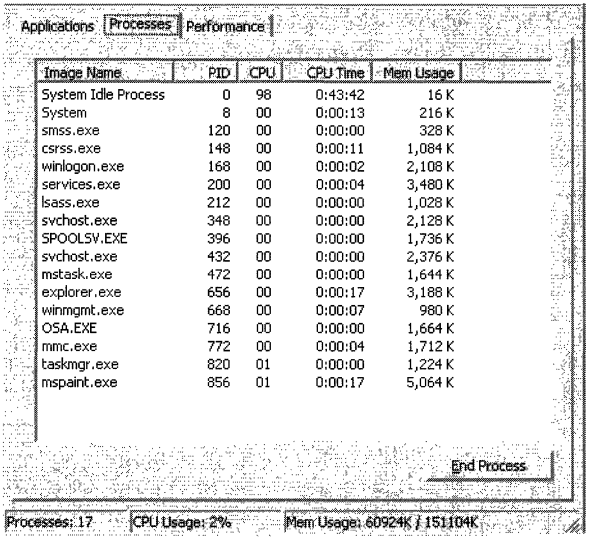


Figure 5.10 Processes Tab in Task Manager

Note System Monitor displays its values in bytes, whereas Task Manager displays its values in kilobytes, which are units of 1,024 bytes. When you compare System Monitor and Task Manager values, multiply System Monitor values by 1,024.

To add to or remove performance measures from the display for the processes listed, on the **View** menu, click **Select Columns**. Table 5.5 briefly describes the measures and their System Monitor counterparts, if any.

Table 5.5 Comparison of Process Data Supplied by Task Manager and System Monitor

Task Manager process measure	Description	System Monitor process object counters
Base Priority	The base priority of the process, which determines the order in which its threads are scheduled for the processor. The base priority is set by the process code, not the operating system. The operating system sets and changes the dynamic priorities of threads in the process within the range of the base. Use Task Manager to change the base priority of processes. For more information about changing priority to improve processor performance, see "Analyzing Processor Activity" in this book.	Priority Base

(continued)

Table 5.5 Comparison of Process Data Supplied by Task Manager and System Monitor *(continued)*

Task Manager process measure	Description	System Monitor process object counters
CPU Time	The total processor time, in seconds, used by the process since it was started.	None
CPU Usage	The percentage of time the threads of the process used the processor since the last update.	% Processor Time
GDI Objects	The number of Graphics Device Interface (GDI) objects currently used by a process. A GDI object is an object from the GDI library of application programming interfaces (APIs) for graphics output devices.	None
Handle Count	The number of object handles in the process's object table.	Handle Count
I/O Other	The number of input/output operations generated by a process that are neither reads nor writes, including file, network, and device I/Os. An example of this type of operation would be a control function. I/O Others directed to CONSOLE (console input object) handles are not counted.	I/O Other Operations/sec For more information about monitoring I/O, see the following chapters in this book: "Examining and Tuning Disk Performance" "Monitoring Network Performance"
I/O Other Bytes	The number of bytes transferred in input/output operations generated by a process that are neither reads nor writes, including file, network, and device I/Os. An example of this type of operation would be a control function. I/O Other Bytes directed to CONSOLE (console input object) handles are not counted.	I/O Other Bytes/sec
I/O Read Bytes	The number of bytes read in input/output operations generated by a process, including file, network, and device I/Os. I/O Read Bytes directed to CONSOLE (console input object) handles are not counted.	I/O Read Bytes/sec
I/O Reads	The number of read input/output operations generated by a process, including file, network, and device I/Os. I/O Reads directed to CONSOLE (console input object) handles are not counted.	I/O Read Operations/sec

(continued)