

THE EXPERT'S VOICE® IN NETWORKING

Includes
Windows and Linux
VMware Workstation
evaluation software

Virtualization

From the Desktop to the Enterprise

*Learn to deploy and manage virtual machines, clusters,
distributed file systems, and virtual storage in the first book
to cover the entire realm of virtualization*

Chris Wolf and Erick M. Halter

Apress®

VMware, Inc. Exhibit 1020 Page 1

Virtualization

From the Desktop to the Enterprise

CHRIS WOLF AND ERICK M. HALTER

Apress®

Virtualization: From the Desktop to the Enterprise
Copyright © 2005 by Chris Wolf and Erick M. Halter

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN: 1-59059-495-9

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Jim Sumser

Technical Reviewer: Harley Stagner

Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Tony Davis,

Jason Gilmore, Jonathan Hassell, Matthew Moodie, Chris Mills, Dominic Shakeshaft, Jim Sumser

Assistant Publisher: Grace Wong

Project Manager: Kylie Johnston

Copy Manager: Nicole LeClerc

Copy Editor: Kim Wimpsett

Production Manager: Kari Brooks-Copony

Production Editor: Kelly Winkist

Compositor: Van Winkle Design Group

Proofreader: April Eddy

Indexer: Carol Burbo

Artist: Diana Van Winkle, Van Winkle Design Group

Interior Designer: Diana Van Winkle, Van Winkle Design Group

Cover Designer: Kurt Krames

Manufacturing Manager: Tom Debolski

Distributed to the book trade in the United States by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013, and outside the United States by Springer-Verlag GmbH & Co. KG, Tiergartenstr. 17, 69112 Heidelberg, Germany.

In the United States: phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders@springer-ny.com, or visit <http://www.springer-ny.com>. Outside the United States: fax +49 6221 345229, e-mail orders@springer.de, or visit <http://www.springer.de>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

*This book is dedicated to my wonderful wife, Melissa, and son, Andrew.
True success is not measured by professional accomplishments
but rather by the love and respect of one's family.
As George Moore says, "A man travels the world over
in search of what he needs and returns home to find it."
—Chris Wolf*

*This book is dedicated to my family: Holly, Zack, Ella, and Gates,
and to the teachers who taught me to write and think...and Elvis too!
—Erick M. Halter*

Contents at a Glance

About the Authors	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii
CHAPTER 1 Examining the Anatomy of a Virtual Machine	1
CHAPTER 2 Preparing a Virtual Machine Host.	35
CHAPTER 3 Installing VM Applications on Desktops	69
CHAPTER 4 Deploying and Managing VMs on the Desktop	85
CHAPTER 5 Installing and Deploying VMs on Enterprise Servers.	139
CHAPTER 6 Deploying and Managing Production VMs on Enterprise Servers.	173
CHAPTER 7 Backing Up and Recovering Virtual Machines.	223
CHAPTER 8 Using Virtual File Systems	263
CHAPTER 9 Implementing Failover Clusters	309
CHAPTER 10 Creating Load-Balanced Clusters.	347
CHAPTER 11 Building Virtual Machine Clusters	379
CHAPTER 12 Introducing Storage Networking	413
CHAPTER 13 Virtualizing Storage.	435
CHAPTER 14 Putting It All Together: The Virtualized Information System	471
APPENDIX A Virtualization Product Roundup	495
INDEX	527

Contents

About the Authors	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii

CHAPTER 1	Examining the Anatomy of a Virtual Machine	1
	Introducing VM Types	2
	Hardware Emulators	2
	Application Virtual Machines	3
	Mainframe Virtual Machine	3
	Operating System Virtual Machines	3
	Parallel Virtual Machines	3
	Deploying VMs	5
	Choosing VM Hardware	5
	Introducing Computer Components	8
	CPU	9
	RAM	9
	Hard Drive	10
	Introducing Virtual Disk Types: Microsoft and VMware	11
	Virtual Hard Disk and Virtual Disks	12
	Dynamically Expanding and Dynamic Disks	12
	Fixed and Preallocated Disks	13
	Linked and Physical Disks	13
	Undo and Undoable Disks	13
	Differencing Disks	14
	Persistent and Nonpersistent Independent Disks	14
	Append Disks	15
	Resizing Disks	15
	Introducing Networking	16
	VM Networking Protocols	17
	TCP/IP	17
	DHCP	17
	NAT	18
	Introducing Networking VMs	20

Introducing Hardware	21
Network Interface Card	22
Switches	22
BIOS	24
Generic SCSI	25
I/O Devices	25
Introducing VM Products	26
Virtual PC	26
VMware Workstation	28
Microsoft Virtual Server 2005	29
VMware GSX Server	31
VMware ESX Server	32
Virtual Infrastructure	33
VMware VirtualCenter and VMotion	33
VMware P2V Assistant	33
Migrating Between VMs	34
VMware ACE	34
Summary	34

■ CHAPTER 2 **Preparing a Virtual Machine Host**

Implementing Best Practices	35
Evaluating Host Requirements	36
Selecting a Motherboard	38
CPU Speed and Quantity	40
Controller Chipset	40
Memory Requirements	41
Bus Types	42
Integrated Devices	43
Board Form Factor	44
Overall Quality	44
Considering Your Network	45
Public or Private VMs	45
Availability and Performance	47
Simplicity	48
Mesh Networks	48
Teaming and Load Balancing	49
Network Adapter Teaming	49
VM Networking Configurations	51
Supporting Generic SCSI	53
Windows Guests	53

Linux Guests	55
Considering Storage Options	56
Physical Hard Drive Specifications	57
RAID	59
Host Disk Sizing	61
Guest Disk Sizing	62
Storage Area Networks	63
Summary	67
 CHAPTER 3 Installing VM Applications on Desktops	 69
Deploying VMs with Microsoft Virtual PC	69
Installing VMware Workstation for Windows	71
Installing VMware Workstation for Linux	74
Installing the RPM	75
Installing the TAR	76
VM Host Tuning Tips	78
Summary	84
 CHAPTER 4 Deploying and Managing VMs on the Desktop	 85
Deploying VMs with VMware Workstation	85
Installing VM Tools	92
VMware Tools for Windows	93
VMware Tools for Linux	95
VMware Virtual Hardware Options for Windows and Linux	95
Microsoft Virtual PC: Building a Windows VM	101
Microsoft Virtual PC: Building a Linux VM	105
Virtual PC Virtual Hardware Options	106
Installing Virtual Machine Additions	108
Managing VMs	109
Backing Up and Modifying VM Configurations	109
VMware *.vmx Configuration Files	115
Virtual PC *.vmc Configuration Files	117
Copying and Moving VMware Workstation Guest VMs	119
VMware Universally Unique Identifiers	121
Copying and Moving Virtual PC VMs to Other Hosts	121
Running VMs As Services	123
Introducing VM CLI Administration and Keyboard Shortcuts	128
VMware Workstation CLI	129
Virtual PC CLI Administration	131

Monitoring and Configuring VM Performance	134
VMware Performance Counters	135
Virtual PC Performance Options	137
Summary	138

■ CHAPTER 5 **Installing and Deploying VMs on Enterprise Servers** 139

Installing Microsoft Virtual Server	140
Installing VMware GSX Server for Windows	143
Installing VMware GSX Server for Linux	146
Installing the RPM	147
Installing the TAR	149
Installing the VMware Management Interface	150
Working with the VMware Virtual Machine Console	152
Changing GSX Server's Remote Console Port Number	153
Installing VMware ESX Server	154
Verifying ESX Server Configuration Information	161
Viewing Configuration Files	161
Using Linux Survival Commands	163
Working with the Management Interface	165
Understanding MUI and SSL	165
Configuring the ESX Server Installation: Part One	166
License Agreement	167
Startup Profile	167
Storage Configuration	167
Swap File	168
Network Configuration	168
ESX Security	169
Configuring the ESX Server Installation: Part Two	170
Summary	171

■ CHAPTER 6 **Deploying and Managing Production VMs on Enterprise Servers** 173

Deploying VMs with VMware GSX Server and ESX Server	173
Building VMware GSX Server VMs	173
Building VMware ESX Server VMs	176
Mounting ISO Images	177
Installing VM Tools for GSX Server and ESX Server VMs	178
Using VMware Tools for Windows	178
Using VMware Tools for Linux	179

Configuring VMware GSX Server and	
ESX Server Virtual Hardware Options	180
Hard Disk	181
DVD/CD-ROM Drive	182
Floppy Drive	182
Ethernet Adapter	183
Sound Adapter	183
Configuring Legacy Devices	183
Configuring Generic SCSI Devices	184
Configuring a USB Controller	185
Scripting ESX Server USB Connectivity	191
Building Microsoft Virtual Server VMs	193
General Properties	194
Virtual Machine Additions	194
Memory	195
Hard Disks	195
CD/DVD	195
SCSI Adapters	196
Network Adapters	196
Scripts	196
Floppy Drives	196
COM Ports	197
LPT Ports	197
Managing Server-Class VMs	197
Modifying VM Configurations: Renaming and Moving	197
Using VMware Universally Unique Identifiers (UUIDs)	200
Importing Workstation and GSX Server VMs into ESX Server	202
Working with VMware GSX Server and	
ESX Server *.vmx Configuration Files	207
Working with Virtual Server *.vmc Configuration Files	210
Performing Command-Line Management	211
VMware	211
Microsoft	213
Using the Windows System Preparation Tool	214
Monitoring VM Performance	215
Monitoring ESX Server Performance	215
Monitoring VMware GSX Server Performance	219
Monitoring Virtual Server Performance	220
Performing Fault Monitoring and Fault Tolerance	221
Summary	222

CHAPTER 7	Backing Up and Recovering Virtual Machines	223
	Performing Traditional Agent-Based Backups	224
	Running Backup Agents on VMs	224
	Running Backup Agents on the Host	230
	Performing Non-Agent-Based Backups	230
	Using Windows Backup	231
	Backing Up Linux File Systems	234
	Performing Flat-File Backups	239
	Running VMware Workstation Flat-File Backups	241
	Running VMware GSX Server Flat-File Backups	247
	Running Virtual PC 2004 Flat-File Backups	251
	Running Virtual Server 2005 Flat-File Backups	252
	Taking Online Snapshots	257
	Performing a Full System Recovery	259
	Restoring Online VM Backups	260
	Restoring Flat-File VM Backups	261
	Summary	261
CHAPTER 8	Using Virtual File Systems	263
	Introducing DFS	263
	Implementing Windows DFS	265
	Implementing Linux DFS	271
	Using Samba with Kerberos Authentication	284
	Adding Samba to Active Directory	285
	Setting Up Samba DFS Shares	286
	Introducing AFS	288
	Implementing AFS	292
	Installing AFS for Linux Systems	302
	Summary	307
CHAPTER 9	Implementing Failover Clusters	309
	Introducing Failover Clustering	310
	Defining Essential Terms	311
	Introducing Cluster Architecture	312
	Introducing N-tier Clustering	313
	Working with Failover Cluster Products	314
	Planning for Failover Clusters	315
	Choosing the Right Model	316
	Configuring Cluster Hardware	319

Setting Up Microsoft Server Clusters	320
Looking Under the Hood	320
Planning Resource and Group Configuration	323
Installing the Windows Server 2003 Cluster Service	326
Using the Cluster Administrator	328
Setting Up Linux Failover Clusters	331
Setting Up the Red Hat Cluster Suite	331
Using Linux-HA Clusters	333
Summary	345
 CHAPTER 10 Creating Load-Balanced Clusters	347
Round-Robin DNS: The Beginning	349
Planning for Load-Balanced Clusters	351
Selecting Applications	351
Verifying Licensing	351
Analyzing Risks	352
Estimating Server Capacity	352
Building Windows Network Load-Balanced (NLB) Clusters	353
Enabling the NLB Service	353
Understanding Unicast and Multicast	354
Understanding Convergence	356
Setting Priority	357
Setting Port Rules	358
Understanding Remote Control	358
Using the Network Load Balancing Manager	358
Implementing Best Practices for NLB Cluster Implementations ..	359
Configuring and Managing Windows NLB Clusters	360
Building Linux Virtual Server (LVS) Clusters	374
Understanding LVS Architecture	375
Implementing LVS Implementation	377
Summary	377
 CHAPTER 11 Building Virtual Machine Clusters	379
Building Microsoft VM Clusters	379
Setting Up Windows Server Clusters	380
Setting Up iSCSI Windows Server Clusters	391
Installing the Windows Server 2003 Cluster Service	404
Setting Up Windows NLB Clusters	406
Building Linux VM Clusters	409
Summary	411

CHAPTER 12	Introducing Storage Networking	413
	Introducing SCSI	414
	Speaking SCSI	414
	ID vs. LUN	415
	Using SCSI Buses	415
	Understanding Termination	418
	Introducing Fibre Channel	421
	Introducing Fibre Channel Cables	423
	Introducing Fibre Channel Hardware Devices	424
	Understanding Zoning	428
	Configuring Fibre Channel Hardware	429
	Extending the SAN with FCIP and iFCP	430
	Introducing iSCSI	430
	Understanding iSCSI Architecture	430
	Securing iSCSI	431
	Using SAN Backup and Recovery Techniques	432
	Performing LAN-Free Backups	432
	Performing Server-Free Backups	433
	Performing Serverless Backups	434
	Summary	434
CHAPTER 13	Virtualizing Storage	435
	RAID: The Root of Storage Virtualization	435
	Introducing Common RAID Levels	435
	Implementing RAID	442
	Introducing the SNIA Shared Storage Model	443
	Why a Model for Shared Storage?	444
	Benefits of the Model	445
	A Note on the Graphical Conventions Used in the Model	445
	The Classic Storage Model	446
	The SNIA Shared Storage Model	447
	Applying the SNIA Shared Storage Model	461
	Understanding Host-Based Architecture	461
	Understanding Storage-Based Architecture	463
	Understanding Network-Based Architecture	463
	Adding Fault Tolerance to the SAN	466
	Performing Backups	466
	Introducing Hierarchical Storage Management	466

Using Virtual Tape Libraries	467
Dividing Physical Libraries	468
Writing to Magnetic Disk	468
Summary	469

■ CHAPTER 14 Putting It All Together: The Virtualized Information System

Reviewing the Elements of the Virtual IS	471
Failover Cluster	472
Load-Balanced Cluster	473
Virtual Machine Host	474
Storage Area Network	476
Distributed File System	479
Maintaining a Standby VM Server	480
Setting Up the VM	482
Maintaining a Standby Server with Scheduled Backups and Restores	483
Maintaining a Standby Server with Shared Storage	483
Maintaining a Standby Server with Disk Mirroring	485
Automating Standby Server Startup	489
Summary	494

■ APPENDIX A Virtualization Product Roundup

Global Namespace:	
The New Paradigm in Distributed Data Management	495
The Problem: Network Data Management and Movement	496
The Solution: Global Namespace	497
StorageX Uses Global Namespace to Deliver a Complete Network Data Management Platform	502
What Is Unique About the StorageX Global Namespace?	502
Conclusion	504
Server Consolidation and Beyond:	
Enhancing Virtual Machine Infrastructure Through Automation	505
Evolution of Virtualization in the Data Center	506
Enhancing Data Center Flexibility Through PlateSpin PowerP2V™	506
Comparing Other Methods of Converting Between Physical and Virtual Machine Infrastructure with PlateSpin PowerP2V	507

Replicating Entire Test Lab Environments Using Virtual Machines . .	510
Using Virtual Machines As Hot Backup Servers	
for Planned and Unplanned Downtime	511
Moving a Virtual Machine from One VM Host to Another (V2V) . . .	512
Production Server Virtualization.	513
Server Migrations Across Geographic Regions.	514
The Need for Continuous Resource Analysis and Rebalancing . .	514
Dynamic Virtual Machine Portability:	
Using Virtual Machines to Prevent SLA Violations	515
How Dynamic Virtual Machine Portability	
Can Enhance Business Service Management	516
Conclusion.	519
Rocket Division Software	519
iSCSI Target.	519
StarPort iSCSI Initiator, RAM Disk, and Virtual DVD Emulator. . . .	521
Mission and Technologies.	522
Market Focus	523
Network Instruments' Observer	523
Too Much Traffic in the Kitchen	523
Two Unique Networks Working Together	523
Maximizing the Network.	524
Complete Network Control	524
Capacity Analysis	524
Proactive Management	525
About Jack in the Box, Inc.	525
About Real-Time Expert	525
About Advanced Single Probes, Advanced Multi-Probes,	
and Advanced Expert Probes	525
About Network Trending	526
About "What-If" Analysis	526
About Network Instruments, LLC.	526
INDEX	527

About the Authors



CHRIS WOLF has worked in the IT trenches for more than a decade, specializing in virtualization, enterprise storage, and network infrastructure planning and troubleshooting. He has written four books and frequently contributes to *Redmond* magazine and *Windows IT Pro* magazine. Chris has a master's of science degree in information technology from the Rochester Institute of Technology and a bachelor's of science degree in information systems from the State University of New York–Empire State College.

Currently, Chris is a full-time member of the faculty at ECPI Technical College in Richmond, Virginia. When not teaching, Chris stays very involved in consulting projects for midsize to enterprise-class organizations and is a regular speaker at computer conferences across the nation. Chris is a two-time Microsoft MVP award recipient and currently has the following IT certifications: MCSE, MCT, CCNA, Network+, and A+.



ERICK M. HALTER, an award-winning IT/networking and security management educator for more than three years, is now the senior security administrator for a technology-based law firm where he's virtualizing the network and optimizing system processes for the Web. Erick has earned several industry certifications, including CCNP, MCSE: Security, Master CIW Administrator, SCNP, Security+, Linux+, and Net+, and he's an industry consultant. Erick completed his undergraduate studies in English and is currently earning a graduate degree in IT. He has more than a decade of practical experience in troubleshooting Microsoft, Cisco, and Linux networking technologies. He resides in Richmond, Virginia, with his wife and three dogs.

About the Technical Reviewer

■ **HARLEY STAGNER** has been an IT professional for seven years. He has a wide range of knowledge in many areas of the IT field, including network design and administration, scripting, and troubleshooting. He currently is the IT systems specialist for WWBT Channel 12, a local NBC affiliate television station in Richmond, Virginia. He is a lifelong learner and has a particular interest in storage networking and virtualization technology.

Harley has a bachelor's of science degree in management information systems from ECPI Technical College in Richmond, Virginia. He also has the following IT certifications: MCSE, CCNA, Network+, and A+.

Acknowledgments

Writing a book of this magnitude has certainly been a monumental task, and to that end I owe thanks to many. First, I'd like to thank you. Without a reading audience, this book wouldn't exist. Thank you for your support of this and of other books I've written to date.

Next, I'd like to thank my wonderful wife, Melissa. Melissa has been by my side throughout many of my writing adventures and is always willing to make the extra cup of coffee or do whatever it takes to lend support. I must also thank my mother, Diana Wolf, and my father, the late William Wolf. Thank you for encouraging me to always chase my dreams.

At this time, I must also thank my coauthor, Erick Halter. My vision for this book may not have been realized if not for Erick's hard work and persistence.

Several contributors at Apress were also extremely dedicated to this book's success. First, I must thank my editor, Jim Sumser, who shared in my vision for this book. Next, I must thank Kylie Johnston, the project manager. After having worked with Kylie before, I already knew that I'd be working with one of the industry's best. However, I also realize how easy it is to take everything Kylie does for granted. Kylie, you're a true professional and a gem in the mine of IT book publishing. Next, I must thank Kim Wimpsett, whose keen eye for detail truly made this book enjoyable to read. A technical book's true worth is most measured by its accuracy. With that in mind, I must also thank our technical editor, Harley Stagner, who diligently tested every procedure and script presented in this book.

I must also thank my agent, Laura Lewin, and the great team of professionals at Studio B. For one's ideas and vision to have meaning, they must be heard. Studio B, you're my virtual megaphone.

Finally, I must thank several technical associates who also added to the content of this book with their own tips and war stories. Please appreciate the contributions and efforts of the following IT warriors: Mike Dahlmeier, Jonathan Cragle, David Conrad, Scott Adams, Jim Knight, John Willard, Iantha Finley, Dan Vasconcellos, Harvey Lubar, Keith Hennett, Walt Merchant, Joe Blow, Matt Keadle, Jimmy Brooks, Altaf Virani, and Rene Fourhman.

—Chris Wolf

I aggravated and neglected a lot of people during this writing project: thank you for being patient and not giving up on me. Moreover, I am indebted to Chris, the crew at Apress, and the folks at Studio B for providing me with this opportunity to write about virtualization. Thank you. Thank you. Thank you.

—Erick M. Halter

Introduction

Virtualization is a concept that has evolved from what many first recognized as a niche technology to one that's driving many mainstream networks. Evidence of virtualization exists in nearly all aspects of information technology today. You can see virtualization in sales, education, testing, and demonstration labs, and you can see it even driving network servers.

What's virtualization? Well, to keep it simple, consider *virtualization* to be the act of abstracting the physical boundaries of a technology. Physical abstraction is now occurring in several ways, with many of these methods illustrated in Figure 1. For example, workstations and servers no longer need dedicated physical hardware such as a CPU or motherboard in order to run as independent entities. Instead, they can run inside a virtual machine (VM). In running as a virtual machine, a computer's hardware is emulated and presented to an operating system as if the hardware truly existed. With this technology, you have the ability to remove the traditional dependence that all operating systems had with hardware. In being able to emulate hardware, a virtual machine can essentially run on any x86-class host system, regardless of hardware makeup. Furthermore, you can run multiple VMs running different operating systems on the same system at the same time!

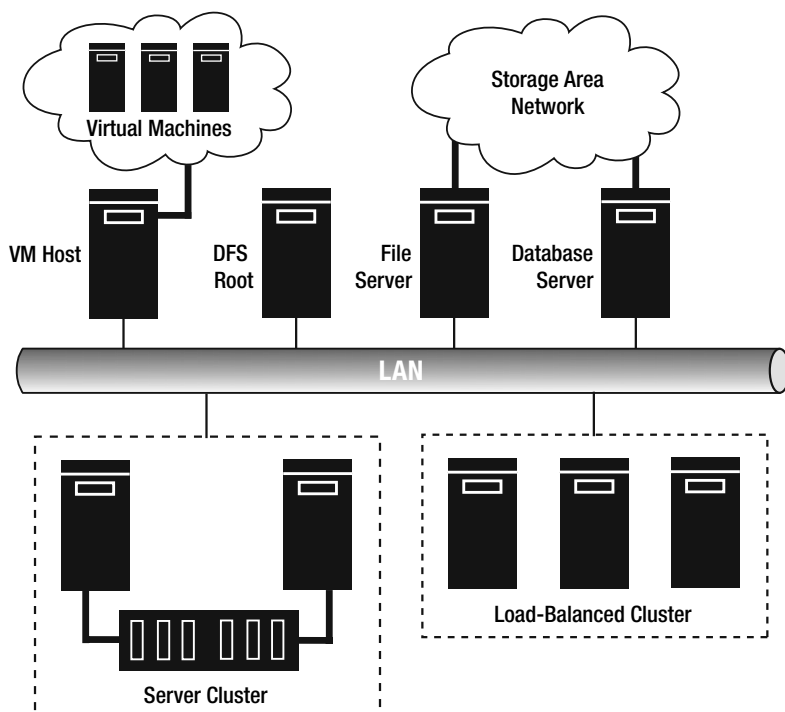


Figure 1. *A virtualized information system*

Virtualization extends beyond the virtual machine to other virtualization technologies such as clustering. *Clustering* allows several physical machines to collectively host one or more virtual servers. Clusters generally provide two distinct roles, which are to provide for continuous data access, even if a failure with a system or network device occurs, and to load balance a high volume of clients across several physical hosts. With clustering, clients won't connect to a physical computer but instead connect to a logical virtual server running on top of one or more physical computers. Clustering differs from virtual machine applications in that it allows for automated failover between physical hosts participating in the cluster. You could view failover as the movement of a virtual server from one physical host to another.

Aside from virtual machines and clustering, you'll also see the reach of virtualization extend to network file systems and storage. In reaching network file systems, technologies such as Distributed File System (DFS) allow users to access network resources without knowing their exact physical location. With storage virtualization, administrations can perform restores of backed-up data without having to know the location of the physical media where the backup resides.

Now, if your head is already spinning, don't worry, because you're probably not alone. With such a vast array of virtualization technologies available, it can be difficult to first tell one from another and also come to an understanding as to which technologies are right for you. That's why we decided to piece together a reference that explains each available virtualization technology, whether you're interested in running virtual machines on your desktop or are planning to add virtualization layers to an enterprise network environment. In this book, we'll guide you through all aspects of virtualization and also discuss how to fit any and all of these complex technologies into your IT life. Let's start by looking at the format of each chapter in this book.

Chapter 1: Examining the Anatomy of a Virtual Machine

Two major software vendors, EMC (Legato) and Microsoft, are leading the virtual machine software charge. In spite of their products' architectural differences, the terminology and the theory that drives them are similar.

In Chapter 1, we'll start by explaining the buzzwords and the theory associated with virtual machines. We'll address such topics as virtual networks, virtual hard disks, and CPU emulation. We'll also give you an overview of the major virtual machine products, outlining the differences between EMC's VMware Workstation, GSX Server, and ESX Server products, as well as Microsoft's Virtual PC and Virtual Server 2005.

Chapter 2: Preparing a Virtual Machine Host

With an understanding of the ins and outs of virtual machines, the next logical step in VM deployment is to prepare a host system. Several factors determine a host's preparedness to run a VM application, including the following:

- Physical RAM
- CPU
- Hard disk space
- Networking

When selecting a host, you'll need to ensure that the host meets the VM application's minimum hardware requirements and also that enough resources are available for the number of VMs you plan to run simultaneously on the host. Properly preparing a host prior to running virtual machines on it will almost certainly result in better stability, scalability, and long-term performance for your virtual machines. After finishing Chapter 2, you will not only be fully aware of the techniques for preparing a host system for virtual machines but will also be aware of the many gotchas and common pitfalls that often go unrecognized until it's unfortunately too late.

Chapter 3: Installing VM Applications on Desktops

When deciding to run virtual machines on workstations, you have two choices of workstation application: VMware Workstation (shown in Figure 2) and Virtual PC (shown in Figure 3). VMware Workstation is supported on Windows NT 4.0 (with SP6a) or higher Windows operating systems and can also run on Linux (Red Hat, Mandrake, or SuSE). Microsoft Virtual PC is supported on Windows 2000 Professional, Windows XP Professional, and Windows XP Tablet PC Edition.

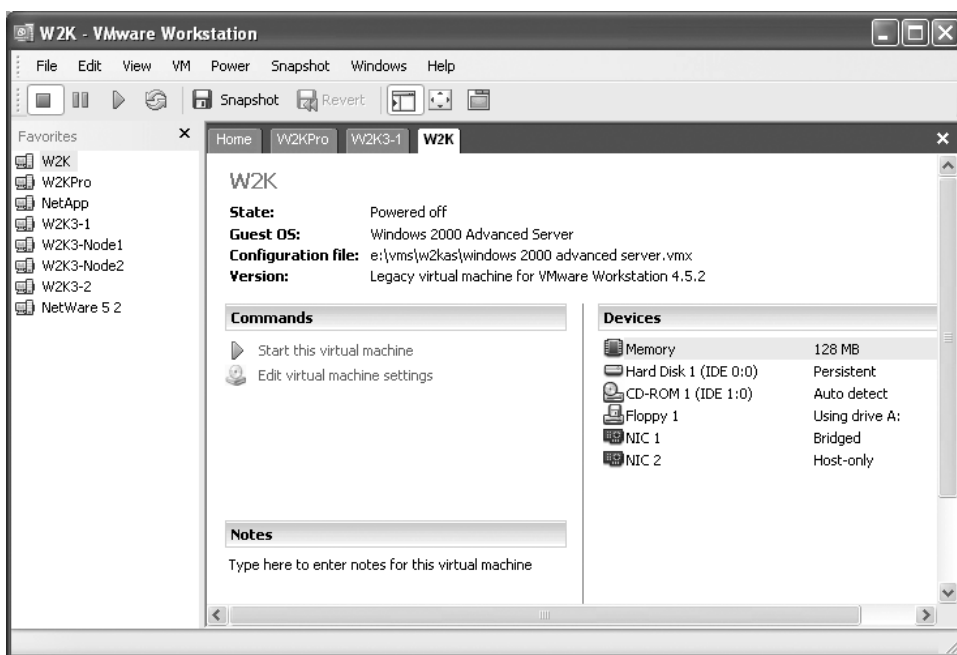


Figure 2. *VMware Workstation UI*

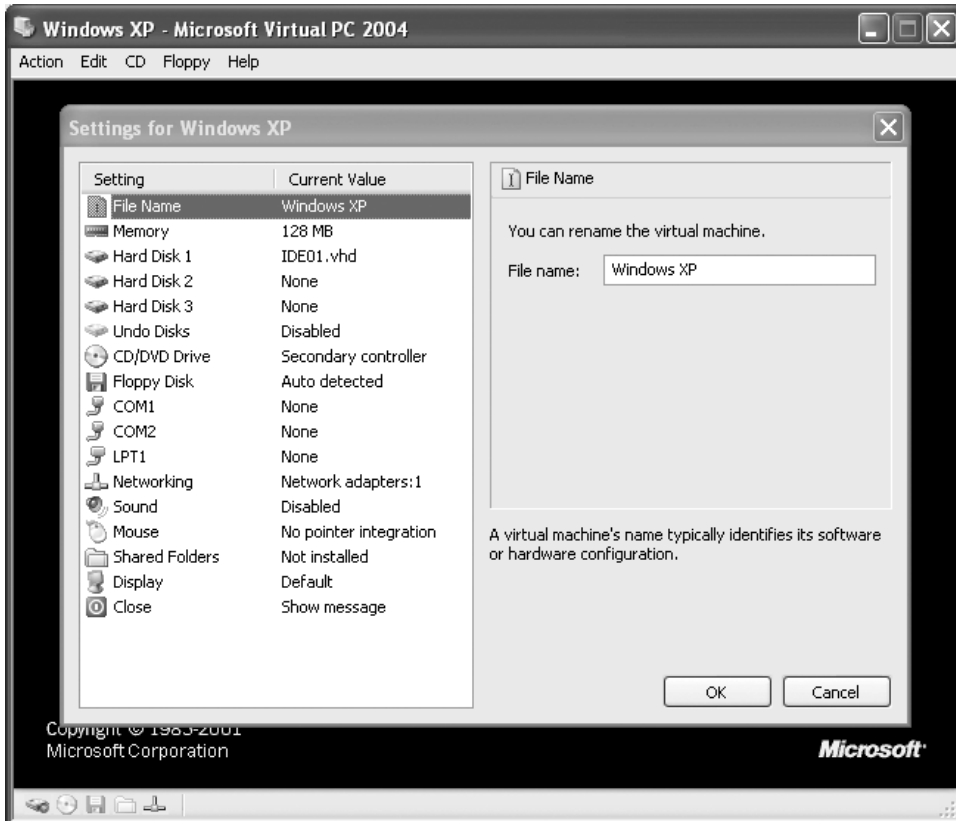


Figure 3. *Virtual PC UI*

As you can see, your current operating system may decide the VM application you choose. Chapter 1 will also help with the VM application decision, as it will outline all of the differences between each program. Once you've decided on a VM application, you'll be able to use this chapter to get you through any final preparations and the installation of VMware Workstation and Virtual PC.

Chapter 4: Deploying and Managing VMs on the Desktop

Chapter 4 provides guidance on deploying specific VM operating systems on your workstation system, including examples of both Windows and Linux VM deployments. Once your VM is up and running, you can perform many tasks to optimize and monitor their performance and to also ease future VM deployments. Topics in this chapter include the following:

- Monitoring the performance of VMs
- Staging and deploying preconfigured VMs

- Running VMs as services
- Configuring VMs to not save any information
- Administering through the command line and scripts

As you can see, this chapter is loaded with information on VM management. This is the result of years of experience, so you'll find the tips and techniques presented in this chapter to be as valuable as a microwave oven. Although they may not heat your food any faster, they definitely will save you plenty of time managing your virtualized network.

Simply installing and running virtual machines is really just the tip of the iceberg. Once your VMs are configured, you can perform many tasks to make them run better and run in ways you never thought possible. In Chapter 4, you'll see all of this and more.

Chapter 5: Installing and Deploying VMs on Enterprise Servers

Chapters 5 and 6 are similar in format to Chapters 3 and 4. In Chapter 5, we'll walk you through the installation and deployment process of VM applications on server systems, with the focus on using the VMs in a production role.

As with Chapter 3, in this chapter we devote time to both VMware GSX Server deployments on Linux and Windows operating systems and Microsoft Virtual Server 2005 deployments on Windows operating systems. With the decision to run VMs in production comes a new list of responsibilities. Whether you're looking to run domain controllers, file servers, or even database servers as VMs, you must consider several performance factors for each scenario before setting up the VMs. Many of us have learned the sizing game the hard way, and we don't want you to have to suffer as well. An undersized server can sometimes be the kiss of death for an administrator, as it may be difficult to get additional funds to "upgrade" a server that's less than two months old, for example. Sizing a server right the first time will not only make VM deployment easier but it may also help to win over a few of the virtualization naysayers in your organization. When deploying cutting-edge technology, you'll always have pressure to get it to run right the first time, so pay close attention to the advice offered in this chapter. We've made plenty of the common deployment mistakes already, so you shouldn't have to do the same!

Chapter 6: Deploying and Managing Production VMs on Enterprise Servers

Running VMs in production may involve managing fewer physical resources, but the fact that multiple OSs may depend on a common set of physical hardware can cause other problems. For example, rebooting a VM host server may affect four servers (a host plus three VMs) instead of one. Problems such as this put more pressure on you as an administrator to consider how an action on one server can impact several other servers.

With VMs in production, keeping the host system happy will likely result in well-running virtual machines. Again, with many systems depending on the hardware of one system, a hung CPU, for example, could have devastating consequences. This is why monitoring and mainte-

nance is still crucial after deployment. To help ease your mind, in this chapter we'll give you guidance on all the most common VM server administrative tasks, including the following:

- Automating the startup and shutdown of virtual machines
- Monitoring VM and host performance and alerting when performance thresholds are passed
- Using Perl and Visual Basic scripts for management, monitoring, and alerting
- Spotting host system bottlenecks and taking the necessary corrective action

As with its predecessors, you'll find Chapter 6 to have valuable information for managing production virtual machines.

Chapter 7: Backing Up and Recovering Virtual Machines

Now that you have your VMs up and running, you can't forget about protecting them. In this chapter, we'll cover all the methods for backing up and recovering virtual machines. We'll discuss the following:

- Backing up VMs with backup agent software
- Backing up VMs as "flat files"
- Backing up the VM host
- Using the available scripted backup solutions

As you can see, you'll have plenty of alternatives when deciding on how to best protect your virtual machines. In this chapter, we'll not only show you each VM protection methodology but we'll also outline the common pitfalls that exist with certain choices.

Chapter 8: Using Virtual File Systems

While the first half of the book is devoted to the design, deployment, and management of virtual machine solutions, the second half deals with the remaining virtualization technologies currently available. Chapter 8 leads off by explaining virtual file systems. With virtual file systems, you can manage files and file shares transparent to their physical location. For users, this means they won't need to know where a file is physically located in order to access it. For administrators, this means that if a file server needs to be brought down for maintenance, you can move its data temporarily to another server so that it remains available to users, without the users ever noticing a difference. Also, it's possible to configure replication with your virtual file system solution, which can allow for both load balancing and fault tolerance of file server data.

To tell the complete virtual file system story, we'll explain the most widely employed solutions available today, including both DFS and Andrew File System (AFS).

Chapter 9: Implementing Failover Clusters

The general concept of clustering is to allow multiple physical computers to act as one or more logical computers. With server or failover clusters, two or more physical servers will host one or more virtual servers (logical computers), with a primary purpose of preventing a single point of failure from interrupting data access. A single point of failure can be any hardware device or even software whose failure would prevent access to critical data or services. With the server cluster, a virtual server will be hosted by a single node in the cluster at a time. If anything on the host node fails, then the virtual server will be moved by the cluster service to another host node. This allows the virtual server running on the cluster to be resilient to failures on either its host system or on the network. Figure 4 shows a typical server cluster.

One aspect of the server or failover cluster that's unique is that all physical computers, or nodes, can share one or more common storage devices. With two nodes, this may be in the form of an external Small Computer System Interface (SCSI) storage array. For clusters larger than two nodes, the shared external storage can connect to the cluster nodes via either a Fibre Channel bus or an iSCSI bus.

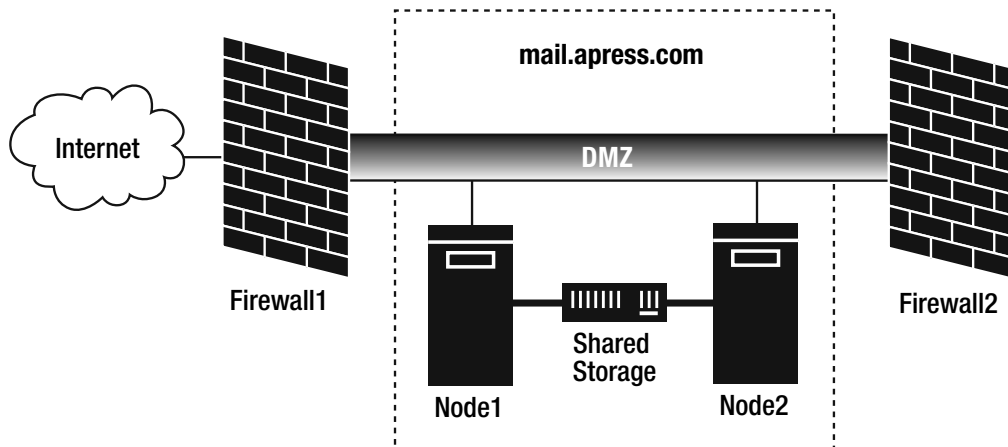


Figure 4. *Two-node mail server cluster*

In Chapter 9, we'll fully explain server clustering, outlining its deployment options and common management issues. To tell the complete clustering story, we'll cover the deployment and management of both Windows and Linux clustering solutions.

Chapter 10: Creating Load-Balanced Clusters

Load-balanced clusters give you the ability to relieve some of the load on an overtaxed server. With load balancing on Microsoft servers, you can configure up to 32 servers to share requests from clients. On Linux, you can even go beyond the 32-node limit imposed by Microsoft server operating systems.

In short, load balancing allows you to configure multiple servers to act as a single logical server for the purpose of sharing a high load of activity imposed by network clients. In a load-balanced cluster, two or more physical computers will act as a single logical computer, as

shown in Figure 5. Client requests are evenly distributed to each node in the cluster. Since all clients attempt to access a single logical (or virtual) server, they aren't aware of the physical aspects of the network server they're accessing. This means a client won't be aware of which physical node it's in communication with. Configuring load-balanced clusters will give you a great deal of flexibility in a network environment that requires a high level of performance and reliability. Because of its natural transparency to clients, you can scale the cluster as its load increases, starting with two nodes and adding nodes to the cluster as the demand requires.

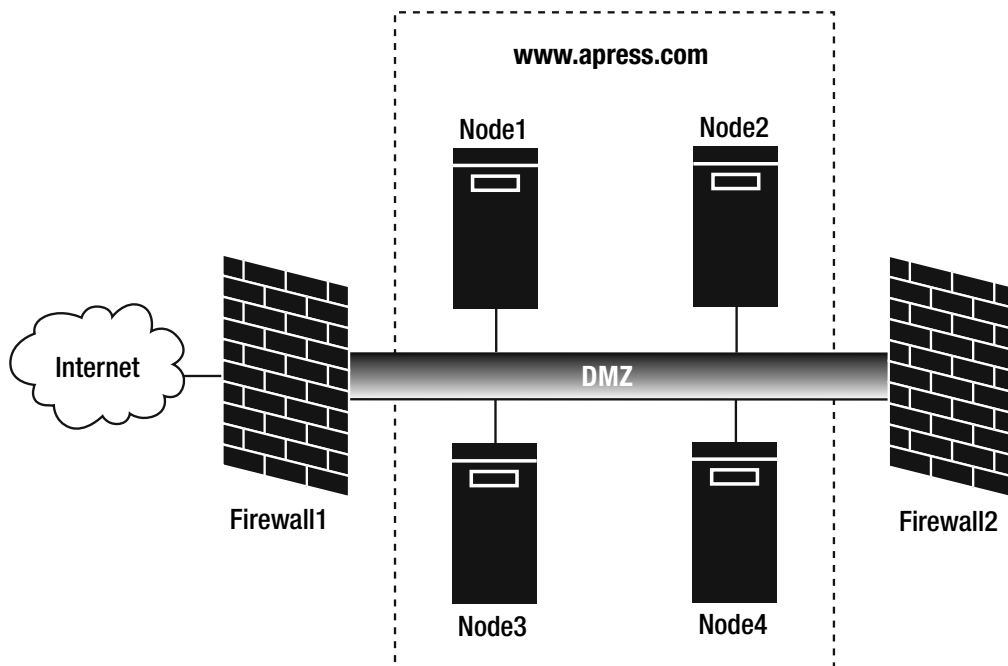


Figure 5. *Four-node load-balanced Web server cluster*

Since load-balanced clusters don't share a common data source (as with server or failover clusters), they're typically used in situations that require fault tolerance and load balancing of read-only data. Without shared storage, writing updates to a load-balanced cluster would be difficult to manage, since each node in the cluster maintains its own local copy of storage. This means it's up to you to make sure the data on each cluster node is completely synchronized. Because of this limitation, load-balanced clusters are most commonly a means to provide better access to Web and FTP services.

In Chapter 10, we'll take you through the complete design and implementation process for load-balanced clusters. We'll show you examples of when to use them and also detail how to deploy load-balanced clusters on both Windows and Linux operating systems.

Chapter 11: Building Virtual Machine Clusters

Although many organizations have or plan to run clusters in production, few have resources to test cluster configurations. That's where building virtual machine clusters can help. With virtual machine clusters, you can build and test working cluster configurations on a single system. Having this ability gives you several advantages, including the following:

- The ability to have others train on nonproduction equipment
- The ability to perform practice restores of production clusters to virtual machines
- The ability to perform live demonstrations of cluster configurations using a single system

Before virtual machines, most administrators had to learn clustering on production systems, if at all. Few organizations had the resources to run clusters in lab environments. When it comes time to test disaster recovery procedures, many organizations can't practice recovery for clusters, again because of limited resources. By being able to run a working cluster inside of virtual machines, organizations now have a means to test their backups of server clusters and in turn prepare recovery procedures for the production cluster. Applications and resources that run on server clusters are often the most critical to an organization. Oftentimes, when a disaster occurs, it's the clusters that must be restored first. Without having the ability to test recovery of the cluster and thus practice for such an event, recovering a cluster in the midst of a crisis can be all the more nerve-racking.

After reading Chapter 11, you'll understand the methodologies needed to configure nearly any cluster configuration using virtual machine technology.

Chapter 12: Introducing Storage Networking

The development of storage virtualization has been fueled by the rapid growth of storage networking. In short, storage networking allows you to network storage resources together for the purpose of sharing them, similar to a TCP/IP network of workstations and servers.

To understand the methodologies and benefits of virtualized storage resources, you must first be comfortable with storage networking technologies. This chapter lays the technical foundation for Chapter 13 by fully dissecting storage networking. Topics covered in Chapter 12 include storage area networks (SANs), network-attached storage (NAS), and direct-attached storage (DAS). Several modern storage networking protocols will be discussed, including the following:

- Fibre Channel Protocol (FCP)
- Internet Fibre Channel Protocol (iFCP)
- Fibre Channel over Internet Protocol (FCIP)
- Internet SCSI (iSCSI)

In addition to covering all the relevant storage networking protocols, we'll also dive into the hardware devices that drive storage networks. The following are some of the most common storage networking hardware devices that will be examined in this chapter:

- Fibre Channel switches
- Fibre Channel bridges and routers
- Fibre Channel host bus adapters (HBAs)
- Gigabit interface converters (GBICs)

Many in IT don't find storage to be the most thrilling topic, but nearly all realize its importance. Because of the inherently dry nature of storage, you may find that this chapter serves dual purposes: it lays the foundation for understanding storage virtualization, and it may substitute as an excellent bedtime story for your children, guaranteed to have them sleeping within minutes!

Chapter 13: Virtualizing Storage

Storage virtualization stays within the general context of virtualization by giving you the ability to view and manage storage resources logically. Logical management of storage is a significant leap from traditional storage management. For many backup administrators, having to restore a file often meant knowing the day the file was backed up and also knowing the exact piece of backup media on which the file was located. With storage virtualization, many backup products now abstract the physical storage resources from the administrator. This allows you as an administrator to simply tell the tool what you want, and it will find the file for you.

With data continuing to grow at a near exponential rate, it can be easy to become overwhelmed by the task of managing and recovering data on a network. As your data grows, so do the number of storage resources you're required to track. Having the right tools to give you a logical view of physical storage is key to surviving storage growth without having to seek mental counseling. Okay, maybe that's a stretch, but you'll certainly see how much easier your life as an administrator can become after you finish reading this chapter.

Chapter 14: Putting It All Together: The Virtualized Information System

Following a theme common to most technical references, Chapters 1–13 cover virtualization technologies one at a time, making it easy for you to find specific information on a particular topic. However, although it's nice to understand and appreciate each technology, it's also crucial to understand their interrelationships. In this chapter, you'll see examples of networks running several combinations of virtualization technologies simultaneously.

Many find relating virtualization technologies to their organization's networks to be challenging. Some common questions we run into quite frequently include the following:

- How can I justify an investment in virtualization to upper management?
- What are the best uses for server-class virtual machine products?

- What are the best uses for workstation-class virtual machine products?
- How can I optimize data backup and recovery between VMs and my production storage area network?
- What situations are best suited for clustering solutions?
- What questions should I ask when sizing up suitable hardware and software vendors?
- What precautions must be observed when integrating different virtualization technologies?

In addition to answering the most common questions surrounding running virtualization technologies in production and test environments, we'll also provide examples of the methods other organizations are using to make best use of their virtualization investments. This chapter wraps up with a detailed look at a process for maintaining standby virtual machines that can be automatically brought online if a production VM fails.

Appendix

Although virtualization product vendors such as EMC and Microsoft will go far to aiding you with support utilities, several other vendors also offer products to add virtualization layers to your existing network and to also aid in managing virtualized network resources.

In this appendix, we'll shower you with examples of some of the latest and greatest virtualization products on the market today. Several of these vendors have allowed us to include evaluation versions of their products on the book's companion CD. With so many virtualization software and hardware vendors contributing to the virtualized information system, you'll probably find managing your virtual resources to be much simpler than you ever imagined.

Summary

Virtualization is no longer an umbrella for disconnected niche technologies but is rather what's seen by many as a necessity for increasingly complex information systems. Virtual machine technology has broad appeal to many different levels of IT professionals. Sales associates can run software demonstrations on virtual machines. Instructors now have a tremendous amount of flexibility when teaching technical classes. They can now demonstrate several operating systems in real time, and their students no longer have to team up with a partner in order to run client-server networking labs. Now students can run client and server operating systems on a single computer in the classroom.

VM technology today has reached several other IT professionals as well. Network architects and administrators can now test software deployments and design solutions on virtual machines, prior to introducing new technologies to a production environment. For our own testing, we've found that our wives are much happier since we no longer each need half a dozen computers in the home office. Now a single computer with several virtual machines works just fine. However, both of us no longer receive a Christmas card from the electric company thanking us for our high level of business each year.

Although the savings on electricity might be a bit of a stretch, the expanded roles of virtualization certainly aren't. When approaching this book, start with Chapter 1 if your first interest is

virtual machines. This will set a solid foundation for Chapters 2–7. Following Chapter 7, you'll find that the remaining chapters serve as independent references on the various technologies that drive virtualization. Since Chapter 14 focuses on making all virtualization technologies seamlessly operate together in the same information system, you'll want to read it after Chapters 1–13.

To safely integrate virtualization technologies into your IT life, you first need to have a good understanding of what's available, when to use each technology, and also how to use them. That being said, let's not waste any more time discussing how great virtualization technology is. Instead, turn to Chapter 1 to get started with the anatomy of a virtual machine.



Examining the Anatomy of a Virtual Machine

In the preface, you learned how to quickly slice and dice this book to get good, quick results. Now you'll look at what's going on under the hood of virtual machines (VMs). At the component level of VMs and physical hardware, you'll revisit some fairly basic concepts, which you might take for granted, that contribute to the successful virtualization of a physical computer. By taking a fresh look at your knowledge base, you can quickly tie the concept of virtualized hardware to physical hardware.

Wrapping your mind around the term *virtual machine* can be daunting and is often confusing. This stems from the varying definitions of the term *virtual machine*. Even the term *virtual* brings unpredictability to your understanding if you don't understand *virtual* to mean "essence of" or "properties of." In short, if a list of specifications is equal to a machine (think: personal computer), and if software can create the same properties of a machine, you have a VM. If you further reduce a computer to its vital component, electricity, it's easier to understand a VM: the entirety of a computer deals with "flipping" electricity on or off and storing an electrical charge. If software exists such that it can provide the same functionality, a hardware emulator, then a VM exists.

Hardware virtualization offers several benefits, including consolidation of the infrastructure, ease of replication and relocation, normalization of systems, and isolation of resources. In short, VMs give you the ability to run multiple virtual computers on the same physical computer at the same time and store them on almost any media type. VMs are just another computer file offering the same ease of use and portability you've grown accustomed to in a drag-and-drop environment.

Virtualization software protects and partitions the host's resources, central processing units (CPUs), memory, disks, and peripherals by creating a virtualization layer within the host's operating system (OS) or directly on the hardware. "Running on the metal" refers to virtualization software that runs directly on the hardware: no host operating system is required to run the software. Every virtual machine can run its own set of applications on its own operating system. The partitioning process prevents data leaks and keeps virtual machines isolated from each other. Like physical computers, virtual machines require a physical network, a virtual network, or a combination of both network types to communicate.

The virtualization layer abstracts the hardware for every guest operating system: *abstraction* is the process of separating hardware functionality from the underlying hardware. Because the operating system is built on idealized hardware, you can change physical hardware without impacting the function of virtual machines. The virtualization layer is responsible for mapping virtualized hardware to the host's physical resources. Moreover, the further an operating system is abstracted from the hardware, the easier it is to recover. You can think of abstracted operating systems in terms of software applications. Traditional operating systems abstract hardware so software can be written independent of hardware. Without an operating system, programmers would have to write the same program for every system that wasn't identical to the development machine.

Virtualization software takes the concept of abstraction one step further—it abstracts the hardware by employing a virtualization layer. This creates an extra layer between the hardware and operating system. One of the duties of the virtualization layer is to create virtual hardware. Despite that the hardware is really software, operating systems see the virtual hardware as physical devices. Assuming the virtualization software is installed on a host system, a guest operating system can be copied from disparate physical systems, and the VM will function as if nothing happened.

Introducing VM Types

VMs come in several varieties and are defined by how the virtualization process is performed. You accomplish *virtualization* by completely emulating the hardware in a computer system or by mapping resources from the physical computer to a virtual machine. *Emulating* a virtual machine is the process of duplicating the physical structure by using software, and *mapping* is the process of trapping software routines and passing instructions to the physical hardware. Both approaches work well and render the same result, a VM. In this chapter, we'll cover hardware emulators, application VMs, mainframe VMs, operating system VMs, and parallel VMs. In general, a VM describes a complete end-user computing environment created by software.

Hardware Emulators

Hardware emulators programmatically duplicate physical architectures to provide native functionality for software. For instance, Microsoft Virtual PC for Mac emulates the i386 architecture for the PowerPC chip. With Virtual PC, it's as if a fully functioning x86 computer has been reduced to an icon. Hardware emulators are useful for re-creating hardware that no longer exists, sharing expensive resources, and porting software to different computing system architectures. Hardware emulators, unlike VMs, focus on running software written for a specific type of processing architecture on a completely different architecture. For instance, Transitive's QuickTransit emulator allows code written for Sun Solaris to be executed on an Intel x86 processor and allows an operating system written for the Intel processor to run on the PowerPC chip.

Application Virtual Machines

An *application virtual machine* (AVM) is software that isolates a running application from the computer hardware. The result of the isolated application computing environment is that application code is written once for the virtual machine, and any computer capable of running the VM can execute the application. AVMs save developers from rewriting the same application for different computing platforms: only the AVM is ported to different computing platforms. Examples of AVMs are Java and Microsoft's .NET.

Mainframe Virtual Machine

A *mainframe virtual machine* (MVM) is a software computing environment emulating the host computer. Virtual machines copy not only the host's software environment but also its physical environment. For computer users, the VM creates the illusion that each user is in command of a physical computer and operating system. For owners of expensive mainframes, the virtual machine allows efficient sharing of valuable computing resources and the security settings that prevent concurrently running guest VMs from interfering with one another. Any number of IBM mainframe systems fall into this category, such as System/370 or System/390.

Operating System Virtual Machines

Operating system virtual machines (OSVMs) create an environment of an operating system for the computer user. Unlike the MVM emulation, OSVMs achieve virtualization by mapping the physical computer environment on guest operating systems. The computer on which the OSVM runs executes its own operating systems—a virtualized computer and operating systems within a physical computer and operating system. VMware Workstation and GSX Server, as well as Microsoft Virtual PC and Virtual Server, fall into this category.

Parallel Virtual Machines

It may be difficult to differentiate between *parallel virtual machines* (PVMs) and parallel processing. PVMs consist of one computing environment running on multiple computers employing distributed processing. PVMs create the illusion that only one computer is being used rather than many. On the other hand, distributed processing is the act of several if not thousands of computers working together for a greater good. In general, networked computers conquering a large processing task. This task is split into small chunks whereby each computer in the group is charged with completing an assignment and reporting the results. Distributed processing doesn't have a single-user interface for an end user. Recent famous distributed processing projects are the Seti@Home project (<http://www.seti.org>) and Project RC5 (<http://www.distributed.net>). Examples of PVMs include Harness and PVM (<http://www.csm.ornl.gov>).

OPEN-SOURCE VIRTUAL MACHINES

As with the meaning of any word subjected to the passing of time and cultural impacts, definitions change. The term *open source* isn't immune to this. The traditional definition of *open source* has grown beyond its typical meaning referring to software whose code is free to use, look at, modify, and distribute. To get a good feel for the true spirit of the terms *open source* and *free*, you can visit the Open Source Initiative definition at <http://www.opensource.org/docs/definition.php> or the Free Software Foundation definition at <http://www.gnu.org/philosophy/free-sw.html>. Despite what you know *open source* and *free* to mean, no excuse exists for not reading the license (contract) packaged with software. License compliance becomes even more important in corporate production environments, especially when using open-source VMs or emulators. We'll briefly discuss several open-source virtualization applications: Xen, Multiple Arcade Machine Emulator (MAME), Bochs, and DOSBox.

A popular open-source VM gaining momentum in the open-source community is Xen. It originated at the University of Cambridge and is released under the terms of the GNU General Public License (GPL). Not only are the independent developers involved, but recently Xen has been endorsed by several major corporations, including Hewlett-Packard, IBM, Intel, Novell, Red Hat, and Sun Microsystems. Xen uses virtualization techniques that allow you to run multiple Linux-like operating systems at nearly native speeds. Currently, Xen doesn't support Microsoft Windows. You can read more about Xen at <http://www.cl.cam.ac.uk/Research/SG/netos/xen/>.

Arcade emulators are popular in the gaming community and are a great way to “win back” some youthful memories without having to pump quarters into an arcade game. Gaming emulators, such as MAME (<http://www.mame.net>), have been around since the early 1990s and mimic long-lost or aging arcade game hardware. MAME employs a modular driver architecture supporting more than 5,000 roms (games) and nearly 3,000 arcade games. By emulating the hardware, the game code stored in the rom of arcade games can run on a modern computer. Because MAME takes the place of the stand-up arcade console, you can play the actual game. Though the code stored in rom can be easily extracted and saved to the Internet, patents and copyrights protect many games. Protect yourself from any legal ramifications if you choose to download and play the thousands of roms available on the Web.

Bochs (pronounced *box*) is an open-source computer emulator written in C++ with its own custom basic input/output system (BIOS). It's capable of emulating x86 processors, including 386, 486, AMD64 CPU, and Pentium Pro. It can also emulate optional features such as 3DNow, MMX, SSE, and SSE2. Known operating systems it's capable of running with common input/output (I/O) devices include Linux, DOS, Windows 95, Windows NT 4, and Windows 2000. Bochs is generally used in Unix environments to emulate a computer. The emulated computer is executed in a window with your desired operating system. With Bochs, a Unix user can run software packages not normally associated with Unix environments, such as a Windows operating system loaded with Microsoft Office. Bochs is a project supported by SourceForge.net; you can download it at <http://bochs.sourceforge.net/>.

DOSBox is open-source code that emulates a 286/386 CPU in real or protected mode. It uses the Simple DirectMedia Layer (SDL) library to gain access to components on the host computer, such as the mouse, joystick, audio, video, and keyboard. DOSBox can run on Linux, Mac OS X, and Windows operating systems. The main focus of DOSBox is gaming and is consequently not sophisticated enough to take full advantage of networking or printing. If you have a hankering to revisit your favorite game or application, dust off your floppy disks and fire up DOSBox.

Deploying VMs

VMs are comparable to physical machines in many ways, and these similarities make it easy to transfer existing knowledge of operating system hardware requirements to VMs. Decisions that will drive you to use VMs include the need to demonstrate application or network configurations on a single computer, such as showing students or seminar attendees how to install and configure operating systems or applications. You may decide that investing in a complete test network is cost prohibitive and realize that VMs are a practical way to safely test upgrades, try service releases, or study for certification exams. You can also increase your infrastructure uptime and ability to recover from disasters by deploying clustered VMs on individual server hardware.

When preparing to virtualize, remember that more memory, faster processors, and plenty of high-performance hard disk space make for better VMs. Don't expect to have good or even fair performance from a VM if only the minimum system requirements for the guest operating system are satisfied. It's always important to check the minimum and best-practice specifications for any operating system prior to installation. When considering host hardware for guest VMs, bigger is better! You can refer to the tables later in this chapter for the minimum physical hardware and host operating system requirements for running VM applications.

Please note that the hardware on the host computer must have the minimum requirements to support its operating system and the guest operating system. The minimum specifications don't necessarily represent the absolute minimum hardware configurations. The following minimums represent manufacturer minimums and reflect their desire to have owners of old hardware be able to upgrade to the next-generation operating system. When using Virtual PC, for example, to run Windows NT 4 as a guest on a Windows XP host, you'll need 1.5 gigabytes (GB) of disk space and 128 megabytes (MB) of random access memory (RAM) for the host and an additional 1GB of disk space and 64MB of RAM for the guest operating system. In total, if you don't have 2GB of disk space and 192MB of RAM, the guest operating system won't function. Attempting to run resource-starved VMs is an exercise in futility and defeats every benefit VMs have to offer.

When loading a host with guest VMs, too many is as bad as too few—too many will cause host hardware resource usage to approach 100 percent, and too few will underutilize resources, wasting money and increasing administrative overhead. Ideally, enough VMs should be loaded onto a host system to consume 60–80 percent of the host's total resources; this allows for resource usage spikes without sacrificing performance or hardware investment.

Choosing VM Hardware

You can choose from a myriad of computer commodities in today's marketplace. Whether you decide to deploy a virtual infrastructure on inexpensive white boxes, moderately priced commercial systems, or expensive proprietary computers, no excuse exists for not doing your homework with regard to hardware compatibility lists (HCLs). You know operating system vendors publish HCLs, so you should use them to ensure that your choice of host virtualization hardware has been tested thoroughly and will perform satisfactorily.

As with everything, exceptions do exist to HCLs, particularly with regard to operating systems listed as end of life (EOL). Your VM host will invariably have new hardware that wasn't around for legacy operating systems when it was in its prime, such as DOS, Windows NT 4, and NetWare 3.11. Despite that newer hardware may be absent on HCL listings for EOL operating systems, you can still run these systems as VMs. Table 1-1 shows Microsoft's choice of virtual hardware for guest operating systems, and Table 1-2 shows VMware's.

Table 1-1. *Microsoft Virtual Hardware Specifications*

Virtual Device	Virtual PC	Virtual Server
Floppy drive	1.44MB	1.44MB
BIOS	American Megatrends	American Megatrends
CD-ROM	Readable	Readable
DVD-ROM	Readable	Readable
ISO mounting	Yes	Yes
Keyboard	Yes	Yes
Mouse	Yes	Yes
Tablet	No	No
Maximum memory	4GB	64GB
Motherboard chipset	Intel 440BX	Intel 440BX
Parallel port	LPT 1	LPT 1
Serial port	COM 1, COM 2	COM 1, COM 2
Processor	Same as host	Same as host
Sound	SoundBlaster	No
Video	8MB S3 Trio	4MB S3 Trio
IDE devices	Up to 4	No
SCSI	No	Adaptec 7870
NIC	Intel 21141 Multiport 10/100	Intel 21141 Multiport 10/100
USB	Keyboard/mouse only	Keyboard/mouse only
PCI slots	5	5

Table 1-2. *VMware Virtual Hardware Specifications*

Virtual Device	Workstation	GSX Server	ESX Server
Floppy drive	1.44MB	1.44MB	1.44MB
BIOS	Phoenix BIOS	Phoenix BIOS	Phoenix BIOS
CD-ROM	Rewritable	Rewritable	Rewritable
DVD-ROM	Readable	Readable	Readable
ISO mounting	Yes	Yes	Yes
Keyboard	Yes	Yes	Yes
Mouse	Yes	Yes	Yes
Tablet	Yes	Yes	Yes
Maximum memory	4GB	64GB	64GB
Motherboard chipset	Intel 440BX	Intel 440BX	Intel 440BX
Parallel port	LPT 1 and 2	LPT 1 and 2	LPT 1 and 2
Serial port	COM 1–COM 4	COM 1–COM 4	COM 1–COM 4
Processor	Same as host	Same as host	Same as host
Sound	SoundBlaster	SoundBlaster	SoundBlaster
Video	SVGA	SVGA	SVGA
IDE devices	Up to 4	Up to 4	Up to 4
SCSI	LSI 53c1030, BusLogic BT-358	LSI 53c1030, BusLogic BT-358	LSI 53c1030, BusLogic BT-358
NIC	AMD PCnet-PC II 10/100	AMD PCnet-PC II 10/100/1000	AMD PCnet-PC II 10/100/1000
USB	USB 1.1	USB 1.1	USB 1.1
PCI slots	6	6	5

Though you may have success with hardware not found on an HCL in a test environment, best practices dictate thorough testing of non-HCL equipment before deploying the system in a production environment. Despite all the HCL rhetoric, it's sufficient to be aware that hosted operating systems work with each manufacturer's given portfolio of hardware.

Just keep in mind that the purpose of an HCL is to ensure hardware driver compatibility with an operating system. So that you don't have to extensively test hardware before deployment, operating system vendors employ rigorous certification programs for hardware manufacturers. Given that so much time and money is already spent on testing and compiling lists of hardware compatible with an operating system, it makes sense to build or buy computers listed on manufacturer HCLs.

Choosing computer systems and components surviving certification testing and thereby earning the right to appear on a manufacturer's HCL saves time, money, and aggravation; in addition, it saves countless hours of troubleshooting “weird problems” and ensures successful implementation in the enterprise. Taking the time to verify HCL compliance is the difference between hacks and professionals. Let manufacturers retain the responsibility for testing; if not, you're stuck holding the “support bag” when a system fails. You can find popular manufacturer HCLs at the following sites:

- **Microsoft's HCL:** <http://www.microsoft.com/whdc/hcl/default.mspx>
- **VMware's HCL:** <http://www.vmware.com>
- **Red Hat's HCL:** <http://hardware.redhat.com/hcl>

When testing new operating systems, it's tempting to recycle retired equipment and think you're saving money. Be careful; old equipment goes unused for specific reasons, and such hardware has reached the end of its useful life. In addition, old equipment has hours of extensive use probably reaching the failure threshold; using retired equipment is a false economy trap. What money you save initially will surely be wasted in hours of aggravating work later and deter you from embracing virtualization. Being cheap can lead to missing out on the complete rewards of VMs, as discussed earlier. If you're still inclined to drag equipment from basements and overcrowded information technology (IT) closets, be sure to verify HCL compliance or virtual hardware compatibility, clear your schedule, and have plenty of spare parts on hand.

Unlike retired computer equipment, using existing production equipment often provides some cost savings during operating system testing or migration. A hazard of using production equipment is that it may be nearly impossible to take a system offline. In addition, attempting to insert test systems into a production environment can have adverse impacts, and the time invested in reconfiguring systems negates any hard-dollar cost savings. However, if you have the luxury of time and are able to jockey around resources to safely free up a server meeting operating system HCL compliance, you not only save some money but have reason to ask for a raise!

The best solution for VM testing is to buy new equipment and create an isolated test network that won't impact your production environment. A good test network can simply consist of a cross-connect patch cable, a basic workstation, a server meeting HCL compliance, and the best-practice RAM and CPU configurations. A new test network makes experimenting enjoyable, and it will ensure your success when rolling out VMs.

Introducing Computer Components

In the following sections, we'll briefly cover the computer components involved with VMs. Whether you're refreshing your memory or coming to your first understanding of computer components, the following information can help you troubleshoot and optimize VM performance issues that will inevitably arise.

Most of us are already aware of the typical function and structure of basic computer components; however, those of us who are grappling with how virtualization takes place will need some of the granular explanations that follow. These sections distill the functions of computer components to their basic properties. This will help you connect the dots to realize that VMs are just files.

CPU

VMs employ a virtual processor identical to the host computer and accomplish virtualization by passing nonprivileged instructions directly to the physical CPU. Privileged instructions are safely processed via the VM monitor (VMM). By allowing most commands to be directly executed, guest VMs will approximate the speed of the host. Each guest VM will appear to have its own CPU that's isolated from other VMs. In addition, every virtual CPU will have its own registers, buffers, and control structures. If the host system is Intel x86–based, the guest operating systems will use an Intel x86 architecture; the same goes for compatible processors (such as AMD).

Depending on the software version and the manufacturer, such as VMware ESX Server and Microsoft Virtual Server 2005, you may be able to use multiple CPUs if the host hardware physically contains multiple CPUs. While configuring the guest OS, you simply choose as many processors as the guest will use, up to the number in the host OS. Table 1-3 gives you a quick look at the maximum number of processors Microsoft and VMware VMs can handle on the host hardware and the maximums that can be allocated for guest VMs.

Table 1-3. *Virtual Machine Maximum CPU and Memory Specifications*

Virtual Machine	Host Processors	Guest Processors	Host RAM	Guest RAM
Virtual PC 2004	Up to 2	1	4GB	3.6GB
Virtual Server 2005	Up to 4	1	64GB	3.6GB
Virtual Server 2005 Enterprise	Up to 32	1	64GB	3.6GB
VMware Workstation	Up to 2	1	4GB	3.6GB
VMware GSX Server	Up to 4	1	64GB	3.6GB
VMware ESX Server	Up to 16	2	64GB	3.6GB

RAM

We all know the most celebrated type of computer memory, the type of memory that's ephemeral and loses the contents of its cells without constant power, the type of memory we're always in short supply of—RAM. Like CPU virtualization, guest VMs access RAM through the VMM or virtualization layer. The VMM is responsible for presenting VMs with a contiguous memory space. The host VM, in turn, maps the memory space to its physical resources. The management of the virtual memory pool is completely transparent to the guest VM and its memory subsystems.

From a performance and scaling capability, you'll be most concerned with how VMs handle nonpaged and paged memory. Both memory types are created on system boot. Nonpaged memory consists of a range of virtual addresses guaranteed to be in RAM at all times, and paged memory can be swapped to slower system resources such as the hard drive. The ability to use paging will allow you to create and run more VMs on a physical computer. However, swapping memory from the hard disk to RAM will reduce performance.

VMware can use a memory pool consisting of paged and nonpaged resources: swapped pages, shared pages, contiguous and noncontiguous physical pages, and unmapped pages. When creating guests with VMware products, you have a choice of running the VMs in RAM, running them in mostly RAM and some paging, or running them in mostly paging and some RAM. In performance-driven environments, shove all your VMs into RAM. If you want to maximize hardware investments, you may want to take a small performance hit and allow some

paging to be able to consolidate more servers on one box. In a test environment where it's necessary to have many systems running to simulate a complete network, more paging is acceptable.

Virtual PC and Virtual Server both prohibit memory overcommitment. They prevent the host operating system from swapping virtual guest RAM to the physical hard disk. Being that the physical resources available to Virtual PC and Virtual Server are limited to nonpaged host RAM, both experience maximum performance at all times.

Excessive paging can paralyze the host computer and cause guest operating systems to appear to hang. Nonpaged memory, on the other hand, will limit running VMs to the amount of physical RAM installed. Although you may not be able to turn up as many VMs as you could with paging, you'll experience increased performance by having VMs in RAM at all times. When you start testing and benchmarking VMware and Microsoft virtualization applications, be sure to compare apples to apples by using the RAM-only feature in VMware.

When using Virtual Server in the enterprise, you may be boosting the physical memory of servers beyond 4GB. If this is the case, you'll have to use physical address extensions (PAE) and know that Microsoft limits its support to PAE systems listed on the Large PAE Memory HCL. Table 1-4 lists the supported operating systems and RAM configurations.

Table 1-4. *Microsoft Operating Systems and RAM Configurations*

Operating System	PAE Memory Support
Windows 2000 Advanced Server	8GB of physical RAM
Windows 2000 Datacenter Server	32GB of physical RAM
Windows Server 2003, Enterprise Edition	32GB of physical RAM
Windows Server 2003, Datacenter Edition	64GB of physical RAM

Hard Drive

Unlike RAM, a hard drive is considered the primary permanent storage device of VMs. Despite having apparently the same function, Microsoft and VMware use different terminology to describe their virtual disks. In the next section, we'll discuss each term independently. Understanding virtual drives is as simple as being able to make the logical leap from knowing that a physical drive is a group of rotating magnetically charged platters to understanding that a virtual hard drive is like one big database file that holds a lot of stuff.

To make that leap, first contemplate a traditional installation of an operating system: the operating system maps the blocks of hard disk space into a file system and prepares the drive to store files. The file system is the way files are organized, stored, and named. File storage is represented on the platters as magnetic patterns of noncontiguous blocks. If the blocks were made contiguous, or linked in a single arrangement, the sequential blocks of data could represent a single file nested in an even larger file system. The virtualization layer constructs a virtual hard drive using this approach, encapsulating each virtual machine disk into a single file on the host's physical hard drive. This is a strict virtual disk and is the abstraction of a hard drive.

Virtual disks are robust and mobile because the abstraction process encapsulates disks creating a file. Moving a disk is as easy as moving a file from a compact disc (CD) to a hard disk or floppy disk. You can achieve this despite the myriad of disk and controller manufacturers. A virtual disk, whether Small Computer System Interface (SCSI) or Integrated Drive Electronics (IDE), is presented to the guest operating system as a typical disk controller and interface. Virtualization software manufacturers employ just a handful of drivers for guest VMs when creating disks. By using a handful of rigorously tested drivers, VMs don't have as many problems as traditional operating system drivers. In the next section, you'll look closely at Microsoft's and VMware's implementations of virtual disk types.

Introducing Virtual Disk Types: Microsoft and VMware

VM disks come in several forms, such as physical, plain, dynamic, and virtual. Each disk type offers benefits and drawbacks. Determining which type of disk to use will be based on the function of the guest VM. Once again, be alert that Microsoft and VMware use different nomenclature to describe similar or identical disk types. Table 1-5 and Table 1-6 compare and contrast the disk naming conventions between the manufacturers for each virtualization application.

Table 1-5. *Virtual Disk Types*

Disk Type	Virtual PC	Virtual Server	Workstation	GSX Server	ESX Server
Virtual hard drive	×	×			
Dynamically expanding	×	×			
Fixed	×	×			
Linked	×	×			
Undo disk	×	×			
Differencing	×	×			
Virtual hard drive			×	×	×
Physical			×	×	×
Dynamically expanding			×	×	
Preallocated			×	×	
Independent persistent			×	×	
Independent nonpersistent			×	×	
Persistent					×
Nonpersistent					×
Undoable			×	×	×
Append					×

Table 1-6. *Functionally Equivalent Virtual Disk Types*

Disk Type	Virtual PC	Virtual Server	Workstation	GSX Server	ESX Server
Virtual hard disk/virtual disk	×	×	×	×	×
Dynamically expanding/dynamic	×	×	×	×	×
Fixed/preallocated	×	×	×	×	×
Linked/physical	×	×	×	×	×
Undo disk/undoable	×	×	×	×	×
Differencing	×	×			
Independent persistent/persistent			×	×	×
Independent nonpersistent/nonpersistent			×	×	×
Append					×

Virtual disks, whether configured as SCSI or IDE, can be created and stored on either SCSI or IDE hard drives. VMware guest operating systems currently support IDE disks as large as 128GB and SCSI disks as large as 256GB for Workstation and GSX Server. ESX Server can support a total of four virtual SCSI adapters each with fifteen devices limited to 9 terabytes (TB) per virtual disk. Microsoft's Virtual PC and Virtual Server offer you the ability to create up to four IDE disks as large as 128GB, and Virtual Server can support up to four SCSI controllers hosting seven virtual disks at 2TB. If you're doing the math, that's more than 56TB for Virtual Server and more than 135TB for ESX Server. Are your enterprise-class needs covered? Who's thinking petabyte?

You can also store and execute virtual disks over a network or on a storage area network (SAN)! In addition, you can store virtual disks on removable media: floppy disk, digital video disk (DVD), CD, and even universal serial bus (USB) drives for VMware VMs. Despite the portability of a virtual disk with removable media, you'll want to stick with faster access media to maintain reasonable performance. When creating virtual disks, it isn't always necessary to repartition, format, or reboot. You can refer to Tables 1-2 and 2-2 to view the disk media types that Microsoft and VMware support.

Virtual Hard Disk and Virtual Disks

The easiest way to think of a virtual hard disk is to think about what it isn't, a physical disk. Microsoft refers to its virtual disk as a *virtual hard disk* (VHD), and VMware refers to a virtual disk as a *virtual disk*. In either case, *virtual disk* is a generic term that describes all the disk types and modes of virtual disks utilized by VMs. In general, virtual disks consist of a file or set of files and appear as a physical disk to VMs. Microsoft VM disks end with a .vhd extension, and VMware VM disks end with .vmdk.

Dynamically Expanding and Dynamic Disks

When creating virtual disks, it's necessary to determine the maximum size of the disk you want to create. By not allocating the entire disk space initially, you can create what's referred to as a *dynamic disk*. Microsoft refers to a dynamic disk as a *dynamically expanding disk*, and VMware refers to it as a *dynamic disk*. Dynamically expanding and dynamic disks start out

small (just small enough to house the guest operating system) and grow to the maximum specified size as data is added to the guest OS. The advantages to using dynamic disks are that they consume a lot less real estate on your physical media, effortlessly move between physical systems, and are easier to back up. The trade-off for the convenience of being able to have dynamically expanding disks is that overall VM performance decreases. The dynamic disk setting is the default for Virtual PC and Workstation. Dynamic disks are great for using in educational labs, development and test environments, and demonstrations. Because dynamic disks tend to significantly fragment your physical hard drive, it's generally not a good option to deploy them in performance-sensitive environments.

Fixed and Preallocated Disks

Unlike dynamic disks, fixed and preallocated disks start out at their predefined size at the time of creation. Fixed disks are still virtual disks; they just consume the entire allotted disk space you specify from the beginning. Microsoft refers to predefined disks as *fixed*, and VMware refers to them as *preallocated*. This type of disk is the default for GSX Server. The advantage of using fixed disks is that space is guaranteed to the VM up to what you originally specified. Because the file doesn't grow, it will perform better than a dynamic disk. Unfortunately, it will still fragment your hard drive. You can use fixed disks in a production environment where high performance isn't critical, such as for a print server.

Linked and Physical Disks

Virtual machines have the ability to map directly to a host's physical hard drive or to a physical partition. VMware refers to this disk type as a *physical disk*, and Microsoft refers to it as a *linked disk*. Physical disks allow guest and host operating systems to concurrently access the physical hard drive. You should take care to hide the VM's partition from the host operating system to prevent data corruption.

You can use physical disks for running several guest operating systems from a disk or for migrating multiboot systems to a VM. When using physical disks to port operating systems to a VM, you should take extreme care. Porting existing operating systems to a VM is like swapping hard drives between different computers—the underlying hardware is different, and peripheral drivers are absent from the new marriage, which usually ends with an unbootable system.

VM physical disk usage is beneficial in terms of performance because the VM can directly access a drive, instead of accessing a virtual disk file. Using this disk type makes a lot of sense for performance-hungry enterprise applications such as a VM running Oracle or Exchange. If you're of the mind-set that anything you can do to tweak performance is a plus, physical disks are the way to go.

Undo and Undoable Disks

Undo means to revert to a previous state, and if you apply this to VMs, you get what's referred to as an *undo disk* for Microsoft and an *undoable disk* for VMware. Undoable disks achieve their magic because the guest operating system doesn't immediately write changes to the guest's disk image. Changes made during the working session are instead saved to a temporary file. When the current session terminates, the VM shuts down, and you're interactively prompted to save the session changes or to discard them. If you desire to save the changes,

you must commit the changes. Committing the changes merges the temporary file data with the original image. If you don't commit the session, the temporary file is discarded.

Undo disks are great to use if you're going to be making changes to a VM without knowing the outcome, such as performing an update, installing new software, or editing the registry. Undo disks are also good to use when it's necessary to have the original state of a virtual disk unchanged, such as in educational institutions, test environments, or kiosks.

Differencing Disks

Microsoft Virtual PC and Virtual Server have a disk type called *differencing*. Differencing uses a hierarchical concept to clone an existing disk image. Differencing starts with a baseline virtual disk image to create additional virtual images. These additional images may be referred to as *child disks*; they record the difference between the baseline parent disk image and the child differencing disk. Because multiple VMs can be created with only one parent disk, differencing saves physical space and time.

Differencing is handy for creating similar VMs and is useful for creating Web servers hosting different sites, identical file servers, or multiple application servers. When differencing disks are stored on a network, multiple users can access and use the disks simultaneously. When the sessions have ended, changes can be directed to the local computer for storage. As a precaution and because of the chained effect of differencing, remember to write-protect the parent disks: better yet, burn it to a CD-R/CD-RW or DVD-R/DVD-RW. Virtual Server will warn you that changes to the parent disk can cause corruption and may render parent and child disks useless. Differencing disks are good to use when you need to quickly roll out identical or near identical guest VMs.

Like Microsoft's differencing disk technology, VMware has the ability to share the base image of a virtual disk. Concurrently running VMs can use the same base image. Differences from adding and removing applications are saved to independent redo log files for each VM. Sharing a base image requires the following steps:

1. Create a VM with all necessary preinstalled software. This becomes the base image that won't change.
2. Next, use the undoable disk mode to create new redo logs.
3. Now, direct VMware to use the redo logs as the new disk for any number of virtual machines. The new VMs write to the redo file as a virtual disk and leave the base image unfettered.

Having the ability to share the same base image helps maintain a level of standardization, saves time, and saves disk space. After creating the base image, it should be write-protected and not edited.

Persistent and Nonpersistent Independent Disks

Independent disks require you to explore the concept of a *snapshot*. We all know that a snapshot is a photograph, and photographs trap a given point in time to capture an event. When applying the word *snapshot* to a VM, you're preserving the state of a VM at a given point in time. For instance, after installing and running a piece of software, the snapshot allows you

to roll back time as if the software were never installed. The VM is restored to the point in time when the snapshot was taken.

Persistent and nonpersistent are two disk types employed in VMware's virtualization products. Persistent and nonpersistent disks don't employ snapshots. When you direct VMs to use persistent disk mode, the virtual disk acts like a conventional physical hard disk. Writes are written to the disk without an option for undo and are made at the time the guest operating system writes or removes data. Persistent independent disks can't be reverted to a previous state after disk writes take place. Whether or not a snapshot is taken, the changes are immediate, and on system reboot the changes still exist. Persistent disks are good to use when you want to most closely approximate the function of a typical computer; for instance, you could benchmark the overhead of virtualization. Configure one computer like a typical computer, and configure the other for virtualization. The difference between system scores will provide the overhead involved with the virtualization layer.

Nonpersistent independent disks are generally associated with removable media. Despite when the guest operating system is running and disk writes take place, the changes to the disk are discarded, and the disk reverts to its original state on reboot. This is like reverting to a snapshot on every power cycle. Nonpersistent independent disks are an excellent way to distribute a preinstalled operating system with applications. Also, this is a great way to distribute software that's complicated to set up or where the known end-user environment goes unsupported.

Append Disks

Append mode disks are used with VMware ESX Server and are similar to undoable disks. A redo log is maintained during a running session. Upon termination, you aren't prompted to save the session's changes to the virtual disk. The system automatically commits the changes to the redo log. If at any time you want to revert to the VM's original state, delete the log file.

Append mode comes in handy when it isn't necessary to roll back session changes on each reboot but you'd like to have the option at some point in the future. Append disks are an excellent way to systematically learn the differences between two different states of a VM, such as in the study of network forensics. Because you have the ability to know the exact state of a system before and after an event, such as when figuring out how viruses or Trojans work, you can analyze the changes in individual files.

Resizing Disks

Sometimes it's difficult to size fixed and expanding disks for future growth in an enterprise. Traditionally, when you deplete your hard drive resources, you've earned yourself a format and reinstall. In the virtual world, the solution to increasing a VM's disk size is use-imaging software such as Norton Ghost. Power down your VM, and add a virtual disk that's the size you need. Boot the VM to your imaging software, and proceed to create a disk-to-disk image. When you're satisfied with the results, delete the undersized virtual disk.

Fixed disks generally offer performance close to that of a traditional operating system and are best suited for task-intensive applications. Fixed and expanding disks can also span several physical disk partitions. Because fixed disks consume their entire allocated space upfront, they take longer to back up and can be difficult to move to a different host. Dynamic disks offer smaller file sizes consuming less physical storage space, better portability, and easier backups. Because dynamic disks expand on the fly, performance decreases for disk writes.

Playing it safe with regard to dynamic and fixed disks can be difficult. When creating virtual disks, pay particular attention to the default disk mode Microsoft and VMware preselects for you: fixed or expanding. As a rule of thumb, server-class products default to fixed disks, and workstation-class products default to dynamic. You'll want to specify a disk type based on the VM's purpose. For instance, for demonstration and testing purposes, you may want to use dynamic disks. Dynamic disks won't needlessly consume space and will allow you to easily turn up several servers on restricted disk space. Unfortunately, dynamically expanding disks are slower and in the long run fragment hard drives, impacting performance. When performance is the major concern (as with production servers), fixed disks are always the best bet.

Introducing Networking

A network, at minimum, consists of interconnected computers or hosts. A *host* refers to a device that has an Internet Protocol (IP) address and uses networking protocols to communicate. An IP address is like a Social Security number because it uniquely identifies a single entity—a Social Security number identifies a citizen in the United States, and an IP address identifies a host (think: computer). IP addresses are grouped into two ranges and five classes: public and private, and A–E. Private addresses are prohibited from directly communicating on the Internet, but public addresses can.

Though it isn't necessary to memorize the categories and groups of IP addresses, it's a good idea to be aware of them. Understanding computer addresses is vital to your success in virtualization. Not knowing the two ranges and five classes makes it difficult to use Network Address Translation (NAT) and route virtualized computer traffic. In addition, you may want to break out your notes on subnetting before setting up virtual networks or studying for your VM certification. To serve as a quick reminder, Table 1-7 outlines the structure and uses of IPv4 classes.

Table 1-7. *IPv4 Classes*

Class	IPs for Private Range	IPs for Public Range	Subnet Mask	Purpose
A	10.0.0.0–10.255.255.255	1.0.0.0–127.255.255	255.0.0.0	Used on large networks
B	172.16.0.0–172.255.255.255	128.0.0.0–191.255.255.255	255.255.0.0	Used on medium networks
C	192.168.0.0–192.168.255.255	192.0.0.0–223.255.255.255	255.255.255.0	Used on small networks
D		224.0.0.0–239.255.255.255	255.255.255.255	Reserved for multicast traffic
E		240.0.0.0–254.255.255.255		Reserved for experimental use

VM Networking Protocols

Interconnected computers can exist that don't require IP addresses, but they aren't considered hosts unless they have an IP assignment (strictly speaking). VMs use virtual hardware to build networks that consist of network interface cards (NICs), bridges, and switches. Other than the computer language of the Internet that most networked computers use to communicate, Transmission Control Protocol/Internet Protocol (TCP/IP), VMs use Dynamic Host Configuration Protocol (DHCP) and NAT. In the following sections, we'll briefly cover these protocols before discussing how VMs handle these services.

TCP/IP

TCP/IP is a framework of globally agreed on directives that control communications between two hosts. TCP/IP is a suite of several protocols. TCP is responsible for monitoring the transfer of data, correcting errors, and breaking computer data into segments. The segmented data is forwarded to IP, and IP is responsible for further breaking the data into packets and sending it over a network. The conventions used by TCP/IP allow hosts to communicate reliably over the Internet and on typical computer networks.

Note Without giving yourself some context, *virtualization* will continue to be an abstract word. For instance, it may help if you start thinking about the theoretical approach to networking protocols with the aid of the International Standard Organization Open System Interconnect (ISO OSI) model. Even though you may think you've escaped the wrath of the OSI model after passing that last IT certification exam, it's a serious approach that will help you understand the concepts of virtualization. As you're reading this book, quiz yourself on what's currently being discussed, and note how it relates to one of the seven layers of the OSI model. Making yourself familiar with how the virtual equates to the physical will later help you troubleshoot VMs. For instance, when you have a host-only virtual network and two hosts aren't able to communicate, which VM device file are you going to troubleshoot?

DHCP

DHCP provides an automated way for passing network configuration information to hosts. DHCP can inform a host of many things, including its name, gateway, and IP address. Network administrators often use this protocol to centrally manage hosts, and using DHCP alleviates the necessity of physically keying information into every network computer (which can be tedious and time-consuming). In addition to reducing administrative overhead, DHCP makes it easy to physically move a computer from one network to another. Without an IP address, a computer can't connect to a network or the Web. Table 1-8 outlines the DHCP lease process; knowing this process makes it easier to troubleshoot VMs in host-only mode or VMs requiring the services of NAT.

Table 1-8. *DHCP Process*

Process State	Purpose
Discover	The host computer broadcasts an announcement to any listening DHCP servers.
Offer	All DHCP servers receiving the announcement broadcast an IP address offer to the host.
Request	The first offer to reach the host is the winner. The host broadcasts a request to keep the IP address.
Acknowledgment	The “winning” DHCP server acknowledges the host’s request to use the address.

VMware and Microsoft virtualization products are capable of providing a DHCP service for host-only and NAT-configured virtual networks. Like a traditional DHCP server, the virtual DHCP device allocates IP addresses from a pool of specified networking addresses. So that the VM will function correctly, the DHCP device assigns an IP address and any other necessary networking information including the NAT device information—default gateway and DNS server addresses. The DHCP server that’s bundled with Microsoft and VMware is limited. If you need a DHCP server for a test environment, the supplied server should be sufficient. If you expect your virtual machines in a host-only network to be bridged to a production network, you’ll run into some difficulties. For instance, if you’re relying on the DHCP service to register a host’s name in DNS, the bundled server isn’t up to the task.

You know that it’s important to not have multiple DHCP servers on a single network with conflicting lease information. In addition, you know that when using a DHCP server it’s important to not have its broadcast traffic cross into other networks. Packet leakage from virtual networks can occur if bridged network adapters aren’t properly configured. If a physical host receives an offer from the DHCP server supporting a virtual network, it won’t be able to communicate because it will be configured with information from the virtual network. This scenario is possible when using DHCP servers built into Microsoft and VMware virtualization applications if IP forwarding is configured on the host network adapter. In this case, though host-only and NAT networks are intended to keep their traffic in the virtual environment, packet leakage occurs if the physical host is actively forwarding traffic. This can occur because the host adapter is placed in promiscuous mode for NAT to route traffic from the outside physical network to the internal virtual network. In general, built-in DHCP devices are good to use when you don’t need the extended scope options provided by a production server.

NAT

The NAT protocol is a lot like nicknames. For instance, Alaska is known as the Last Frontier or Land of the Midnight Sun. Though the nicknames aren’t identical to the word *Alaska*, they directly translate to the uniquely identifying characteristics of the state. To further illustrate, it’d be difficult to confuse the Sunshine State with the Lone Star State. To draw another analogy, what language translators are to foreign diplomats, NAT is to IP addresses. NAT is like the aforementioned because it can uniquely identify a host across two networks: it’s the translation of IP addresses used within one network to a different IP address in another network.

Networking computers with NAT creates a distinction between internal and external networks. Traditionally, NAT maps a private local area network (LAN) to a global outside network. The mapping process of NAT creates a security barrier qualifying the movement of network traffic between the two. This is how nearly all corporate networks running on private LANs access the public Internet—by NATing either through a router or through a firewall. When NAT is used, networks can be any mix of public, private, internal, or external configurations. Figure 1-1 illustrates the NATing process.

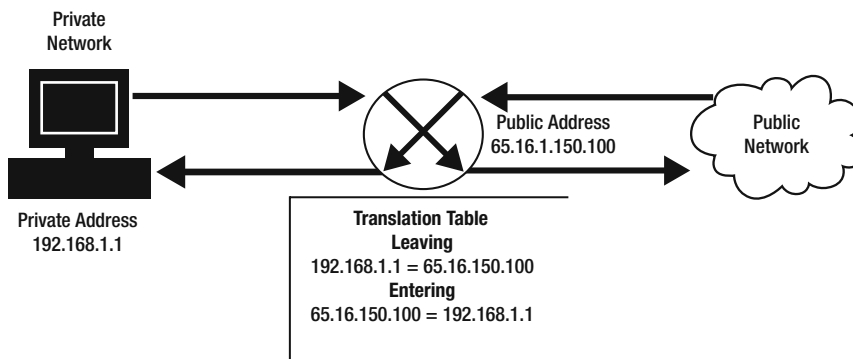


Figure 1-1. *The NATing process*

The terms *private* and *public* strictly refer to whether the network traffic generated by an IP address is capable of being routed on the Internet. If the address is public, it's routable on a private network and on the Internet; if it's private, it's routable on a private network and not the Internet. If Internet connectivity were required in the latter case, NAT would have to be used to translate the private address into a public address.

The NAT device connects VMs to external networks through the host's IP address. A NAT device can connect an entire host-only network to the Internet through the host's computer modem or network adapter. Using a NAT device tends to be the safest and easiest way to connect VMs to physical networks. The NAT device is responsible for tracking, passing, and securing data between host-only and physical networks. For instance, NAT identifies incoming data packets and forwards them to the intended VM. Conversely, NAT identifies and passes information destined for physical networks to the correct gateway.

The practical application of NAT devices built into Microsoft's and VMware's virtualization applications comes in handy when only the host's IP address is available to connect a virtual network to a physical network. You can leverage NAT to connect a virtual network to the Web through the host's network adapter or dial-up adapter using Microsoft Internet Connection Sharing (ICS). In addition, you can use NAT to connect virtual networks to your host's wireless adapter or even connect a virtual network to an asynchronous transfer mode (ATM) or Token Ring network with the aid of Microsoft Routing and Remote Access Service (RRAS) or proxy software.

Introducing Networking VMs

You probably graduated from sneakernet a long time ago, and whether you use wireless or traditional copper technology to network physical and virtual hosts, the same hardware and communication protocols you already know are required. VMs use physical and virtual NICs to create four common network configurations. Table 1-9 summarizes the virtual NICs used and maximum quantity available for use in Microsoft's and VMware's virtualization applications. The maximum number of virtual NICs to be installed in a guest VM can't exceed the number of available Peripheral Component Interconnect (PCI) slots, and the total that can be installed will be reduced to the number of free slots. For instance, if three virtual SCSI adapters are installed in GSX Server, despite that it can handle four network adapters, only three PCI slots are available for NICs.

Table 1-9. *Virtual Machine Network Interface Card Specifications*

Virtual Machine	NIC	Speed	Maximum	PCI Slots
Virtual PC 2004	Intel 21141	10/100	4	5
Virtual Server 2005	Intel 21141	10/100	4	5
Virtual Server 2005 Enterprise	Intel 21141	10/100	4	5
VMware Workstation	10/100	AMD PCnet-II	3	6
VMware GSX Server	10/100/1000	AMD PCnet-II	4	6
VMware ESX Server	10/100/1000	AMD PCnet-II	4	5

The following are the four virtual network types supported by Microsoft and VMware virtualization applications (sans the “no networking” networking option):

Host-only networking: Entire virtual networks can be created that run in a “sandboxed” environment. These networks are considered sandboxed because the virtual network doesn't contact existing physical networks. The virtual network isn't mapped to a physical interface and is called *host-only networking*.

NAT networking: NAT networks employ NAT to connect virtual machines to a physical network. This is achieved by using the host's IP address. NAT is responsible for tracking and delivering data between VMs and the physical LAN.

Bridged networking: Bridged networking is the process of connecting (*bridging*) a VM to the host's physical LAN. This is accomplished by placing the host's NIC in promiscuous mode. A bridged VM participates on the same LAN as the host as if it were a peer physical machine.

Hybrid networking: Using variations of host-only, NAT, and bridged networking, you can create complex hybrid networks. Hybrid virtual networks can simultaneously connect several physical networks and host-only networks. Hybrid networks are an excellent choice to create test environments.

Configuring the four types of networks varies from one VM application to the next. Though we'll discuss the required steps to create each network in later chapters, now is a good time to point out that varying degrees of difficulty (and lots of "gotchas") will be encountered as we step through the configurations of each. For example, Virtual PC and Virtual Server require you to manually install a loopback adapter on the host in order to allow the host to talk to the VMs on a host-only network. This is in contrast to VMware installing its own virtual network adapter on the host to do this automatically.

Connecting physical and virtual computers is a great way to share data, increase resource utilization, and learn. Knowing which network configuration to create, in the end, will be based on its purpose. If a network is being created to test and upgrade to a new operating system or roll out new software, it's best to keep experimentation confined to host-only networks. If host-only networks need access to external resources, NAT can be safely implemented to mitigate liability through limited communication to a physical network. On the other hand, if you're building highly reliable and portable VMs, physical/bridged networking is the way to go.

Note Host-only networks, because of their sandboxed environments, are a great way to emulate and troubleshoot production network problems, learn the basics of networking protocols, or teach network security with the aid of a sniffer (network analyzer). In the host-only sandbox, everyone wins: you aren't exposed to a production network or the traffic it creates in your broadcast and collision domains, and others are safe from your experimentation. For instance, with the aid of two to three VMs, you can quickly audit and identify the four broadcast messages involved with the DHCP lease process, or you can view the process of "stealing" information by performing a man-in-the-middle attack with the analyzers built into Microsoft operating systems, free Internet downloadable analyzers such as Ethereal, or commercial products such as Network Instruments' Observer.

Introducing Hardware

Computer hardware, for example, encompasses monitors, keyboards, RAM, CPUs, and sound cards. The word *hardware* should conjure up images of computer parts that can be touched and moved. Because VM software is capable of abstracting the physical computer into virtual hardware, fully loaded virtual machines can be created. VM manufacturers have successfully virtualized every major component of a computer, such as network adapters, switches, SCSI adapters, I/O devices, and even the BIOS.

Being that computers are generally useless without the ability to communicate with other computers, the ability of a VM to be networked is extremely important. Interconnecting VMs into the varieties of networks available requires using physical and virtual hardware. Traditional physical networks rely on network cards, bridges, switches, routers, and servers to get the job done, and VMs are no different.

Network Interface Card

A NIC, or network adapter, is a device that connects a host to a network. The network connection can be wireless, wired, or optical. An IP address is assigned to a NIC, which enables a host to communicate on IP-based networks.

In addition to a network adapter having an assigned IP, it also has a unique 48-bit address. This address is like a Social Security number in that it uniquely identifies one entity. As depicted in Figure 1-2, the address is expressed as an alphanumeric sequence in hexadecimal format and is referred to as a *media access control* (MAC) address. The first 24 bits of the number uniquely identify the manufacturer, and the last 24 bits identify the card. Hosts use MAC addresses to directly communicate with each other.

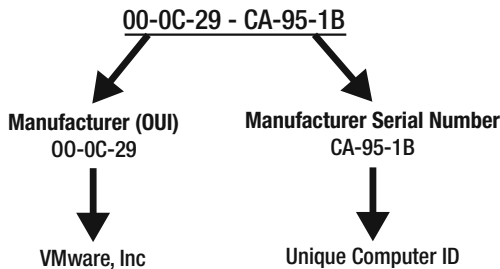


Figure 1-2. MAC address breakdown

Virtual networks cards, like physical adapters, have their own MAC and IP addresses. The guest operating system uses a virtual network card as if it were a physical NIC. NICs fall into categories: physical and virtual. The virtual NIC can be mapped to a dedicated physical network interface, or many virtual NICs can share a single physical interface. The VMM manages security, isolation of traffic, and guest NICs. Sending and receiving network-based information for virtualized computers generates an extra load for the host computer. To some degree, the increased overhead of networking will impact the overall performance of the host and guest operating systems.

Switches

You can think of the function of a *switch* in terms of the hub-and-spoke scheme of airline carriers. Passengers board an airplane at outlying airports and fly to distant interconnecting hubs. At the hub, passengers change planes and fly to their final destination. Air-traffic controllers orchestrate the entire process.

Switches connect multiple networking devices together to transport traffic, and hosts use a protocol called Carrier Sense Multiple Access/Collision Detection (CSMA/CD) to ensure no collisions take place. A collision occurs when two hosts use the same data channel simultaneously to send network traffic. If a collision is detected, the host will wait for a specified random period of time and attempt to retransmit the data. Microsoft and VMware both support VM virtual switches; the following lists highlight the limitations of the virtual switches included with each VM.

Virtual PC's virtual switches have the following limitations:

- Virtual PC supports nine virtual Ethernet switches; three are reserved for host-only, bridged, and NAT networking.
- Windows hosts can support an unlimited number of virtual network devices to a virtual switch.
- Linux hosts can support 32 devices.
- Virtual PC supports most Ethernet-based protocols: TCP/IP, NetBEUI, Microsoft Networking, Samba, Network File System, and Novell NetWare.

Virtual Server's virtual switches have the following limitations:

- Virtual Server supports nine virtual Ethernet switches; three are reserved for host-only, bridged, and NAT networking.
- Windows hosts can support an unlimited number of virtual network devices to a virtual switch.
- Linux hosts can support 32 devices.
- Virtual Server supports most Ethernet-based protocols: TCP/IP, NetBEUI, Microsoft Networking, Samba, Network File System, and Novell NetWare.

Workstation's virtual switches have the following limitations:

- Workstation supports nine virtual Ethernet switches; three are reserved for host-only, bridged, and NAT networking.
- Windows hosts can support an unlimited number of virtual network devices to a virtual switch.
- Linux hosts can support 32 devices.
- Workstation supports most Ethernet-based protocols: TCP/IP, NetBEUI, Microsoft Networking, Samba, Network File System, and Novell NetWare.

GSX Server's virtual switches have the following limitations:

- GSX Server supports nine virtual Ethernet switches; three are reserved for host-only, bridged, and NAT networking.
- Windows hosts can support an unlimited number of virtual network devices to a virtual switch.
- Linux hosts can support 32 devices.
- GSX Server supports most Ethernet-based protocols: TCP/IP, NetBEUI, Microsoft Networking, Samba, Network File System, and Novell NetWare.

ESX Server's virtual switches have the following limitations:

- ESX Server supports nine virtual Ethernet switches; three are reserved for host-only, bridged, and NAT networking.
- Windows and Linux hosts can support 32 devices.
- ESX Server supports most Ethernet-based protocols: TCP/IP, NetBEUI, Microsoft Networking, Samba, Network File System, and Novell NetWare.

Virtual switches connect VMs and are logically similar to physical switches. Switches place individual hosts into separate data channels, called *collision domains*, and are utilized in virtual or physical networking. The following are the types of switch devices available, based on the protocols employed:

- Bridged connects VM directly to the host's network.
- NAT uses traditional NAT techniques to connect the VM to the host's network.
- Host-only virtual machines are networked in a sandboxed network and don't affect the host.

Switches speed up the process of network communication by tracking the port and MAC addresses of connected hosts and storing the data in a table. When traffic is sent to the switch, the switch looks up the destination MAC address in the table and forwards the traffic to the associated port. If a switch doesn't know where to directly send incoming traffic, it will function like a hub and forward the traffic out all ports except the source port.

BIOS

A physical computer's BIOS is like the VMM in that it maps hardware for use by the operating system. Like physical machines, VMs have a BIOS. The BIOS maps your computer's hardware for use by software and functions the same for virtual machines as it does for a physical machine. It's a set of instructions stored in nonvolatile memory, a read-only memory (ROM) chip, or a variant thereof that interacts with the hardware, operating system, and applications. Moreover, BIOS software consists of a group of small software routines accessible at fixed memory locations. Operating systems and applications access the specific memory areas with standardized calls to have I/O functions processed. Without BIOS, a computer won't boot. Microsoft and VMware VMs are each packaged with different BIOSs:

- **Microsoft:** American Megatrends BIOS (AMIBIOS) APM1.2 and Advanced Configuration and Power Interface (ACPI) version 08.00.02
- **VMware:** Phoenix BIOS 4.0 Release 6–based VESA BIOS with DMI version 2.2/SMBIOS

The type of BIOS each manufacturer uses impacts the hardware that's inevitably available to the VMs, including the virtualized (synthetic) motherboard. Like physical machines, the BIOS of a VM can be edited. The ability to edit each VM allows you to customize the components available to the VM after boot. You may find it necessary to edit a VM's BIOS to change the boot device priority from a floppy disk to a CD, configure a bootup password to protect a VM from unauthorized access, or disable power management for VM production servers.

Because VMs will encounter the same problems as a physical machine, now is a good time to review the boot process of a computer. If a system fails to properly boot, knowing the boot process will help you quickly identify the cause of the failure.

The boot process begins with the BIOS when a computer initializes. After power is applied, the BIOS performs several tasks: starting the processor, testing hardware, and initializing the operating system. Let's quickly review the sequence:

1. When the computer is first turned on, the computer's power supply sends a signal to the main board informing it to initialize the CPU and load the BIOS bootstrap.
2. The BIOS checks the computer hardware through a series of small tests, called a *power-on self-test* (POST).
3. After an error-free POST, the BIOS searches for an operating system. Depending on how the BIOS is configured, any number of devices can be booted: floppy drives, hard drives, CD drives, and USB drives.
4. Once the BIOS locates a bootable medium, it searches for the boot sector to start the operating system.

Network booting is an additional option and is accomplished after the computer gets its IP address from the DHCP server. The client receives a bootable disk image over the network and executes it. Microsoft and VMware virtualization applications both support booting from a network adapter.

Generic SCSI

Generic SCSI is important to VMs being able to support the entire spectrum of hardware in the enterprise infrastructure. Generic SCSI affords VMs the ability to utilize SCSI devices, such as tape drives, DVD-ROM drives, CD-ROM drives, and even scanners. The prerequisite to a SCSI device working on a VM is that the guest operating system must support the device.

You can add generic SCSI support by giving VMs direct access to attached SCSI devices through the VMM. Though generic SCSI is supposed to be device independent, you may experience difficulties depending on the particular SCSI device and guest operating systems used. For instance, you might find your 12-head tape library appears as a single drive. At the time of this writing, only VMware virtualization applications support generic SCSI.

I/O Devices

Computers would practically be useless to us without our favorite I/O devices. What would we do without our optical mice, keyboards, color laser printers, and flat-panel displays? Virtualization software provides similar device support for getting data in and out of a VM as a physical computer; however, Microsoft and VMware vary on supported devices. The following list details what both manufacturers support natively:

- Virtual floppy drives
- Virtual serial ports
- Virtual parallel ports

- Virtual keyboard
- Virtual mouse and drawing tablets
- Virtual CD drives (CD-R only for Microsoft and CD-R/CD-RW for VMware)
- Virtual USB ports (keyboard and mouse emulation only for Microsoft)
- Virtual sound adapter (not available for Microsoft Virtual Server)

Introducing VM Products

Virtualization software is useful for developing new applications in different operating environments without having to simultaneously use multiple physical computers. In addition, because you can create complex networks, it's easy to test *what if* scenarios with operating system service packs and software application patches. More important, virtualization products give you excellent disaster recovery capability and extreme portability for the corporate infrastructure because a complete computer system is reduced to a file.

Microsoft and VMware have virtualization products in workstation and server varieties and are robust enough to use outside a test environment and in the enterprise. VMware and Microsoft offer a host of specialty products—for instance, VirtualCenter, P2V Assistant, and Assured Computing Environment (ACE)—that ease the administration of a virtualized infrastructure. In addition, a handful of open-source projects offer virtualization software and are well worth investigating and using.

Virtual PC

Virtual PC was originally created to emulate a PC in a Macintosh environment. Microsoft acquired Virtual PC from Connectix. Not only did Microsoft acquire a new product, it acquired the needed experience to get up to speed in the exploding field of virtualization. Does anyone remember what Microsoft accomplished with its \$50,000 purchase of Tim Paterson's QDOS? At any rate, Microsoft continues to support the Mac with Virtual PC and affords Mac users the ability to use PC-based applications, file structures, and peripheral devices. Virtual PC for Mac is well worth the minimal investment for the luxury of running PC operating systems and applications. If you've used Connectix's product in the past, you'll find Virtual PC to be the same powerful virtual machine it has always been—for the Mac or the PC.

Virtual PC is hosted by a conventional operating system, and once Virtual PC is loaded, guest operating systems can be installed. Guest operating systems are controlled like normal Microsoft applications: with a click of the mouse, you can switch between Windows 9x, Windows NT, and even DOS. Being able to run legacy operating systems in a virtual environment maintains compatibility with an established infrastructure without having to delay the migration to more advanced operating systems.

Because Virtual PC affords developers the ability to run several operating systems simultaneously on a single physical machine, development time and overall hardware costs are reduced. Remember that the more operating systems you want to run at one time, the more RAM you need to maintain reasonable levels of performance. Every virtual machine created in Virtual PC will run in an isolated environment and uses standardized hardware. To run Virtual PC, your host system will need to comply with minimum specifications. Table 1-10 lists the hardware and operating system requirements.

Table 1-10. *Microsoft Virtual PC Minimum Supported Host Specifications*

Requirements	Minimum Specification
CPU	400MHz
SMP	Host-only
Memory	128MB
Video	8-bit adapter
Disk space	2GB
Hard drive	IDE or SCSI
Network card	Optional
Audio	Optional
Host OS support	Windows XP or Windows 2000
Guest OS support	OS2, DOS 6.22 through Windows XP Professional

Connectix Virtual PC officially supported more operating systems than Microsoft Virtual PC; however, you can continue to “unofficially” run several varieties of Linux operating systems successfully. Be aware that when you run unsupported operating systems, you’re on your own when it comes to support. It’s important in the enterprise to have documented manufacturer support, so choose virtualization software accordingly. Microsoft continues to support OS/2 in the Virtual PC environment.

Microsoft’s Virtual PC is a mature product and suitable for running multiple operating systems, using legacy operating systems, training new employees, or allowing corporate technical support to test rollout and migration plans. In addition, Virtual PC is excellent for training technical students on virtual networks using multiple operating systems without impacting the entire school or when acquiring additional hardware is cost prohibitive.

One glaring difference between Virtual PC and Virtual Server is the availability of official Microsoft support. Virtual PC is intended to be used with desktop operating systems, and Virtual Server is intended for server operating systems. You can load server operating systems on Virtual PC; however, you’re on your own for support. If you’re using it for a test environment, then support isn’t a big deal. In the physical world, would you use a desktop operating system in place of a server operating system? Also, be careful when moving virtual machines—the saved states between Virtual Server and Virtual PC are incompatible. Table 1-11 lists some other differences between the two.

Table 1-11. *Differences Between Virtual PC and Virtual Server*

Virtual Machine	Sound Card	Virtual SCSI Support	CD-ROM Drives
Virtual PC	Yes	No	One
Virtual Server	No	Yes	Many

VMware Workstation

Microsoft Virtual PC and VMware Workstation are in the same single-user VM class and have many similarities. Similarities between the two products are uncanny and include juicy stories with regard to mergers and acquisitions. To fill you in a bit, after a brief courtship by Microsoft ending in failure, EMC stepped up to the plate and acquired VMware. Since its inception at Stanford in the late 1990s, VMware Workstation has become a mature and reliable product.

VMware Workstation, despite having a single-user license, is excellent for network administrators and developers who want a better, faster, and safer way to test in production environments. On the development side, programmers can test code across multiple platforms on one workstation, ensuring application compatibility. Administrators can test complex network configurations using a variety of network operating systems, including Microsoft, Novell, and Linux products. In addition, administrators can create virtual networks to safely test how a new patch or upgrade will affect production systems.

VMware accomplishes hardware virtualization by mapping physical hardware to virtual machine resources. Every virtualized host is a standard x86 system with its own virtualized hardware. For example, each VM will have its own memory, CPU, I/O ports, hard disk, CD drive, and USB ports. Like Virtual PC, VMware Workstation is hosted by a conventional operating system. Workstation officially supports more than 20 operating systems and “unofficially” runs many others. Because Workstation can support applications requiring legacy operating systems, migrating to new operating systems and hardware won’t impede enterprise workflows. Table 1-12 lists the minimum hardware and operating system requirements to run VMware Workstation.

Table 1-12. *VMware Workstation Host Specifications*

Requirements	Minimum Specification
CPU	400MHz
SMP	Host-only
Memory	128MB
Video	8-bit adapter
Install footprint	100MB for Windows/20MB for Linux
Hard drive	IDE or SCSI
Network card	Optional
Audio	Optional
Host OS support	Windows NT4 SP6a through Windows 2003 Datacenter Linux distributions including Mandrake, Red Hat, and SuSE
Guest OS support	DOS 6.x through Windows 2003 Enterprise, Linux, Solaris, Novell, and FreeBSD

When you want to begin transforming the business infrastructure into a virtual data center or support the increasing demands on educational equipment, VMware Workstation is the best place to begin. Workstation will reduce overall costs, decrease development time, and provide programmers, network administrators, and students with an overall excellent test bench.

Microsoft Virtual Server 2005

Microsoft expands its virtual machine portfolio with the addition of Virtual Server 2005. Enterprise-class virtual machines, such as Virtual Server 2005, typically provide value by offering the ability to consolidate multiple servers onto one piece of physical hardware, the opportunity to re-host legacy applications, and increased assurance with regard to disaster recovery capability.

Virtual Server comes in two editions, Enterprise and Standard. Both editions are identical except for the number of processors each supports—four for Standard and up to thirty-two for Enterprise. Microsoft's Virtual Server is a hosted solution designed to run on Windows Server 2003 and Internet Information Services (IIS) 6 with ASP.NET-enabled extensions. Virtual Server can run many x86 operating systems; however, Microsoft officially supports the following guest operating systems:

- Windows NT Server 4.0 with Service Pack 6a
- Windows 2000 Server
- Windows 2000 Advanced Server
- Windows Small Business Server 2003, Standard Edition
- Windows Small Business Server 2003, Premium Edition
- Windows Server 2003, Web Edition
- Windows Server 2003, Standard Edition
- Windows Server 2003, Enterprise Edition

Table 1-13 lists the minimum specifications for the guest operating systems.

Table 1-13. *Virtual Server Minimum Specifications for Guest Operating Systems*

Guest Operating System	RAM	Disk Space
Windows XP Professional	128MB	2GB
Windows XP Home	128MB	2GB
Windows 2000 Professional	96MB	2GB
Windows NT 4 SP6 or higher	64MB	1GB
Windows Millennium Edition	96MB	2GB
Windows 98 SE	64MB	500MB
Windows 95	32MB	500MB
MS-DOS 6.22	32MB	50MB
OS/2 (with exceptions)	64MB	500MB

Because you'll be left to support unlisted operating systems such as Linux on your own, Virtual Server may not be a good choice in production environments for unlisted operating systems. In addition, if your VM implementation requires multiple processors, Virtual Server 2005 will leave you in the cold—it will virtualize only a uniprocessor configuration for any given VM. Table 1-14 outlines the minimum hardware and software requirements for Virtual Server.

Table 1-14. *Microsoft Virtual Server Minimum Supported Host Specifications*

Requirements	Minimum Specification
CPU	550MHz
SMP	Host-only
Memory	256–512MB Small Business Server (SBS)/512MB Datacenter 512MB
Video	8-bit adapter
Disk space	2–4GB (4GB for SBS)
Hard drive	IDE or SCSI
Network card	Optional
Audio	Not supported
Host OS support	Windows SBS 2003 through Windows 2003 Datacenter
Guest OS support	OS2, DOS 6.22 through Windows XP Professional and Windows 2003 family

Virtual Server has an easy-to-use Web interface where system resources can be leveraged to allocate resources to guest operating systems as necessary. If a guest needs a bit more processing time from the CPU or more RAM, simply adjust Virtual Server, and your problem is solved. The scripted control of guest systems makes managing the systems straightforward, and it has the characteristic Windows look and feel. Virtual Server also integrates with Windows tools such as Microsoft Operations Manager (MOM) 2005, which is a performance and event management tool, and Systems Management Server (SMS) 2003, which is a software change and configuration management tool.

Getting your existing servers into a virtual environment will be the first hurdle in the virtualization process. If all the servers in your infrastructure had to be formatted and reloaded, making the leap from a physical environment to the virtual would be a near impossible task. To aid in the migration, the Microsoft Virtual Server Toolkit (MVST) is available as a free download as of this writing. It automates the process of moving the server operating system and applications from a physical machine to a virtual machine running in Virtual Server.

Virtual Server is a solid addition to Microsoft's line of products and affords homogenous corporate environments the ability to stay that way. Though you'll have to use an additional server license to run Virtual Server guest operating systems, it's an excellent way to start saving by consolidating servers and to free time for support departments by reducing administrative overhead. Virtual Server is easy to install and offers traditional Microsoft quality in its management of system resources.

Licensing may pose a hidden gotcha when it comes to software pricing. Some software vendors base licensing fees on CPUs, but what happens if you run a single CPU-licensed application on a quad processor box in a guest VM utilizing one virtual processor? You'll want to check with your software vendors on the ramifications of using applications in VM

scenarios. Don't get put in a position of pirating software. For the Microsoft folks, Microsoft has a licensing brief available that's well worth reading. As of this writing, you can download it at http://download.microsoft.com/download/2/f/f/2ff38f3e-033d-47e6-948b-8a7634590be6/virtual_mach_env.doc. The good news is that Microsoft doesn't require redundant client access licenses (CALs) for guest VMs and host VMs. You'll need only application CALs for guest VMs because every user and device required to have an OS CAL must access the host operating system running Windows 2003 first.

VMware GSX Server

Like Microsoft's Virtual Server, GSX Server can serve in the enterprise and affords similar implementation benefits. GSX Server is x86-based and is great for server consolidation, for expediting software development practices, and for disaster recovery purposes. Because GSX Server is flexible, it supports a wide variety of server products, such as Microsoft Server products, Linux products, and even Novell NetWare.

GSX Server allows administrators to remotely manage VMs and provision new servers through a Web interface based on Secure Sockets Layer (SSL) or through the host's console. Each new guest server is isolated and hosted on the same physical box, and communication between the systems will be facilitated through the use of virtual or bridged networking. Provisioned guest VMs can have direct or mapped access to host machine resources, such as memory, CPU, networking devices, peripherals, and disk drives. GSX Server hosts can run on symmetric multi-processor (SMP) systems; however, guest VMs can use a single processor. If you need scalability for guest operating systems, you'll have to move up to ESX Server. Table 1-15 lists the minimum hardware and host operating system requirements for GSX Server.

Table 1-15. *VMware GSX Server Minimum Supported Host Specifications*

Requirements	Minimum Specification
CPU	733MHz
SMP	Host-only
Memory	512MB
Video	8-bit adapter
Install footprint	130MB for Windows/200MB for Linux
Hard drive	IDE or SCSI
Network card	Optional
Audio	Optional
Host OS support	Windows SBS 2003 through Windows 2003 Datacenter Linux distributions including Mandrake, Red Hat, and SuSE
Guest OS support	DOS 6.x through Windows 2003 Enterprise, Linux, Solaris, Novell, and FreeBSD

GSX is a hosted application and can be installed on Windows or Linux operating systems. Hosted virtualization software means you'll sacrifice an operating system license to host guest VMs: this is true for Microsoft virtualization products and VMware products except ESX Server. If you're using all open-source products, then the cost of a license may be less important. If you

have a heterogeneous environment, GSX Server will be your product of choice because of VMware's extensive operating system support.

If you want to start rolling out virtual servers in a production environment, GSX Server is the easiest way to begin adding value and ease of management to your infrastructure. GSX Server can be loaded onto an existing system without requiring you to invest in an additional license or in more hardware. You may also want to start by not attacking critical systems first. Try virtualizing a print server, application, or Web server first. Once you get the hang of GSX Server system controls, you can attack the bigger projects and even look into the enhanced benefits of ESX Server.

VMware ESX Server

VMware ESX Server is a seasoned and robust virtualization software package. It's designed to create data center-class, mission-critical environments. ESX Server, like Virtual Server, can help consolidate the infrastructure, provide easier management, and increase resource utilization. ESX Server treats the physical host as a pool of resources for guest operating systems by dynamically allocating resources as necessary. All resources can be remotely managed and automatically allocated. Table 1-16 lists the minimum hardware requirements for ESX Server-hosted operating systems.

Table 1-16. VMware ESX Server Minimum Supported Host Specifications

Requirements	Minimum Specification
CPU	900MHz
SMP	Host and guest supported (requires additional software)
Memory	512MB
Video	8-bit adapter
Install footprint	130MB for Windows/200MB for Linux
Hard drive	SCSI disk, RAID logical unit number (LUN), or Fibre Channel LUN with unpartitioned space
Network card	Two or more Intel, 3COM, or Broadcom adapters
Audio	Optional
Guest OS support	DOS 6.x through Windows 2003 Enterprise, Linux, Solaris, Novell, and Free BSD

ESX Server separates itself from the rest of the pack with its ability to cluster VMs across physical systems and to provide scalability for guest operating systems with SMP support. Unfortunately, to have the advantages of SMP, you have to purchase VMware's virtual SMP product (Virtual SMP). VMware Virtual SMP, despite being an add-on module, truly scales the infrastructure by allowing for the division of multiple physical processors across many virtual hosts. ESX Server really leverages existing hardware that's underutilized and easily handles resource-hungry applications, such as Exchange and Oracle. Moreover, you can increase the value of ESX Server by creating a virtual infrastructure with the aid of VMware's additional tools and management packages: P2V Assistant, VMotion, and Virtual Center.

Virtual Infrastructure

You learned about the process of abstraction earlier in this chapter, and now you'll apply the term to utility computing, or *computing on demand*. To make service and application software immediately available (utility computing) in the enterprise, you have to treat the entire infrastructure as one resource, and you therefore need an extra layer of abstraction.

To achieve this layer of abstraction between networking, storage, and computing resources, VMware uses several products to create a virtual infrastructure node (VIN). A VIN consists of an ESX Server loaded with VMware VirtualCenter, VMotion, and Virtual SMP. VMs cobbled from the varying systems in the enterprise infrastructure appear to be a single, dedicated resource.

VMware VirtualCenter and VMotion

VMware VirtualCenter is a set of globally accessible management tools that will give you a single administration point for ESX Server and GSX Server VM resources. Because VirtualCenter can provide immediate relocation of resources, end users experience virtually no downtime with respect to services and applications. VirtualCenter is able to accomplish this by using a database backend.

VirtualCenter requires one of three different databases to function—Oracle 8i, SQL 7/2000, or Access—and it maintains a copy of each system's environment. In addition, VirtualCenter can run prescheduled tasks such as rebooting guest VMs. VirtualCenter, running as a service on Windows 2000, Windows 2003, and Windows XP Professional, uses a single dashboard-type interface that continuously monitors performance and system resources. You can equate VirtualCenter to a virtual keyboard-video-mouse (KVM) switch that's capable of restoring and provisioning servers.

VMware VMotion can move active VMs between running host systems. With VMotion, hardware maintenance no longer impacts production environments: prior to maintenance, simply move the live VM to a different host with VMotion. To take advantage of VMotion, you need to have previously created a VIN, have a VirtualCenter Server, and have a minimum of two ESX Servers loaded with VirtualCenter, VMotion, and Virtual SMP. VMotion is responsible for optimizing performance by balancing resources through workload management. Changes to systems don't impact the end user.

VMware P2V Assistant

VMware P2V (read: Physical-2-Virtual) Assistant is a migration tool. It creates an image of a physical system and transforms it into a VM. P2V Assistant creates an image of the physical system using built-in or traditional imaging systems, such as Norton Ghost. P2V saves time by eliminating the need to reinstall existing operating systems and applications, and created VMs can safely run Workstation, GSX Server, and ESX Server. P2V uses an intuitive graphical user interface (GUI) and begins the migration by creating a complete image of the physical system. Next, P2V substitutes physical drivers for virtual. Lastly, P2V makes recommendations you may need to act on prior to booting the system. Currently, P2V supports only Microsoft operating systems, and VMware recommends you purchase a rather pricey support contract before attempting to virtualize a system via P2V Assistant.

Migrating Between VMs

Though the following procedure isn't something you'd want to perform and place in production, it certainly is a good way to exhibit the power of virtualization and disk imaging in a lab or demonstration. As a last resort, it may bail you out of a tight spot. When deciding on which virtualization product to standardize on in your infrastructure, you may be overwhelmed with the choices. Microsoft supports its operating systems in Virtual Server, and VMware supports the ever-increasing market of open-source operating systems. Is it a toss-up?

Well, if it's a toss-up and you later change your mind on which virtualization software to implement, a migration route exists that may help you "switch teams" using disk-imaging software such as Norton Ghost. For instance, you can use Norton Ghost to migrate from a Windows VMware VM to a Windows Virtual PC VM. Simply use the imaging software to create an image, and then restore it to your desired destination. The only difficulty you may run into in the restoration process is that Microsoft's hardware abstraction layer (HAL) may get feisty. If this is the case, you'll need to copy the original HAL from your OS installation medium. For instance, you can boot off a Windows XP disk to the Recovery Console and copy the HAL over. After booting your machine, Windows will autodetect the new VM hardware.

VMware ACE

VMware's ACE provides an isolated end-user VM. ACE allows administrators to create VMs strictly adhering to corporate policies and procedures that can be deployed to the end user through removable media or over the network (think: abstracted BOOTP and Norton Ghost images). The ACE is implemented by creating a VM loaded with an OS and any necessary applications. ACE control policies are then applied to the packaged VM, which can then be rolled out. Provisioning VMs and installing ACE is one seamless installation as far as the end user is concerned. Some may see ACE as being similar to Citrix because administration occurs from a central location, but ACE differs in that each user has their own complete x86 VM. With Citrix, all users share the same hardware. It's possible for an individual user session in Citrix to impact other users on the system by depleting physical resources. Because ACE requires its own hardware, no user can deplete the local resources of another.

Summary

In this chapter, we covered everything from basic hardware and networking to commercial and open-source VM applications. You now have a good idea of how hardware can be virtualized through emulation and mapping techniques and can readily visualize an entire VM as a database with lots of stuff in it. More important, you recognize the value of virtualization software in the enterprise and in education—virtualization can protect networks from disasters, quickly deploy new systems, decrease administrative overhead, leverage existing hardware investments, and simulate entire networks.

In Chapter 2, you'll learn how to prepare a host system for virtual machines, and you'll look at the many gotchas and common pitfalls that often go unnoticed. Then, with your understanding of VM nomenclature and concepts, you'll be ready to tackle Chapter 3 and the rest of this book.



Preparing a Virtual Machine Host

In Chapter 1, we visited important issues surrounding virtualization technology, including hardware, software, virtualization theory, and manufacturer nomenclature. This chapter will address the planning issues for selecting and preparing a host system to support virtual machines. You'll need to confirm that your host computer meets the VM application's minimum hardware requirements, and you must verify that it has the available resources to support the number of VMs you plan to run simultaneously. You'll look at preparation issues for both Windows and Linux hosts. Several factors determine the readiness of a computer to support a VM application, such as the motherboard, RAM type, CPU speed, hard disk type, and network adapter speed. Using best-practice principles, we'll cover how to quantify host requirements that will result in better stability, scalability, and performance for your guest virtual machines.

Implementing Best Practices

You'd think with RAM, processing power, and storage space being so inexpensive it'd be fairly easy to achieve some type of best-practice hardware implementation in today's modern IT infrastructure. Unfortunately, this has never been the case. Budgets, personal preference, and "we always do it this way" all take a toll on the idea of a best practice. Though some may say a best practice is what works for you, don't buy into that definition. A best practice is always the high road—the road less traveled. A best practice is a radical deviation from "if it ain't broke, don't fix it." In general, best practices describe the activities and procedures that create outstanding results in a given situation and can be efficiently and effectively adapted to another situation. For instance, for us IT people, this means if studies and white papers find that commercial-grade servers have less downtime and problems than white boxes, we should quit building our own servers for the enterprise infrastructure. If studies and reports prove that virtualization truly saves money, decreases administrative overhead, and better supports disaster recovery strategies, we should implement it.

In test and educational environments, using the minimum available resources to learn how to use VM applications, such as Microsoft's Virtual PC and VMware's Workstation, is okay. In these environments, the luxury of time is on your side, and occasionally rebooting is no big deal. In a production environment, though, end users don't have time to wait, and rebooting any major system can mean the loss of real money. Given this, take a moment to look at best-practice hardware requirements for each VM application before deploying VMs in your

infrastructure. Take the time to check your hardware against each manufacturer's HCL, and compare your hardware to the listed best practices in each section. Don't be a victim: spend a little time now checking out your deployment hardware; it will save you late nights and week-ends later. Simply loading Microsoft's Virtual Server or VMware's GSX Server on an existing system and "letting it eat" is a recipe for disaster.

Evaluating Host Requirements

When it comes to hosting VMs, bigger is always better. However, stuffing quad processors and a Fibre Channel SAN into a notebook isn't an option either. (We'll give you an in-depth look at VMs and SAN interactivity in Chapter 13, but we'll touch on the basics in this chapter.) Now, let's take a moment to put some perspective on the saying "bigger is always better." Host hardware—whether laptops, workstations, or servers—all perform different functions and will have very real and different needs, so focus your budget on where it will produce the most overall good. You can avoid bottlenecks by investing equally in a host's three major systems:

- **Storage systems:** Hard drives, SANs, CD/DVD-ROMs, RAM, and cache
- **Networking systems:** NICs, switches, and routers
- **Processing systems:** CPU and front-side bus (FSB)

When sizing a computer to host VMs, you can easily pinpoint areas of concern by reflecting on the system's purpose and keeping the three major subsystems in mind. For instance, laptops are great because they can be carried everywhere and utilized in sales presentations or classrooms. For the convenience of shoving a mainframe system into the space of a book, you can make performance and hardware longevity sacrifices. Because the space constraints put on portable computers cause them to overheat, manufacturers use lower-spindle speed hard disks, variable-speed processors, and smaller memory capacities. In this case, you'd want to spend money on adding as much RAM as possible to your system, followed by as much processing power as possible.

Workstations, on the other hand, have the dubious honor of being shin-bangers. The purpose of a workstation has stayed consistent over time, and you know they provide fast, reliable access to centralized network resources. Workstations employ higher-speed disk drives, larger quantities of RAM, faster processors, and good cooling systems. To improve VM workstation hosting, you'll want to add additional disk drives and controllers, followed by more RAM quantities rather than add a second processor. Lastly, add a good, fast NIC. By placing the host OS on one bus and hard drive, and the guest OSs on a second bus and hard drive, you more closely approximate the structure of two computers in one physical box; this creates a much better host.

Servers are different from laptops and workstations in that users will be depending on VMs: accessibility and availability will be your highest priority. You'll want to use Redundant Array of Inexpensive Disks (RAID)—configured hard drives (SAN if possible), several-gigabit NICs in a teamed configuration, multiple processors, and maximum RAM quantities. In case your mind is already racing ahead with *what ifs* (and to feed the supergeek in us), we'll cover more closely the mathematic justification for using multiple controllers and disks in the "Considering Storage Options" section toward the end of this chapter. We'll also discuss hard disk and RAID selection.

Table 2-1 is designed to help you begin prioritizing your VM hosting needs. Keep your eye on what type of VM host is being built, and then make decisions that will adequately support the host in its final environment. Table 2-1 summarizes our recommended priorities, but you can prioritize resources according to your needs.

Table 2-1. *Comparing VM Host System Priorities (1 = High, 6 = Low)*

Resource	Laptop	Workstation	Server
Multibus disks		1	2
RAID/SAN			1
RAM	1	2	5
Processing speed	2	3	6
Multiple processors		5	4
Networking capacity	3	4	3

When prioritizing investment and resource allocation for VM hosts, you want to determine what's going to be the subsystem bottleneck and mitigate its impact on performance by adding bigger, better, and faster technology. Bottlenecks will manifest based on the host's purpose:

- Will it be a portable or fixed system?
- Are resources local or available across networks?
- Is the host serving one person or one thousand?

Later in the “Considering Your Network” and “Considering Storage Options” sections, we'll further discuss the impacts of direct-attached storage, SANs, teaming, and load balancing.

All the best-practice suggestions for sizing host systems will do you absolutely no good without applying a good dose of common sense and experience—reference manufacturer OS system requirements, and check hardware compatibility. This is particularly true with regard to memory quantities and disk space. On any given server, the host and each VM both require sufficient memory for the tasks each will perform. You can calculate your total memory needs by simply calculating the number of VMs to be hosted, adding one for the host, and multiplying the sum by no less than the OS's minimum memory requirement. For example, if you plan to host three Windows 2003 Web servers on one piece of hardware, you need enough RAM to support four servers. But what version of Windows 2003 are you using? Windows 2003 has different minimum and maximum supported memory configurations, as listed in Table 2-2.

Table 2-2. *Windows 2003 Server RAM Requirements*

Memory Requirement	Web Edition	Standard Edition	Enterprise Edition	Datacenter Edition
Minimum	128MB	128MB	128MB	512MB
Recommended	256MB	256MB	256MB	1GB
Maximum	2GB	4GB	32GB/64GB Itanium	64GB/512GB Itanium

Assuming the host is using Windows 2003 Standard, you'll want to install 2GB of RAM (4 servers × 512MB of RAM = 2GB). We'd also like to point out that if you're hosting VMs on Windows 2003 Standard with Virtual Server or GSX Server, you'll be limited to a total of 4GB of RAM for the host and its guest VMs. If you plan on taking full advantage of larger memory pools, you'll want to move up to VMware's ESX Server or Windows 2003 Enterprise.

Determining required disk space is similar to sizing memory but with a few added gotchas. You'll need to include hard disk space for each guest's operating system footprint, for memory swap files, for dynamic disks, and for fixed disk storage. You'll also need to include it for suspending running VMs (equivalent to the RAM allocated to the VM). For a disk-sizing example, let's use the three previously mentioned Web servers configured with fixed disks. Table 2-3 lists the Windows 2003 install footprint for each edition.

Table 2-3. *Windows 2003 Disk Space Setup Requirements*

Web Edition	Standard Edition	Enterprise Edition	Datacenter Edition
1.5GB	1.5GB	1.5GB/2GB Itanium	1.5GB/2GB Itanium

You'd need enough disk space to adequately cover four server OS installations (1.5GB × 4 = 6GB), room for four swap files (1GB × 4 = 4GB), fixed disk data storage for four servers (4 × ?), and room for three guest VMs to be suspended (512MB × 3 = 1.5GB). At the moment, you'd need a minimum of 11.5GB just to get the servers loaded and running. The one variable that may be difficult to contend with is the space for data storage: being that needs vary, you'll need to rely on your experience and common sense for fixed disk space sizing. Assume that the servers in this example will store graphics-intensive sites and need to house documents for downloading (3 × 40GB = 120GB). You'll set aside a total of 8GB for the host and its applications. After calculating fixed storage and adding the install, suspend, and swap requirements, your host will need approximately a minimum 140GB of storage. If you're implementing an IDE solution, mirrored 160GB hard drives will be adequate. A SCSI RAID 5 solution can squeeze by with three 80GB hard drives. If you think any of your VMs may run out of space, you'll want to consider adding hard disk space so you can create additional fixed or dynamic disks in the future.

It's impossible to consider every scenario in which VMs will be installed. Therefore, you'll have to rely on the manufacturer's minimum requirements, on postings from Internet forums, on trusted colleagues, and on your experience. In Chapters 3 and 5, where we cover installing virtualization applications, we'll suggest best-practice minimums. The best-practice minimums offered are merely a guide to point you in the correct direction for sizing a VM host system.

Selecting a Motherboard

The motherboard, the centralized circuit board all other devices and chipsets connect to, interacts with every subsystem of a computer. It's largely responsible for an entire computer's performance, stability, and scalability. Choosing a motherboard to use for your VM host may be less of an issue if you're purchasing a system from a major vendor, such as Dell, Sony, or Hewlett-Packard. However, you'll need to make sure the motherboard supplied with a

proprietary system supports your intended purpose and will scale as needed. Selecting the correct motherboard will go a long way to ensuring that your VM host performs optimally. Additionally, the right motherboard may reduce some of the cost involved by including integrated NICs, hard drive controllers, video adapters, and sound cards.

Motherboards can be complicated and conceivably the most important part of any computer. A plethora of motherboard manufacturers exist, so carefully consider your available options and narrow the field of choices by determining if a particular motherboard supports the correct memory, CPU, I/O device options, and reliability constraints you need. You can buy motherboards for less than \$20 and spend as much as several thousand. You don't have to know the intricacies of every chipset and processor made, but you should know enough to make sure your requirements and expectations are satisfied.

Good performance levels are critical for ensuring that data processing takes place in a reasonable amount of time. But what's performance without stability and reliability? You'll want to stick with motherboard manufacturers that consistently produce quality merchandise over time and continue to support EOF products: you'll eventually need to update a BIOS or driver to fix a bug or to upgrade when a new OS or system patch is released. Because motherboard drivers are provided in part by the chip manufacturer and the motherboard producer, it's critical you pick a well-known and respected brand to get the most out of available support—Microsoft doesn't build every driver in its operating systems. Because reputable motherboard manufacturers will have already taken the time to thoroughly test their motherboards for stability and reliability over a sizable chunk of time, using respected brands will save you many hours of research and testing time.

Determining hardware compatibility for motherboards can be difficult because of the open nature of computers. Peripheral components that have been in use for longer periods of time in the field are more likely to be supported by reputable manufacturers, such as Adaptec, Intel, QLogic, ATI, and 3Com. If you need a particular device to function with a new motherboard, you'll need to check and verify that the manufacturer has tested and supports the device. Moreover, read the fine print in the warranty. Some manufacturers may void your hardware warranty for not using proprietary or approved hardware.

So, given the preceding rhetoric on buying a quality motherboard, what should you have on your checklist to help narrow the field to create the best possible foundation for your VM host? During the motherboard selection process, you'll want to specifically look at the following:

- CPU speed and quantity
- Controller chipset
- Memory requirements
- Bus types
- Integrated devices
- Board form factor
- Overall quality

We'll cover each of these in the following sections.

CPU Speed and Quantity

The speed and quantity of processors will significantly impact the overall performance of your host. If you're building a production server, you'll definitely want to use multiple processors. Microsoft virtualization applications don't support SMP for guest VMs, but the host can take full advantage of multiple processors (assuming you're using a multiprocessor OS). VMware is similar in that it doesn't support SMP for guest VMs running on Workstation and GSX Server. Multi-processor support is limited to the host machine's OS. Where VMware differs from Microsoft is that ESX Server supports SMP for guest VMs with SMP licensing. For instance, if your host server has sixteen processors, you can configure four guest VMs, each having four processors.

Purchasing multiprocessor motherboards can be expensive: add the cost of each processor, and you could be in the range of an unrealistic budget. When contemplating using multiple processors, you can keep the lid on runaway costs by avoiding purchasing the fastest processors. There's little difference in performance between 2.8 gigahertz (GHz) and 3.2GHz. But there's a huge difference between 1.8GHz and 2.8GHz. By selecting processors that are two or three steps down from the fastest available, you can keep your system selection within reason. Also, you'll need to keep in mind that multiple processors don't scale linearly. That is, two 3GHz processors aren't twice as fast as a single 3GHz processor. You'll get diminishing returns for each added CPU and each increment in speed. Rein in the desire to go for the fastest possible CPU because there's more to a fast host than clock speed. Additional information you'll want to look for in the motherboard's documentation is compatibility information, such as what processors are supported with the chipset or the requirement of having to implement identical CPUs using the same stepping level.

Caution Running multiprocessor servers with CPUs that have different stepping levels can and often does create squirrely computers.

When deciding which CPU to purchase, you'll also need to consider cache memory. Without cache memory, CPU data requests would all have to be sent over the system bus to the memory modules. To avoid this bottleneck and achieve higher performance levels, CPU manufacturers incorporate cache into their processors. Additionally, motherboard manufacturers will incorporate cache into the motherboard. You'll see cache memory listed as Level 1, Level 2, and Level 3. Cache is fast and generally runs at the same frequency of the processor or system bus. Manufacturers are able to produce inexpensive chips by cutting cache levels. Conversely, manufacturers produce high-end chips by increasing cache levels. In general, the more cache made available to the processor, the faster everything will run. You can find Intel processors with cache levels as high as 4MB and as little as 128KB. If you stick with moderate cache levels for production servers, 1–2MB, your guest VMs will be rewarded with significantly better performance. If you want in-depth explanations of processor cache types, you'll want to visit your CPU manufacturer's Web site.

Controller Chipset

The chipset dictates the overall character of the motherboard; it's fundamentally responsible for determining which CPU, memory, and peripheral types are supported. For instance, Intel, Via, and Ali currently produce the majority of chipsets used, and each vendor typically is geared to

support specific processor types, such as Intel or AMD. Communication between the motherboard's components is specifically controlled by the north and south bridge chips. The north bridge chip is generally located near the CPU socket and interconnects the processor, memory, video bus, and south bridge chip. The south bridge chip facilitates communication between the north bridge, peripheral cards, and integrated devices. With so much riding on one chip, the north bridge, it should come as no surprise that the speed at which the FSB functions significantly impacts the host's performance. When it comes to chipsets, you'll want to specifically make sure your CPU selection is supported (including the type and speed) by the motherboard's FSB. You'll also want to look at the motherboard's specifications to see which memory type is supported. Fast processors and RAM require a fast FSB, and a mismatch can create a bottleneck. If you're comparing off-the-shelf systems, you'll want to select a motherboard capable of supporting a processor requiring FSB speeds of 400MHz or greater for production VM hosts. If you're interested in knowing which Intel processors support which bus speeds, the site at <http://processorfinder.intel.com> provides a list of CPUs and required bus speeds.

Memory Requirements

Despite your desires, the memory you choose for your system will be limited to what your motherboard is capable of handling. If you have an idea of the type of memory you want to use and the quantity, you must first make sure the motherboard supports the total memory requirement for your host and guest VMs. Second, you must determine the number of available memory slots, the type of memory supported, and the memory module size supported. Manufacturers are specific about which type of memory can be used in any given motherboard. In large part, the type of memory available for you to choose will be tied to what the CPU and chipset can support. Your motherboard will probably support several memory technology types—double data rate, second generation (DDR2); synchronous dynamic random access memory (SDRAM); and DDR SDRAM. If your system mentions any of the older types of memory—such as single data rate (SDR) SDRAM, fast page mode (FPM), or extended data out (EDO)—choose another motherboard.

When looking at the performance characteristics of RAM, expect to pay more for motherboards using faster types of RAM. When deciding which type of RAM to use, speed isn't as important as quantity in regard to networked VMs. The bottleneck will be the network and not the read/write rate of your RAM. Table 2-4 lists performance data for several memory types.

Table 2-4. *Memory Speeds*

Technology	Speed	Bandwidth
DDR2	PC2-6400	6.4GB/sec
DDR2	PC2-5300	5.3GB/sec
DDR2	PC2-4200	4.2GB/sec
DDR2	PC2-3200	3.2GB/sec
DDR	PC4000	4.0GB/sec
DDR	PC3200	3.2GB/sec
DDR	PC2700	2.7GB/sec
DDR	PC2100	2.1GB/sec
DDR	PC1600	1GB/sec

If you plan on running VMs in RAM only, search for motherboards supporting faster RAM, such as DDR2. Faster RAM will give your VM guests a little extra oomph. Being that your system can be only as fast as the slowest link, you'll probably want to shoot up the middle when it comes to speed and quantity for your host motherboard. That is, there's no point in paying for a motherboard supporting bleeding-edge memory speeds if you're trying to build a server with a single SCSI adapter and hard drive for the host and guest VMs to share. You're better off buying more SCSI controllers, hard drives, and RAM.

When culling the field of motherboard candidates, stick with the boards that come with dual-channel technology. Dual-channel systems allow the bandwidth of two memory modules to be used simultaneously. In cases where dual-channel technology is implemented, you're better off adding memory in pairs to get that added performance boost. For example, if you need 1GB of RAM, get two 512MB modules rather than one large one—you want to make that memory highway as wide as possible.

Caution Don't get burned by using cheap or generic memory. In the long run, it just isn't worth the headache that junk memory causes. For a few extra bucks up front, you can purchase quality memory products from major memory manufacturers such as Crucial, Viking, and PNY. Quality memory will more than pay for itself over time because it will keep you from troubleshooting the bizarre problems cheap memory causes. Also, you'll find that the many discount vendors peddling heavily discounted memory will offer a "lifetime warranty" with no intent of honoring it.

As a final consideration, you may want to install error correcting code (ECC) RAM in hosts supporting production environments with mission-critical applications. ECC RAM utilizes an extra chip that detects if data is correctly read from or written to the memory module. In many cases, and depending on the type of error, ECC RAM can correct the error. Having the ability to detect and correct errors means a server is less likely to crash, but you'll take a small performance hit by implementing ECC memory.

Bus Types

The availability and quantity of Industry Standard Architecture (ISA), PCI, PCI Extended (PCI-X), and Accelerated Graphics Port (AGP) bus types are extremely important when it comes to hosting VMs. Small motherboards don't offer many expansion slots, and few come with ISA capability. ISA expansion slots are an issue if you need to reuse expensive ISA devices, such as multiport modems. Multiple AGP slots are important if your system requires high-performance video and multiple monitors. Multiple monitors are extremely useful for managing a VM host and guest in full-screen view without the aid of a KVM. A motherboard having only two or three PCI expansion slots limits your ability to correctly equip your servers with sufficient hardware for hosting. For instance, if you need two RAID controllers and three network adapters, you'll need five PCI slots on your motherboard. If five slots aren't available, you'll have to depend on integrated devices or expect less in the performance department. PCI-X is an enhancement over the traditional PCI bus in that the speed of the PCI bus is increased from 133MB/sec to a whopping 1GB/sec. In addition, PCI-X is backward compatible with traditional PCI adapter cards running at the lower speeds. PCI-X was designed to provide the higher performance levels Gigabit Ethernet and Fibre Channel technology demanded. For VM hosts, including PCI-X

technology substantially increases performance and is something you'll need to consider in environments with high network utilization or where SAN interactivity is required.

You'll need to plan for the maximum number of expansion cards your server requires in order to correctly size a motherboard. If you find a motherboard that falls short on available and necessary slots, you'll need to turn to using expansion cards that offer increased capability. For instance, you can use multiport NICs and multihead video adapters in situations where slots are limited. Additionally, you can use integrated AGP video in situations where more PCI expansion slots are a priority. In the end, you need to know how many NICs, SCSI, RAID, and video adapters your host will require.

Integrated Devices

Integrated peripherals can seriously cut the cost of a system. You can safely run a computer without a single expansion card, but you won't generally get the same performance of a system using some or all expansion cards. Typically speaking, integrated video, networking, audio, USB, and hard drive controllers work well for most systems. As important as it is to have integrated devices, it's equally as important to be able to disable them in case of failure or poor performance. You'll find systems that share installed RAM. You'll want to avoid this if possible because the integrated devices will be taking away from what's available for the host and guest VMs. A few megabytes chewed up for video here and a few megabytes for a caching disk controller there can equal one less hosted VM. RAM is cheap, and good-quality manufacturers don't hesitate to provide the appropriate resources for integrated devices. Where integrated devices can really pay off for virtualization is with integrated RAID. Integrated RAID normally doesn't have huge amounts of cache, which therefore makes it perfect for running the host operating system. You can purchase a second RAID controller for guest VMs and scale the add-on cache to meet your data processing needs.

You'll want to specifically look for a minimum of integrated components to support your guest VMs. In particular, a good-quality motherboard will have two Enhanced Integrated Digital Electronics (EIDE) controllers with one being Serial ATA (SATA) capability. SATA controllers and hard drives can give you near-SCSI performance. With VM applications in a workstation configuration, the primary controller and hard drive can be used for the host operation system, and the secondary controller and hard drive can be used for guest VMs and a CD-ROM. On servers, you'll want at least one SCSI interface and one EIDE interface. In a server situation, you can hang a CD-ROM off the EIDE controller, use the SCSI controller for memory swap files, and use SCSI adapter cards for your host's and guests' main execution and storage locations. Integrated Ethernet adapters may not be so important in a tower workstation, but in pizza box-type servers where space is at a premium, integrated NICs are essential. If you're looking at a motherboard with integrated NICs, make sure it has two. You can team the interfaces; if one fails, the host continues to support the networking needs of your hosts. Integrated USB isn't necessarily important, but it's nice to have. If you opt for integrated video or can't find a motherboard without it, make sure the motherboard comes equipped with an AGP. Some manufacturers integrate an AGP video card but don't provide the AGP expansion slot. You'll be forced to use the slower PCI video technology, which may cause a bottleneck on the system bus. Integrated video support on a motherboard isn't uncommon and is generally a good way to save a few dollars. Integrated video isn't usually a problem with servers in that most servers don't require much more than 4–8MB of RAM. If you plan on playing graphics-intensive games on your host machine and using guest VMs for business, integrated video will prove to be useless.

Many new motherboards are capable of booting from USB, and this is a great alternative to booting a computer from a floppy disk or CD-ROM. USB also allows you to use external storage. Backing up your VMs to an inexpensive USB drive is a great way to add a layer of disaster recovery capability. Also, USB-attached storage is an excellent way to move VMs from a test environment to a production environment: it's oftentimes faster than pushing 20–30GB across the network. Lastly, integrated audio is something you may want to consider for basic acoustics. Integrated audio won't give you the advanced features of an adapter card. So, if you're looking to use a VM as a stereo, you can forget about using integrated audio.

Other integrated features you may want to look for in a motherboard are system monitoring and jumperless configurations. System monitoring includes the ability to read BIOS post codes for troubleshooting and the ability to track case and processor temperatures for reliability and performance reasons. Being that VM hosts will be consuming much more of a processor's available cycles than normal, it's important to keep operating temperatures at an optimum level. You can have a server that was once reliable easily become unstable with the addition of three or four VM guests. Being able to track system temperatures is a great way to stay out of the red zone when loading up a server on the road to moderate hardware utilization (60–80 percent). Jumperless settings are more of a convenience than anything else. They prevent you from having to open the system case to make a change on the motherboard. Also, jumperless settings mean that those of us with fat fingers no longer have to worry about dropping little parts in the case. For people who want to tweak performance with overclocking (not recommended in a production environment), jumperless motherboard configurations are a great way to quickly reconfigure the system.

Board Form Factor

If you have a computer case to host your hardware, you want to make sure you get a motherboard to fit it. Nothing is worse than spending a week researching the optimum motherboard, purchasing it, and finding it doesn't fit. Motherboards all require a specific power supply and are designed to fit a particular case. The power connectors differ from older-style AT motherboards to the newer ATX type. Moreover, newer motherboards use *soft-off* functionality, where power switching takes place on the motherboard instead of the power supply. As motherboards begin to support more integrated devices and power-attached peripherals, and as cases take on show-car characteristics, the power supply needs to have a rating sufficient to power the motherboard and everything beyond—hard drives, CD/DVD-ROM drives, fans, cameras, lights, and so on. If you require redundancy or are deploying your VMs in a production environment, you'll want to make sure your motherboard and case support redundant power supplies with high watt ratings (350–500).

Overall Quality

Price isn't necessarily a good indicator of quality, but good-quality products generally cost more. There's a lot of truth in the saying, "You get what you pay for." Nobody goes out of their way to purchase shoddy products, but if you don't research your motherboard purchase, you may think the product is substandard because it doesn't follow through on your expectations. Poor-quality motherboards can cause VMs to perform strangely and spontaneously reboot. You don't want to spend countless hours troubleshooting software problems when it's really a quality issue with hardware. Ferreting out quality requires looking at technical documentation from manufacturer Web sites and seeking advice from trusted professionals, Web forums, and

articles from reputable periodicals. You'll find that trustworthy motherboard manufacturers quickly stand out in your search.

Features that indicate good-quality motherboards are good component layout, good physical construction, a replaceable complementary metal oxide semiconductor (CMOS) battery, and extensive documentation. Nothing should get in the way of adding more RAM or processors to a system. Conversely, motherboard electronic components, such as capacitors, shouldn't get in the way of housed peripherals. Good motherboards have a general "beefy" quality about them. The board will be thick, capacitors will be big, and plenty of real estate will exist between everything to allow for adequate cooling. Lastly, the CMOS battery should be something you can replace from the local electronics store. When the battery goes bad, you'll want an immediate replacement part. A reboot on a bad CMOS battery could mean having to set the configuration for every integrated device manually. After figuring out RAID settings, interrupt requests (IRQs), and boot device priority because of a bad battery, you'll want immediate restitution. Fortunately, guest VMs aren't really affected by CMOS batteries other than that the guest VMs may set their system clocks to that on the physical host. Assuming you don't care about time stamps in log entries, you may be all right. If you host secure Web sites, you may find that your SSL certificates become temporarily invalidated because of an erroneously reported date. In the end, if you reboot your servers with some regularity, you'll want to immediately replace a bad CMOS battery.

Whether you choose to purchase a system or build your own, spend some time researching available options, and make sure the motherboard and vendor can deliver your expectations. Download the motherboard's manual, and make sure it's well-documented; for instance, make sure jumper configurations are clearly marked, processor and RAM capabilities are listed, and warranty information is noted. Being that the motherboard is the most important component in a system, purchasing the best possible motherboard will give you an excellent foundation for VM hosting and further your success with virtualization.

We understand that we're giving you a lot to consider, and at this point you may be completely overwhelmed with all this information. However, all major computer vendors publish data sheets for their systems. Using these data sheets gives you an easy way to compare the hardware features of multiple systems, which in turn will enable you to make a more informed decision. Additionally, reliable vendors will supply presales support to address all your concerns.

Considering Your Network

Building and deploying VM networks isn't really any different from deploying typical physical networks. In fact, the same issues that affect physical networks apply to virtual networks. Knowing how the devices from Chapter 1 function will help you determine what should be included in your VM networking plan. Before utilizing VMs, you'll need to decide on some networking priorities. Specifically, you'll need to decide if your VM implementation requires privacy, performance, fault tolerance, or security. You'll look at these networking requirements in the context of host-only, NAT, and bridged networking.

Public or Private VMs

The first thing you'll have to decide is if your VM network will be public or private. If you're building a private network of VMs, you'll want to use host-only networking. In a private network, you're creating a completely isolated environment in which you're required to supply all

traditional networking services, such as DHCP, Domain Name Server (DNS), Windows Internet Naming Service (WINS), Network Time Protocol (NTP), NetBIOS Name Server (NBNS), and so on; in other words, you'll be building a complete network from the ground up. The advantage to building a private VM network is that it will have zero impact on existing networks, and existing networks will have no impact on it. Host-only networking is a great way to learn about VMs and new operating systems. Host-only networks are also good for engaging in the practice of network forensics: you can analyze the behavior of network applications, ferret out nuances of communication protocols, and safely unleash the ravages of viruses, malware, and scumware for thorough analysis. In private networks, nothing leaves the sandbox.

Note In the future, if you ever need to connect the private network to production, you can maintain the integrity of your private network by configuring the VM host system to act as a router, for instance, by using Windows RRAS or ICS. This is also useful if you're limited on IP addresses. We'll cover the reconfiguration of host-only networks and NAT in Chapter 4.

If you're building VMs and need to access the outside world, it's still possible to keep a level of privacy and security for your VMs. NAT provides the basic firewall protection you'd expect to get from address translation because network communication can't be initiated from the host (public) network to the private (VM) NATed network.

After building and completing the configuration of your private network, you can use NAT to connect to your host's network. Whatever the host is connected to and has access to, your VMs will be able to access. Oftentimes, the best way to connect VMs to the Internet or local LAN resources is to use NAT. If you're testing new services and want hosts outside your virtual network to have access, you can use port forwarding to direct traffic to the VM hosting the service. Because all VMs share the host's IP address, port forwarding is your only option to open the private network for virtualization applications that support it, such as VMware products. Be advised that using NAT will adversely affect overall networking performance; however, it's a small price to pay for the security benefits. We'll discuss the configuration of port forwarding and NAT in more detail in Chapter 7.

While contemplating if and how you're going to integrate your VMs into your existing infrastructure, you may need to consider the performance impact of each VMware networking model and contrast it with the ease of configuration. The host-only network model provides the best possible performance for guest VMs. This is because network traffic doesn't pass across any communications medium, including the physical network adapters. Virtual hardware is employed to facilitate communications, and, despite the speed rating of the virtual network adapter, VMs in host-only networks pass traffic as fast as they can read and write to the host's memory. Host-networks can be simple to configure because virtualization applications will provide DHCP: you won't need to configure a separate server for this.

NAT networks incur latency during the translation process. This is because NAT is a fairly sophisticated protocol and tracks information entering and leaving the host. The NAT process puts enough overhead on the host system to negatively impact guest VM performance. Additionally, building NAT networks requires a bit more time during the configuration process and can cause some problems with regard to inbound traffic. Generally, inbound traffic is prohibited. In situations where port forwarding can be configured, members of the physical LAN

may still have problems gaining access to resources in the NAT network because not all protocols are supported. NAT networks are best used for private LANs that need access to the outside world while retaining their private state. Members of NAT networks have the speed benefit of host-only networking when communicating with peer members, but NAT networks take a performance hit when outside communication is required.

Bridged networking directly connects your guest VMs to the host network adapter(s). Many organizations use this to place their VMs right on the production LAN. Bridging guest VMs has several benefits:

- First, bridging a guest VM to the host is often the fastest way to connect to an existing network.
- Second, the guest will experience better network performance in that no additional routing protocols, such as NAT, are required.

Like physical computers, bridged VMs require unique settings, such as a host name, an IP address, and a MAC address. With bridged networking, you'll be limited to the speed of the physical adapter. For instance, if you have a guest with a virtual gigabit NIC and the physical NIC is rated at 100Mb, transmission will take place at 100Mb. Also, if you have a physical NIC rated at 1Gb and a guest's virtual NIC is rated at 100Mb, you can expect to take a performance hit. It isn't that your physical NIC is going to slow down; it's because of the driver technology utilized by the virtualization application.

Availability and Performance

Good networks can be characterized as fast, predictable, and highly available. Good networks limit user wait times and streamline infrastructure operations. Good networks are created over time by embracing simplicity and using redundancy techniques to achieve maximum uptime. The first milestone to cross on the way to the high-availability finish line requires load balancing and teaming; without these, you'll never be able to achieve high-availability performance goals.

The theory behind network performance tuning and high availability is to build fault tolerance into all systems to eliminate any single point of failure for hardware and software while maintaining fast service. Because building good network performance is inclusive of load balancing and teaming, high-performing networks readily lend themselves to high availability and disaster recovery requirements in that failover and standby systems are already present in the network configuration.

With this lofty network performance/availability definition set, what part do VMs play? VMs play into network availability and performance by supporting what you'd expect of a highly available and good-performing server:

- The virtualization layer neutralizes hardware dependence. If the VM application can be installed on new hardware, your VM server will run.
- Clustering on single-server hardware eliminates single-server failure. If one server panics or blue-screens, the other keeps working.
- The single-file design of VMs supports the portability needs of disaster recovery. Copy and save a server like any other file, and then run it anywhere.

- VMs take advantage of teamed NICs to increase throughput and prevent a common single point of failure. Caveats exist, though! We'll discuss this in further detail in a moment.

We already established server portability and virtualization concepts in Chapter 1. In the following sections, while touching on some commonsense networking tips, we'll cover the specifics of network adapter teaming; we'll save server clustering for Chapter 9. Despite the performance promises of adapter teaming and server clustering, keep in mind that simple is almost always better. As networking professionals, we know that needlessly complicated systems are just needlessly complicated systems. If a problem can be solved, corporate objectives can be upheld, and budgets can be satisfied with two tin cans and a piece of string, then you've built a good network. No excuse exists for not building reasonably fast, predictable, and accessible networks—this is the purpose of VMs, and you can add VMs to your bag of tricks for creating and supporting highly available networks employing internetworking best practices.

Simplicity

If you're looking to build a quick network of VMs capable of interacting with your existing infrastructure, then keep IT simple (KITS). Configure your VMs using the defaults of VM applications that are utilizing bridged networking. Microsoft and VMware have considered many of the common options and scenarios network professionals need to have a VM quickly viable; therefore, you can capitalize on the manufacturers' research and development efforts and reap the rewards. You can make virtual disk adjustments as necessary.

When you've outgrown boxed-product tuning and need to increase the load and redline on your servers, you'll have to spend a bit more time testing NIC teaming and server clustering in the lab to achieve the higher levels of reliability and performance you may be accustomed to getting. For instance, if you're looking to ease the bottleneck on a loaded-up VM server or achieve the possibility of increased uptime, you'll want to pursue adapter teaming and server clustering.

You have much to gain from the initial implementation of simple VM workgroup-type networks, but if you want enterprise-class disaster preparedness, you'll have to employ mesh networks using redundant switching and redundant routing. If you want enterprise-class performance, you'll have to employ network adapter teaming and VM server clustering.

Mesh Networks

Implementing redundant switching and routing really isn't within the scope of this book, but we wanted to mention it because of load balancing and teaming. You can achieve redundant switching and routing by using multiple physical NICs, switches, routers, and multiple inter-network connections utilizing teaming and load balancing. Load balancing and teaming are technologies used to keep mesh networks running in the event of a system failure. For instance, if you're building a highly available Web server, you'll need multiple physical servers with multiple physical connections to multiple switches connecting to multiple physical routers that connect to at least two different carriers using two different connectivity fabrics; you shouldn't use a major carrier and a minor carrier that leases lines from the major carrier. If you lose the major carrier, you also lose the minor carrier, and you can kiss your redundant efforts and mesh network goodbye.

The idea with redundant switching and routing is to ensure that your subnet's gateway is multihomed across truly different links. In the event of a NIC, switch, or router failure, multihoming maintains connectivity to your subnet and everything beyond your subnet using alternate hardware and routes to get the job done.

Teaming and Load Balancing

To gain some of the redundancy benefits and all the performance advantages of teaming and load balancing, you don't have to liquidate the corporate budget to create mesh networks. You can achieve the act of teaming in networked environments by aggregating servers or NICs together. In both cases, the end result is more reliable network communications by removing the physical server or NIC as a single point of failure and gaining the advantage of being able to potentially process more client requests (traffic) by having multiple devices performing similar service tasks. For those not in the know, joining multiple servers into a teamed group is called *clustering*, and joining multiple NICs together is termed *network adapter teaming*.

Load balancing is often confused with teaming; however, *load balancing* is the act of processing client requests in a shared manner, such as in a round-robin fashion. One network adapter services the first client request, and the next adapter services the next; or, one server processes the first request, and the next server processes the second client request. Standby NICs and servers are different from load-balanced ones because in the event of a NIC or server failure, you'll have a brief service outage until the server or NIC comes online. Conversely, service failures on networks using clustering and adapter teaming are transparent to the end user.

A network load balancing cluster combines servers to achieve two ends: scalability and availability. Increased scalability is achieved by distributing client requests across the server cluster, and availability is increased in the event of a server failure in that service requests are redirected to the remaining servers. Server clustering technology is available in both Microsoft and Linux operating systems. We'll discuss clustering at length in Chapter 9 and focus on network adapter scalability and availability here.

Network Adapter Teaming

Network adapter teaming provides servers with a level network hardware fault tolerance and increased network throughput. Teamed adapters are logically aggregated by software drivers. The teaming driver manipulates each adapter's MAC address and logically assigns a new MAC address so the team acts like a single NIC. A teamed NIC is transparent to guest VMs. If one physical network adapter fails, automatic failover seamlessly processes network traffic on the remaining NICs in the team. The software drivers involved with teaming manipulate the MAC address of the physical NICs, and this is generally why OS manufacturers avoid supporting related issues with teaming and even suggest avoiding them in server clusters.

Caution VMware doesn't support network adapter teaming for GSX Server hosts running on Linux, and VMware has yet to officially test it as of this book's publication. Additionally, Microsoft doesn't support the teaming of network adapters and sees it as something beyond the scope of its available OS support services. Both VM application manufacturers lay the responsibility for network adapter teaming at the feet of your NIC vendor.

VMware does offer limited adapter teaming support for GSX Server running on Windows OSs. The Windows-hosted GSX Server supports Broadcom-based network adapters and teaming software in three modes:

- Generic trunking (Fast EtherChannel [FEC]/Gigabit EtherChannel [GEC]/802.3ad-Draft Static)
- Link aggregation (802.3ad)
- Smart load balance and failover

Additionally, VMware supports the Windows-hosted GSX Server utilizing Intel-based network adapters running Intel PROSet version 6.4 (or greater) in five modes:

- Adapter fault tolerance
- Adaptive load balancing
- FEC/802.3ad static link aggregation
- GEC/802.3ad static link aggregation
- IEEE 802.3ad dynamic link aggregation

From the ESX Server install, *adapter bonding*, the functional equivalent of teaming, is available. ESX Server doesn't support teaming for the console interface.

Assuming you run into problems with teaming, and keeping in mind that teaming has traditionally been the domain of hardware manufacturers and not OS manufacturers, you'll want to start your research at the NIC vendor's Web site, and you'll want to consult any accompanying network adapter documentation. In addition, most vendor Web sites have good user communities in which knowledgeable people are eager to help share information and solve problems. Be sure to implement any suggestions in a test environment first.

Don't be discouraged by the lack of OS manufacturer support for adapter teaming; the redundancy and performance benefits are too great not to support team adapters. Teaming software from enterprise-class vendors comes with excellent support, seamlessly installs, and functions smoothly with supported switches. Some switches must be configured before teaming can be properly implemented. You'll need to research the specifics regarding the switch to which you'll connect teamed adapters. The trick to ensuring your adapter team is bound to your guest VMs is making sure the bridge protocol is bound to the teamed NICs and unbound from the physical network adapters. If you set up teaming prior to installing VM application software, you'll have solved nearly all your teaming problems before they ever happen.

If, after teaming NICs in a server, you don't experience a fair level of increased network performance, you may have a server that's suffering from high CPU utilization or just experiencing nonlinear scaling of teaming. For instance, if a server is consistently near 90 percent utilization on a 1GHz processor, availing the server to increased network capacity does nothing for packet processing. If you use the metric that 1Hz is required for every 1bps of data being processed, then in the previous scenario, potential throughput is limited to 1bps—that's a 100Mb NIC running at half duplex. Additionally, you'll find that teaming 100Mb NICs improves performance more than gigabit NICs. The reason for this is that 100Mb NICs are generally overtaxed initially, and installing 5Gb NICs in a team doesn't equal a fivefold increase in performance. With basic network adapter teams, you make available the potential to process more traffic and ensure network connectivity in the event of an adapter failure.

Note Virtual NICs support different throughput ratings. Currently, Microsoft's Virtual PC and VMware's Workstation can support emulated NICs rated 100Mb. Microsoft's Virtual Server and VMware's GSX Server and ESX Server support emulated gigabit NICs. Whether your VMs are server or workstation class, your VMs will probably never see their rated available bandwidth, but the potential is there.

VM Networking Configurations

From Chapter 1, you know that Microsoft and VMware can accomplish several types of networking with VMs. Determining the network mode you need to use with your VMs is based on whether interactivity with the outside world is required, such as physical LAN access or Internet connectivity. If the physical LAN needs access to your VMs, you'll want to stick with bridged networking. If you require an isolated network of VMs devoid of external connectivity and invisible to the physical LAN, then host-only networking will be your best choice. If you want a network of VMs that are invisible to the physical LAN and you require external Internet connectivity, you'll want to use NAT networking.

VMware controls the configuration of network types by using virtual network adapters and virtual switches. VMware's VM application software has ten available switches, VMnet0–9, and can use up to nine NICs. The NICs are in turn mapped to a switch. The network configuration of the switch will determine the network access type for the VM. VMware preconfigures three switches, one each for bridged (VMnet0), host-only (VMnet1), and NAT (VMnet8) networking. You can configure the remaining switches, VMnet 2–7 and 9, for host-only or bridged networking. You can create complex custom networks by using multiple network adapters connected to different switches. For instance, by using proxy-type software, a VM can be multihomed on a host-only network and NATed network to create a gateway for an entire virtual LAN.

Bridged networking makes VMs appear as if they're on the same physical network as the host. With bridged networking, you'll impact the production network as if you just set up and configured a new physical machine. For instance, your VM will need an IP address, DNS/NBNS/NetBIOS server configuration, virus protection, and domain configuration settings: if your VM is to be a server, you'll configure it like a typical server, and if it's a workstation, you'll configure it like a typical workstation. It's that easy. The major performance impact

you'll experience is a decrease in network performance from having to share the host's physical network adapter with the host.

Host-only networking is the act of building a private network between guest VMs and the host. These private networks aren't natively visible from the outside world. For instance, you can build a host-only network with multiple VMs, complete with domain controllers, mail servers, and clients, and the traffic will never leave the host computer. Host-only networks are an excellent way to test new software in a secure sandboxed environment. With host-only networking, you can safely experiment with different networking protocols, such as TCP/IP and Internetwork Packet Exchange (IPX)/Sequential Packet Exchange (SPX), and test solutions on isolated prototypes.

NAT can connect a VM to virtually any TCP/IP network resource that's available to the host machine. NAT is extremely useful in situations where a production network DHCP pool is nearly exhausted, Token Ring access is desired, Internet access is required, and security is an issue. NAT performs its magic by translating the addresses of VMs in private host-only networks to that of the host.

Microsoft controls the configuration of virtual networks through the use of virtual network adapters and a preconfigured emulated Ethernet switch driver. You can find the driver in the properties of the physical computer's network adapter settings; it's called the Virtual Machine Network Services driver. If you find yourself troubleshooting network connectivity for Virtual PC guest VMs, you may want to begin fixing the problem by removing the driver and then reinstalling it. The driver file is labeled `VMNetSrv.inf` and is located in the `VMNetSrv` folder of your installation directory.

Virtual PC VMs can support up to four network interfaces, and each VM can be configured for network types that Microsoft refers to as Not Connected, Virtual Networking, External Networking, or Machine-to-Host Networking. Of the four adapters available to you, the first adapter can be set to Not Connected, Local Only, Bridged, or Shared (NAT). The rest of the adapters can be configured as Not Connected, Local Only, and Bridged. Microsoft VM applications allow you to create complicated virtual networks by using the virtual network adapters that include firewalls, routers, and proxy servers.

When you configure a VM as Not Connected, it's a stand-alone computer and is isolated from all machines (including the host). Configuring a VM as Not Connected is a good way to test software or learn new operating systems.

Virtual networking is local-only networking in Microsoft's VM applications, and it's the same as host-only networking for VMware. Local-only networks allow guest VMs to communicate using virtual Ethernet adapters via the host's memory. Local networks don't generate traffic on the physical network; therefore, local networks can't take advantage of the host's network resources.

External networking, or the Adapter on the Physical Computer setting, bridges traffic to the physical adapter, and it's the same as bridged networking for VMware. External networking allows VMs to act as if they were a standard physical computer on your LAN. For example, the physical LAN is responsible for providing a DHCP address and Internet connectivity. External networking is inclusive of shared networking. Shared networking, known as NAT in VMware, provides VMs with a private DHCP address and network address translation services that connect the VM to the host's physical network.

With shared networking, VMs are invisible to the physically attached LAN. Unlike VMware, Microsoft doesn't support port mapping for inbound traffic. Computers that aren't on the virtual network won't be able to access virtual machine services or any of the VM's ports. Additionally, shared networking doesn't directly support host or guest VM intercommunication.

Virtual machine-to-host networking allows VMs to communicate with the host system via the Microsoft Loopback adapter. You'll have to manually configure the Microsoft Loopback adapter to take advantage of virtual machine-to-host networking. You'll configure Microsoft Loopback adapters in Chapter 3. VMware automatically configures a loopback adapter for VM-to-host communication. Assuming that the proper proxy or routing software is installed, as in ICS, you can use host-only networking and loopback networking to connect VMs to the Internet via the host's dial-up adapter. You can also connect to non-Ethernet-type networks, such as Token Ring.

Supporting Generic SCSI

To provide support for physical SCSI devices, such as tape drives, tape libraries, CD/DVD-ROM drives, and scanners, VMware employs a generic SCSI device for Linux and Windows OSs. Generic SCSI allows a VM to directly connect to the physical SCSI device. Assuming the guest OS can supply a driver for the attached physical SCSI device, then VMware VMs can run the SCSI device after installing the appropriate driver. Microsoft virtualization software doesn't currently support generic SCSI devices. The only other thing you need to worry about with generic SCSI is troubleshooting. We'll show how to install some generic SCSI devices in Chapter 3.

Windows Guests

Generic SCSI is intended to be device independent, but it may not work with all devices. Therefore, when using generic SCSI (as with anything new), test your configurations in the lab. To get generic SCSI to properly work with your VM host and guest VMs, as in Windows XP, you may need to download an updated driver from the VMware Web site. Though rare, if, after adding a generic SCSI device to a Windows VM, the guest doesn't display your desired device, you'll have to manually edit the VM's configuration file (*filename.vmx*). Listing 2-1 shows a typical configuration file.

Listing 2-1. Typical VM Configuration File

```
config.version = "7"
virtualHW.version = "3"
scsi0.present = "TRUE"
scsi0.virtualDev = "lsilogic"
memsize = "128"
scsi0:0.present = "TRUE"
scsi0:0.fileName = "Windows Server 2003 Standard Edition.vmdk"
ide1:0.present = "TRUE"
ide1:0.fileName = "auto detect"
ide1:0.deviceType = "cdrom-raw"
floppy0.fileName = "A:"
Ethernet0.present = "TRUE"
usb.present = "FALSE"
displayName = "W2K3"
guestOS = "winNetStandard"
priority.grabbed = "normal"
```

```

priority.ungrabbed = "normal"
powerType.powerOff = "default"
powerType.powerOn = "default"
powerType.suspend = "default"
powerType.reset = "default"
ide1:0.startConnected = "FALSE"
Ethernet0.addressType = "generated"
uuid.location = "56 4d f2 13 f1 53 87 be-a7 8b c2 f2 a2 fa 16 de"
uuid.bios = "56 4d f2 13 f1 53 87 be-a7 8b c2 f2 a2 fa 16 de"
ethernet0.generatedAddress = "00:0c:29:fa:16:de"
ethernet0.generatedAddressOffset = "0"
floppy0.startConnected = "FALSE"
Ethernet0.virtualDev = "vmxnet"
tools.syncTime = "FALSE"
undopoints.seqNum = "0"
scsi0:0.mode = "undoable"
scsi0:0.redo = ".\Windows Server 2003 Standard Edition.vmdk.REDO_a03108"
undopoint.restoreFromCheckpoint = "FALSE"
undopoint.checkpointedOnline = "TRUE"floppy0.startConnected = "FALSE"
tools.syncTime = "FALSE"

```

We hope you won't have to manually edit VMware configuration files to fix issues related to generic SCSI. In the event that you find yourself poking around in a text file similar to Listing 2-1, you can count on four reasons for the SCSI device not properly installing for Windows guests:

- The device isn't physically configured correctly (is it plugged in and turned on?).
- The driver isn't installed on the host.
- The host driver prevents the SCSI device from being detected by the guest.
- The guest requires a device driver that doesn't exist for the host system.

After eliminating any of these reasons for generic SCSI failure, you'll need to use a text editor, such as Notepad or Vi (Vi can be used in Windows and is even available on a Windows Resource Kit), and modify the VM's configuration file to solve the problem. To start, you'll need to locate the VM's configuration file: it uses a .vmx extension, and it should be edited only while the VM in question is powered down.

VMware suggests that only experts edit VM configuration files, but you can safely edit the file if you make a copy before making changes. If something goes awry during the reconfiguration process, you can simply write over your changes with the previously copied file and start again. When troubleshooting generic SCSI for Windows, you'll want to focus on one of three things:

- Is this the installation of a new SCSI adapter?
- Is this the installation of a new SCSI device?
- Is this a failure of the Add Hardware Wizard?

In the first instance, if you're troubleshooting the installation of a new SCSI adapter (meaning this isn't a preexisting SCSI adapter), make sure the downed VM's configuration file contains this:

```
scsiZ:Y.present = "true"  
scsiZ:Y.deviceType = "scsi-passthru"  
scsiZ:Y.fileName = "scsiX:Y"
```

In the second instance, if you're troubleshooting the installation of a new SCSI device on an existing VM-recognized SCSI adapter, make sure the downed VM's configuration file contains this:

```
scsiZ:Y.deviceType = "scsi-passthru"  
scsiZ:Y.fileName = "scsiX:Y"
```

In the third instance of troubleshooting, you may have to address failures related to the Windows Add Hardware Wizard. The wizard often doesn't properly recognize newly installed or preexisting SCSI adapters and devices. If this is the case, make sure the downed VM's configuration file contains this:

```
scsiZ:Y.fileName = "scsiX:Y"
```

In all three troubleshooting scenarios, X, Y, and Z will be defined according to the following:

- X is the device's SCSI bus on the host system.
- Y is the device's target ID in the virtual machine and on the host. It must be identical to function correctly.
- Z is the device's SCSI bus in the virtual machine.

When determining which numbers to use for X, Y, and Z, keep in mind that SCSI buses are assigned numbers after available IDE buses, and the device target ID is normally assigned via switches or jumpers on the device.

Linux Guests

Like Windows operating systems, VMware supports generic SCSI for Linux VMs. Generic SCSI is nothing more than pass-through access to physical SCSI devices for which the host loads drivers. Assuming a driver has been successfully loaded, guest VMs can use any SCSI device that the host can. The generic SCSI driver is responsible for mapping attached SCSI devices in the /dev directory. To take advantage of generic SCSI, it requires driver version 1.1.36 (sg.o) and must be used with kernel 2.2.14 or higher. Each device will have an entry in the directory beginning with sg (SCSI generic) and ending with a letter. The first generic SCSI device would be /dev/sga, the second would be /dev/sgb, and so on. The order of the entries is dictated by what's specified in the /proc/scsi/scsi file, beginning with the lowest ID and adapter and ending with the highest ID and adapter. You should never use /dev/st0 or /dev/scd0. You can view the contents of the /proc/scsi/scsi directory by entering `cat /proc/scsi/scsi` at the command-line interface (CLI).

Juggling host and guest connectivity to disk drives (`sd`), DVD/CD-ROM drives (`scd`), and tape drives (`st`) can be tricky. If you permit simultaneous connectivity for the host and guest systems, your Linux system may become unstable and data corruption/loss can occur. If you don't have two SCSI disk controllers, one for the guest and one for the host, this is a good time to rethink your configuration strategy.

Simultaneous access becomes an issue because Linux, while installing the generic SCSI driver, additionally recognizes SCSI devices, such as the previous devices, as `sg` entries in the `/dev` directory. This creates a dual listing for SCSI devices. The first entry was created during the install of the host's Linux OS. Though VMware does an excellent job of arbitrating access to a single resource, it isn't always successful in making sure the dual device listings aren't simultaneously accessed under the `/dev` and `/dev/sg` listings. Therefore, don't use the same device in a host and guest concurrently.

After adding a generic SCSI device under Linux, you'll need to check the permissions on the device. For the device to be of any use to guest VMs, read and write permissions must be assigned to the device. If standard users, other than the superuser, will need access to the device, groups should be created for access control. We'll show how to install and set permissions on generic SCSI devices in Chapter 3.

Considering Storage Options

Determining which hard disk type and configuration to use for hosts and guests is based on the purpose of the final VM system. Presentation, educational, test, and production environments all have different reliability and availability requirements and widely differing budget constraints. Correctly identifying the purpose of a VM host will help you budget available resources so you can prioritize host needs, as shown in Figure 2-1.

High Priority—RAID SCSI Drives



Production Networks
Workgroups
Test Benches
Educational Environments
Presentations

Low Priority—Inexpensive IDE Drives

Figure 2-1. *Prioritizing host storage*

In the following sections, we'll cover physical and virtual SCSI/IDE disks and their relative performance characteristics. We'll also briefly discuss the commonsense application of each.

Physical Hard Drive Specifications

Physical hard drives have several specifications with which you need to be familiar. Being that virtual hard drives are mapped to the physical drive, the performance of the virtual hard drive is similar or equivalent to the physical drive. That is, you can't have a better virtual hard drive than you do a physical one. The hard drive you choose impacts the host and guest OSs equally. With SCSI and IDE hard drives, you'll want to concern yourself with four specifications: spindle speed, cache buffer size, access time, and data transfer rate.

Spindle Speed

Spindle speed is the rotation speed of the hard drive's platters. Speeds generally step from 4,200 revolutions per minute (RPM), 5,400RPM, and 7,200RPM for IDE drives and from 10,000–15,000RPM for SCSI drives. The higher the speed, the better the performance you'll experience. Conversely, more speed means more money and heat; therefore, you'll need to increase your budget and ventilation. You generally will find low spindle speed (4,200–4,500RPM) small form factor hard drives in notebooks, and these drives are sufficient only for testing or demonstrating a handful of concurrently running VMs. Avoid using similar spindle speeds in production VM host workstation and server systems, or you'll take a serious performance hit.

Cache

The buffer cache on a hard drive is similar to that found on a motherboard. The memory is used to speed up data retrieval. Most hard drives come with a 2MB, 8MB, or 16MB cache. If you're building a workstation for testing, then you can save some money by going with smaller buffer sizes. For servers, video processing, and extensive data manipulation using spreadsheets or large documents, you'll want to go with larger buffers to get that extra performance boost. Moreover, if you're using RAID controllers, you'll want increased cache levels for file servers. Cache creates performance differences significant enough for VM host applications and guest VMs that investing in larger buffers is merited because it decreases overall system response time, which allows you to more seamlessly run multiple VMs.

Access Time

The rate at which a hard drive can locate a particular file is referred to as *access time*. This factor becomes extremely important for file server-type VMs. For instance, if a VM will be frequently manipulating thousands or tens of thousands of files, being able to find each one in a reasonable time becomes problematic on drives with high access times. High access times will cause VMs to appear to hang while the hard drive searches for files. By sticking with the lowest possible access times, you reduce latency. As you stack more VMs onto a single host, this access time becomes more critical.

Transfer Rate

Transfer rate generally refers to one of two types, internal and external. *Internal* transfer rate is the rate a hard disk physically reads data from the surface of the platter and sends it to the drive cache. *External* transfer rate is the speed data can be sent from the cache to the system's interface. Trying to compare the transfer rates among different manufacturers may be difficult; they all typically use different methods to calculate drive specifications. You will, however, find that the transfer rates of parallel ATA IDE drives are less than that of SATA IDE drives, and both are less than that of SCSI transfer rates. When it comes to service life, SCSI will win in the warranty and longevity department.

You'll get better sustained transfer rates from SCSI drives. If you want to keep the data flowing on a busy VM, such as an e-commerce Web server, and sustain higher levels of performance, transfer rates are critical. Transfer rates are less critical if a system operates in bursts, such as a print server. Because external transfer rates depend on the internal rates being able to keep the cache full, internal transfer rates are a better indicator of performance when selecting drives for your VM host machines. You can download data sheets from hard drive manufacturer Web sites to compare transfer rates for hard drives.

We mentioned earlier that having multiple disk controllers and disks provide for better VM hosting, so let's feed the supergeek in us and look more closely at the mathematic justification for using multiple controllers and disks in your VM host. You'll need to pull hard drive data sheets to compare throughput averages of disks to get a better feel for overall performance as compared to theoretical maximums. For this discussion, we'll make some sweeping generalities about average data throughput that many hard drives support. To that end, what do the numbers look like between a VM host utilizing two Ultra 320 SCSI drives across two channels compared to a VM host using two Serial ATA 150 drives across two channels? Serial ATA drives have a maximum throughput of 150MB/sec and can sustain an average transfer rate between 35MB/sec and 60MB/sec. Your total throughput for a two-channel system would provide host and guest VMs with a maximum of 300MB/sec, and in constant data delivery mode, VMs would see about 70–120MB/sec total. Looking at Ultra 320 SCSI drives using one channel, maximum throughput could reach 320MB/sec and would average about 50–80MB/sec. Using a two-channel Ultra 320 SCSI controller, theoretical maximum throughput would be 640MB/sec and would be 100–160MB/sec for constant data delivery. To get equivalent speeds from SATA devices, you'd need a five-channel adapter and five hard drives: this would put you in the price range of a SCSI solution. The upshot with SATA is that storage is inexpensive. In fact, SATA storage is so inexpensive that it's not uncommon for disk-to-disk backup systems to be composed entirely of SATA drives. If you combined the space of five 350GB hard drives, you'd have 1.7TB. To get the equivalent from 143GB SCSI drives, you'd have to purchase 13 hard drives! When you look at SCSI and SATA for large storage solutions, you can't question the cost benefit of using SATA. When it comes to performance, SCSI blows the doors off SATA technology. Determining if you should go with SCSI or SATA can be difficult, though. You can use Table 2-5 as a guide to help determine if your host implementation situation should use SATA or SCSI technology.

Note VMware's ESX Server requires SCSI-attached storage for running guest VMs. However, you can store guests on IDE devices.

Table 2-5. *Hard Disk Implementation Situations*

System	Usage	Storage Disk Type
Desktops	Office suites/games	All SATA
Workstations	Engineering/CAD	Mostly SATA/some SCSI
Entry-level servers	Web/e-mail/automation	All SATA
Mid-level servers	Web/e-mail/automation	Partly SATA/partly SCSI
High-end servers	Web/e-mail/automation/databases	All SCSI
Enterprise servers	Disk-to-disk backup	Mostly SATA/some SCSI

We also want to point out that the performance increase SCSI drives experience over IDE drives isn't all attributed to drive design. The increase in performance is generated by the structure of the SCSI bus. The SCSI controller bus can manage hard drives without having to interrupt the processor for support, and the controller can use all drives attached to the bus simultaneously. IDE drives are limited to sharing the IDE bus. For instance, if you connected two IDE drives to the same bus, the drives would have to take turns communicating over the IDE bus. IDE drives are ideal for single-user computers, low-end servers, and inexpensive storage. If you're building a VM host that will be pounded in a multitasking environment, the extra expense of SCSI drives is well worth the performance boost your VMs will experience.

RAID

To create a RAID group, you can use one of two approaches: RAID implemented in software or RAID implemented in hardware. As with anything, each RAID type has positive and negative points. When considering RAID for your VM solution, you're looking to increase capacity, gain a performance edge, or add fault tolerance.

Hardware RAID vs. Software RAID

If you want the best performance and security from a RAID solution, you'll want to implement a hardware solution. You obviously will not get more performance from a software implementation, as it makes the VM host do more work. The independent architecture of hardware RAID, meaning that the operating system functions independently of the management features of the RAID controller, will deliver better performance and security for VMs. Software RAID is implemented within the host's operating system. Many operating systems come with a software RAID capability as a standard feature, and using it will save you the cost of a RAID controller. Unfortunately, software RAID adds overhead to the host by loading up the CPU and consuming memory. The more time the CPU spends performing RAID tasks, the less time it has to service guest VMs. The extra overhead may be a moot point in low utilization environments where you gain the redundancy advantage RAID has to offer.

Note Let's take a quick reality check with software RAID implementations. If you've purchased server-class virtualization applications to use VMs in a production environment, money exists within your budget for a hardware RAID controller. If you don't have a hardware RAID controller for your host and insist on using software RAID, do everyone a favor and return the VM software. Buy a hardware RAID controller first.

You implement hardware RAID using a special controller card or motherboard chipset. This is more efficient than software RAID. The controller has its own processing environment to take care of RAID tasks. Hardware RAID is managed independently of the host, and all data related to the creation and management of the RAID array is stored in the controller. Software RAID stores RAID information within the drives it's protecting. In the event of operating system failure, which solution do you think is more likely to boot? In the event of disk failure, which solution is more likely to preserve the state of your data? If you choose to use a software implementation of RAID, it should be used only for redundancy purposes and when hardware RAID is cost prohibitive. For better all-around performance and security, hardware RAID is your best choice.

In the not-too-distant past, RAID controllers were available only for SCSI hard drives. With the increased throughput of SATA technology and the need for inexpensive RAID controllers, manufacturers are making hardware RAID available for SATA technology. In situations where hardware RAID is needed and the budget is looking thin, SATA RAID is an excellent choice. These new breeds of RAID controllers come equipped with several disk interfaces, some as high as 16 ports! Like their SCSI counterparts, SATA controllers can have hotswap capability, allowing for the replacement of failed disks on the fly. It isn't uncommon for new entry-level or mid-level servers to come equipped with an SATA RAID option.

RAID Types

With RAID implementation selection out of the way, let's look at the available RAID types and how they impact the performance and redundancy of your VM solution. RAID types commonly found in production environments are RAID 0, RAID 1, and RAID 5, and each offers specific benefits and drawbacks.

RAID 0 isn't really a RAID type because redundancy isn't available (no parity). Despite the lack of redundancy, RAID 0 offers the best possible performance increase for VMs. RAID data is striped across two or more disks. *Striping* is the process of splitting data into blocks and distributing it across the drives in the array. Distributing the I/O load across the disks improves overall VM performance. RAID 0 is great to use where a single hard disk is normally used and the loss of data isn't critical. Generally, you'll find RAID 0 implemented for a computer-aided drawing (CAD) or video-editing workstation.

RAID 1, or *disk mirroring*, maintains an identical copy of all data on different disks in the array. At a minimum, you must use two disks. If one disk goes bad, service availability of the host and guest VMs will be maintained. RAID 1 provides for faster disk reads because two data locations are working to service one request. Conversely, RAID 1 negatively impacts disk write performance because data must be written to two locations. RAID 1 is good for protecting servers that don't require lots of disk space or where extensive file writing will not be an issue, such as for a server hosting a read-only database. In terms of using RAID 1 for a host, it isn't a

bad choice for redundancy when drives are limited. You'll see a noticeable decrease in performance for guest VMs that use virtual disk files.

RAID 5, also known as *block-level striping with distributed parity*, stripes data and parity information across a minimum of three hard drives. RAID 5 has similar performance characteristics as RAID 0 save for the overhead of writing parity information to each drive in the array. The performance impact of having to write the parity information isn't as significant as having to perform disk mirroring. Fault tolerance is derived from data blocks and its parity information being stored on separate drives in the array. If one drive should fail, the array would continue to function because the data parity is stored on a different disk. When a drive fails in a RAID 5 array, it functions in a degraded state until the failed drive is replaced. RAID 5 is common in production environments and offers the best of all worlds—excellent capacity potential, good performance, and better fault tolerance for your host and VMs.

Host Disk Sizing

Sizing system storage is extremely important for both virtual and physical machines. Because you'll be collapsing the infrastructure onto one box, your storage needs in a single host computer will be a multiple of the servers you want to run on it. If you have four servers with 100GB storage capacity, for example, you'll need a host with a minimum of 400GB of space and additional room for the host operating system. If you're using dynamic disks, you'll find that VMs will quickly consume your attached hard drive space. When choosing permanent storage for your hosts and guests, you need to decide if the final solution is for portable demonstrations and academic situations, for a test bench, or for the production environment.

Presentation and educational environments are generally restricted to small budgets and lower-end hardware. VMs typically will be loaded onto inexpensive large-capacity IDE devices and notebooks with limited quantities of RAM. In these cases, rather than spending your budget on SCSI drives, investing in more RAM is the key to getting three or four reasonably responsive dynamic disk VMs running simultaneously in host mode—only networks. For instance, if you need to turn up four VMs, you'll need 256MB for the host and 128MB for each guest, totaling 768MB. Running Windows 2000 and 2003 on 128MB is painful. In practice, you'll find you slide by fairly nicely by running Windows VMs with 192MB of RAM. In the previous situation, that now requires you to have 1GB of RAM.

Being that a test bench should more closely approximate an existing network environment, and being that VMs are often built on the test bench and then copied into production, you'll want to use SCSI drives in this case. Moreover, SCSI drives are a prerequisite if you plan on testing ESX Server. To get the most out of the two to three servers you may have available for testing, you'll want to have two SCSI controllers in your server (one for the host and one for guests) and a moderate amount of RAM: both SCSI controllers and ample quantities of RAM (256–512MB per VM) are required to adequately support a test domain of six to ten VMs and any enterprise applications, such as Oracle, Lotus Notes, or Exchange.

Note You'll also want to have similar, if not identical, NICs in your existing production environment to avoid undue network complications. You'll also want to have enough physical NICs to test any teaming or load balancing implementations that may be in production.

Production environments require high availability, good reliability, and good response times. It's necessary to have multiple RAID controllers per host (or SAN access), high-end SCSI drives, plenty of NICs, and tons of RAM. The more closely you can approximate the contents of multiple servers in one box, the better the VM host you'll have. The key to successfully deploying VMs is to consistently utilize 60–80 percent of the total physical server resources. The extra 30–40 percent is a buffer to offset utilization spikes. The more a server approaches 100 percent utilization, the more it appears to be frozen to an end user. Fast RAID controllers and SCSI disks with large buffers can easily feed a hungry network. RAID drives are more important than ever during the event of system failure because it isn't one server going down—it's a host and all its VMs! Determining the amount of drives you need is as easy as figuring the total number of fixed-disk VMs plus room for the host. You may find yourself “specing out” a physical server with a terabyte of storage or investing in a SAN.

Guest Disk Sizing

IDE device controllers are limited to four devices total, two each for the primary and secondary channels. You can use any CD-ROM and hard drive combination, but you can't exceed the limit of four devices: this is true for physical and virtual machines. All the VMs in this chapter are limited to 128GB virtual IDE devices, whereas physical systems can use IDE drives larger than 350GB. If you have a guest VM that requires more than 384GB of storage, using virtual IDE hard drives isn't a viable option. However, you can run virtual SCSI drives from a physical IDE drive to meet your sizing demands. The virtual SCSI disk performs better than the virtual IDE disk because of the difference in driver technology. Using virtual SCSI disks reduces the impact of virtualization overhead and will more closely approximate the physical performance characteristics of the hard drive.

Virtual SCSI disks are available for VMs in Microsoft Virtual Server and all VMware virtualization applications. Microsoft allows you to configure up to four virtual SCSI adapters with a maximum of seven virtual SCSI disks for a total of twenty-eight virtual SCSI disks. Each virtual SCSI disk can be as large as 2TB. Virtual Server emulates an Adaptec AIC-7870 SCSI controller. For VMware, you can configure up to seven virtual SCSI disks across three virtual SCSI controllers at 256GB per disk. VMs can directly connect to physical SCSI disks or use virtual disks. In either case, you can't exceed a total of 21 virtual drives. When configuring your VM, you can choose between a BusLogic or LSI controller card. The LSI card provides significantly better performance for your VMs. The trick to utilizing the LSI card for your VMs is to have the driver available on a floppy disk while installing your OS. You can download the LSI driver from the VMware Web site. When given the choice, opt for SCSI virtual disks.

The performance difference between mapping a VM to a physical disk and a virtual disk file is an important component to consider in your system design. In the first scenario, the VM treats the disk as a typical server would: it's raw storage, and the VM interacts directly with the physical disks. The VM reads and writes to the disk like a typical server with its own file management system. The only overhead really experienced by the VM guest is that of the virtualization layer.

Virtual disks, in essence, emulate physical hard drives and are stored as files on the host's physical hard drive. The guest VM will mount the file and use it like a hard drive. The guest OS will use its own file management system to read and write to the file. All this reading and writing takes place within the file management system of the host creating the extra overhead. If you opt to use dynamically expanding virtual disk files, this creates additional overhead because the file must be resized before data can be written to it. Dynamically expanding virtual disks also tend to fragment the host's hard drive, which further negatively impacts performance for the host and guest VMs alike. The major advantage virtual disk files have over mapped drives is portability. Moving or backing up a VM is like copying a regular file. In Chapters 4 and 6, we'll further discuss virtual disk performance.

Storage Area Networks

A SAN generally consists of a shared pool of resources that can include physical hard disks, tape libraries, tape drives, or optical disks that are residing on a special high-speed subnetwork (different from the existing Ethernet) and utilizing a special storage communication protocol, which is generally either Fibre Channel Protocol (FCP) or Internet SCSI (iSCSI). Together, it's all known as the *storage fabric*. The storage fabric uses a SAN-specific communication protocol tuned for high-bandwidth data transfers and low latency.

Most SAN interconnectivity is accomplished by using a collection of Fibre Channel switches, bridges, routers, multiplexers, extenders, directors, gateways, and storage devices. SANs afford connected servers the ability to interact with all available resources located within the SAN as if the devices were directly attached. In a reductive sense, you can think of a SAN as a network composed of interconnected storage components. We'll discuss SAN technology, as well as its configuration options and relationships to VMs, in much further detail in Chapter 13.

Servers generally gain access to the SAN via fiber-optic cabling; however, connectivity is also available over copper. What SANs offer VMs is the same thing offered to physical servers: increased performance levels, better reliability, and higher availability. These benefits are achieved because the SAN is responsible for data storage and management, effectively increasing the host's capacity for application processing. Extra processing capability results in lower latency for locally running VMs and faster access times for the end user. Disaster recovery capabilities are enhanced in that the time to restore a VM is significantly decreased and the physical data storage can be in a remote location.

Microsoft VMs and SANs

Microsoft VM virtual hard disks can be stored on a SAN. VMs don't need anything in particular to use a SAN because the host views the SAN as a local volume—the physical computer treats the SAN as a typical storage device, assigning it a local drive letter. The host simply places all files associated with VMs on the SAN via the assigned drive letter. If you're wondering if you can use SAN as a physical disk, you can't. Virtual Server doesn't provide emulation of SAN host bus adapter (HBA) cards. Microsoft will treat a SAN as a locally attached device.

VMware VMs and SANs

Like Microsoft, VMware treats a SAN as locally attached storage for ESX Server. Unlike Microsoft, VMware allows for the mounting of a SAN LUN for ESX Server (using a SAN as a physical disk). A LUN is a slice of the total SAN representing a section of hard drives that's in turn labeled to identify the slice as a single SCSI hard disk. Once the LUN is created, the host or ESX Server guest can address the disks in that particular SAN slice. ESX Server supports up to a maximum of 128 LUNs and can emulate QLogic or Emulex HBA cards.

ESX provides support for multipathing to help achieve a more highly available network by maintaining connections between the SAN device and server by creating multiple links with multiple HBAs, Fibre Channel switches, storage controllers, and Fibre Channel cables. Multipathing doesn't require specific drivers for failover support.

When installing ESX Server, you'll need to detach the server from the SAN. Detaching prevents accidentally formatting any arrays and prevents the SAN from being listed as the primary boot device for ESX Server. Moreover, VMware suggests you allow one ESX Server access to the SAN while configuring and formatting SAN and VMFS volumes. In a SAN solution, you want to install ESX Server on the local physically attached storage and then run all VMs from the SAN via an HBA card. You'll get near physical server performance from your VMs by allowing them sole use of the HBA cards. This also gives you a major backup advantage. Multiple VMs can directly back up to a tape library in a SAN simultaneously. Additionally, don't place ESX Server's core dump file on the SAN, as it can make the system unstable.

LUN Management

On initialization or when a Fibre Channel driver is loaded, ESX Server scans the SAN for LUNs. You can manually rescan for added or deleted LUNs by issuing `vmkfstools -s` at the CLI. If you're using QLogic HBAs, you'll have to flush the adapter's cache for entries in `/proc/scsi/qla2200` or `/proc/scsi/qla2300`. At the CLI, for example, enter the following:

```
echo "scsi-qlascan" > /proc/scsi/qla2300/0
vmkload_mod -u qla2300
vmkload_mod /usr/lib/vmware/vmkm/qla2300_604.o vmhba
```

If, after creating your LUNs and them not appearing after a scan, you'll need to change the `DiskMaxLun` field from Advanced Settings in the Management Interface. The number should be equal to the total number of LUNs plus one. In Figure 2-2, the default value is set to 8. Notice that the name of the server isn't using the default installation name; your system should have FQDN listed.

By default, ESX will see 0–7; you achieve this by typing **8** in the Value field. Rescan to detect the new LUNs. You can verify your result at the CLI by running `ls /proc/vmware/scsi/<fiber channel adapter>`.



Figure 2-2. *DiskMaxLun and DiskMaskLUNs field settings*

Before assigning SCSI or SAN storage to a virtual machine, you'll need to know the controller and target ID used by the service console and the controller and target ID used by the VMkernel. Though physically dismantling your system is an option for information regarding the service console, you can determine controller and target IDs from the CLI. The boot log file, `/var/log/messages`, and the SCSI file of the proc pseudo file system, `/proc/scsi/scsi`, both contain valuable information. You can view the contents of each by entering the following at the CLI:

```
cat /var/log/messages | less
cat /proc/scsi/scsi | less
```

For data regarding the controllers assigned to the VMkernel, you'll want to look to the `/proc/vmware/scsi` directory. This information will be available only if the VMkernel starts. For every controller assigned to the VMkernel, a corresponding directory should exist in `/proc/vmware/scsi`. The subdirectories of each controller will have a listing of devices, target IDs, and LUNs. For instance, if you had the `vmhba0` subdirectory with a `2:0` file, you'd want to perform the following commands at the CLI to view the file contents:

```
cat /proc/vmware/scsi/hba0/2:0 | less
```

After collecting your data, you can use it to configure your VMs in the Management Interface.

LUN Masking

Masking is also available if you don't want the VMkernel scanning or even accessing particular LUNs. Masking is generally performed for security reasons to prevent operating systems from accessing LUNs. You can accomplish masking by setting the `DiskMaskLUNs` field on the system's Options tab under Advanced Settings to this:

```
<adapter>:<target>:<comma separated LUN range>;
```

For example, if you wanted to mask LUNs 3, 4, and 50–60 on `vmhba 1`, target 4, and LUNs 5–9, 15, and 12–15 on `vmhba 2`, target 6, you'll need to set the `DiskMaskLUNs` option to `vmhba1:4:3,4,50-60;vmhba2:6:5-9,12-15`, as in Figure 2-2. You're prohibited from masking LUN 0. When using QLogic HBAs, you'll need to select the correct driver version as well (IBM, HP, or EMC).

Bus Sharing

Unlike masking, ESX provides for bus sharing. The bus sharing feature is useful for high-availability environments, such as clustering, where you want two VMs to access the same virtual disk. By default, ESX Server prevents this from happening: if you need to change the settings, you can use the Management Interface to set bus sharing to one of three options:

- **None:** VMs don't share disks.
- **Virtual:** VMs on the same host can share disks.
- **Physical:** VMS on different hosts can share disks.

You'll want to choose Virtual as the sharing type for building a cluster in a box, and you'll need to choose Physical if you want to hedge against hardware failure. Using physical bus sharing requires the virtual disk to be mutually accessible by each VM.

Persistent Binding

Persistent binding is also available for ESX Server VMs and affords the target IDs to be retained during reboots. Persistent bindings, the assignment of target IDs to specific Fibre Channel devices, are necessary if you're using the SAN for VM direct access: persistent bindings treat the SAN as a physical disk by directly mapping the SAN LUN as locally attached storage. As with all ESX Server administration, you can configure persistent binding at the console or through the Management Interface. As a word of caution, persistent binding can create major problems in FC-AL SAN topologies!

Summary

In this chapter, we reviewed no-nonsense techniques for budgeting, building, and installing Windows or Linux VM host solutions. We pointed out some network and system bottlenecks along the way, and we showed how to avoid them in bridged or NATed networks by installing and configuring multiple network and storage adapters in a teamed configuration. In addition, you know that when you use teamed network adapters, you must configure the connecting switch for Etherchannel and the appropriate teaming protocol. If you don't properly configure your teamed NICs, you may wind up with flapping links or loops that adversely impact the network, negating all performance gains.

Additionally, we discussed the importance and impact of network and storage considerations for host and guest VMs, including locally attached storage and SANs.

Now that you're aware of the techniques for selecting and preparing hardware for host systems, you're ready to move onto the next few chapters where we'll focus on hosting and installing VMs. We'll cover the techniques to help students, technology administrators, and sales professionals get the most out of virtualization by installing guest operating systems on both server and workstation VM applications.



Installing VM Applications on Desktops

With virtualization theory and host system selection considerations from Chapters 1 and 2 out of the way, it's time to move onto installing some major workstation-class virtualization applications. In particular, this chapter will focus on installing Microsoft Virtual PC and VMware Workstation on desktop operating systems. This chapter will also cover installation and optimization methods that will assist students, help-desk personnel, sales professionals, and systems engineers in deploying and configuring virtual machines on desktop and laptop computer systems.

Deploying VMs with Microsoft Virtual PC

Microsoft Virtual PC is intended to be installed on Microsoft desktop OSs (specifically, Windows 2000 Professional, Windows XP Professional, and Windows XP Tablet PC Edition). Virtual PC officially supports many Microsoft guest operating systems from MS-DOS 6.22 to Windows 2003 Server. It even officially supports OS/2. Though you can run many other operating systems without official support, this may be a moot point if you're in an all-Microsoft shop. In the following sections, we'll show you how to install Virtual PC on a Windows XP host computer.

As a word of caution, installing Virtual PC requires administrative privileges. Except for this, Virtual PC is a snap to set up. After the installation, you aren't required to reboot your system before configuring and installing guest VMs, but it's a good idea to give the host system a fresh restart. Perhaps the only hitch in the installation process is that the host computer will temporarily lose network connectivity during the virtual switch driver installation.

After completing the install of Virtual PC, a whole host of configuration options are available to you. These configuration options control the character of Virtual PC and its interactions with the host system resources (for example, memory consumption, CPU utilization, and system security).

We'll cover each setting in Chapter 4 while showing how to install guest operating systems. As a heads up, the easiest way to install an operating system in Virtual PC is to use the setup program. The same techniques you use to load an operating system on a physical machine, such as booting from a floppy disk or CD-ROM, are available to you with Virtual PC. If you're just experimenting with Virtual PC, you can scrape by on some really thin hardware

configurations. However, you'll find that you'll get the best results from Virtual PC by using best-practice minimums that come near to or exceed the following suggestions:

- Two 100 AT Attachment (ATA) IDE interfaces (one for the host and one for guests)
- One 100/1000MB NIC interface
- One CPU at 1GHz
- 1GB of RAM
- Two 80GB ATA IDE hard disks
- 133MHz FSB

After getting your hands on a copy of Virtual PC software, you can start the installation by launching the scant 19MB executable. After some preinstallation churning, you'll be greeted with the Virtual PC 2004 InstallShield Wizard.

Note You'll need an OS license for each guest VM you plan on installing and using with Virtual PC.

The install process will present you with a Customer Information screen where you'll need to supply some standard preinstallation information, such as your username, organization, and product key information. For an added layer of security, you'll also need to decide for whom the completed install of Virtual PC will be available. If you're unsure, select Anyone Who Uses This Computer (All Users).

After entering the license key, Virtual PC is ready to install. If you don't like the default installation directory, change the path by selecting Change. Now is also a good time to second-guess any selections and make changes to the install by selecting the Back button. When you're satisfied with the configuration settings, select Install. The complete installation takes a minute or two at most. Upon successfully completing the install, you'll be presented with the InstallShield Wizard's Completed screen.

When the installation is complete, you can find Virtual PC in the Programs menu. Being that Microsoft left out the option to create a desktop icon, take a moment to create one. Better yet, place a shortcut in the Quick Launch bar.

Microsoft has released Virtual PC 2004 Service Pack 1 (SP1), and it's available for download at <http://www.microsoft.com/downloads>. Take a moment to download SP1 (which is about 25MB) to a convenient location. Before starting the setup, however, be sure that all guest operating systems are turned off and aren't in the saved or paused state. VM saved states aren't compatible from the base Virtual PC application to the upgraded Virtual PC SP1 product. Virtual PC SP1 also includes an updated version of Virtual Machine Additions, so be sure to update all your guest VMs to this newer version by first uninstalling the old version and then installing the new. You should be aware that SP1 doesn't update Virtual PC help files. To get a better overview of what SP1 covers, you can download the SP1's Readme files at <http://www.microsoft.com/downloads>.

Begin the SP1 installation by unzipping the downloaded file and selecting Setup. You'll be greeted with the SP1 installation wizard. If you receive an error message, it may be the result of trying to install SP1 on a trial version of Virtual PC. The install should take only two to three minutes. When complete, you should see the Installation Completed screen. Microsoft requires you to have a validly licensed copy of Virtual PC to install the service pack.

With SP1 in place, take a moment to verify Virtual PC's version. You can determine the version by opening the Virtual PC Console and selecting Help ► About. Your Virtual PC 2004 SP1 version should be 5.3.582.27.

Before moving onto VMware Workstation installation, treat the service pack installation the way you would any other upgrade. Take a moment to browse through the help files and menus to get a feel for what has changed in Virtual PC. You'll want to have a bit of control of the software before installing guest VMs and reading Chapter 4.

Installing VMware Workstation for Windows

Installing VMware Workstation isn't much different from installing Virtual PC or any other hosted virtualization application. As you learned in Chapter 1, VMware supports many more OSs for its virtualization products than Microsoft, and this will be a critical factor for you if official support is required for Linux guest VMs.

Despite that you can really load up Workstation with many guest servers and still get good performance, keep in mind that it's licensed per user. For instance, if five people are going to be accessing guest VM servers on your VM Workstation implementation, you need five workstation licenses (in addition to any guest OS licensing). Workstation is approximately \$200 per license, and GSX Server is about \$1,400 for a two-CPU server. On the other hand, GSX Server provides for unlimited user access to guest VM servers, uses gigabit NICs, and supports an unlimited number of processors. You could probably save a few dollars with Workstation in a small network of five or six users, but it's really designed for a single user.

Before beginning to install VMware Workstation, quickly compare your host's hardware to these suggested best-practice minimums for VMware Workstation:

- Two 100 ATA IDE interfaces (one for the host and one for guests)
- Two 100MB NIC interfaces (one for the host and one for guests)
- One 1GHz processor
- 1GB of RAM
- Two 80GB ATA IDE hard disks
- 133MHz FSB

If you come close to this configuration, you'll have guest VMs that perform well and are a joy to use. Installing VMs on less hardware significantly increases latency and will waste what lab time you may have.

Tip To make sure you get the best performance from your IDE devices, make sure the IDE controllers in Device Manager use direct memory access (DMA). You can verify the settings by right-clicking the primary or secondary controller and selecting Properties. Select the Advanced Settings tab, and make sure that the Transfer Mode setting is set to DMA If Available. DMA access is much faster than programmed input/output (PIO) access.

Without further ado, let's begin installing VMware Workstation on your Windows host running Windows NT 4.0, Windows 2000, Windows XP, or Windows Server 2003.

Like Virtual PC, you'll want to install Workstation using administrative privileges. Launch the Workstation executable file from your install media. The file is about 40MB and will have a filename like `VMware-workstation-4.5.2-8848.exe`. You'll be greeted with Workstation's Installation Wizard.

The Destination Folder screen requires you to select the installation place for Workstation. Generally, the default is fine. If you want to change the install location, select Change, and specify a new directory. The installation location doesn't dictate the storage location of guest virtual disks, however. For performance reasons, you may want to specify your secondary IDE controller's hard drive when you create these. (We'll cover how to set up guest VMs in Chapter 4.)

You'll be given the opportunity to make changes on the Ready to Install the Program screen. If you're satisfied with your previous selections, choose Install to continue the installation. During the software installation, you may be asked to disable the CD-ROM autorun feature. Autorun can have a negative performance impact on your guest VMs, and it will cause unpredictable behavior in guest VMs.

The Installation Wizard asks you if you want it to find and rename virtual disks from previous versions. If you're upgrading, you'll want to select Yes. If this is a fresh install, select No. In addition, you'll be given the opportunity to supply registration information for the software during the install. If you already have a validly licensed product serial number, you can enter it along with your name and company information, or you can obtain a demo key from VMware's Web site. You'll have to register using a valid e-mail address to receive the trial key. Once you receive it, you can cut and paste it into the Serial Number field. If you don't have a serial number, select Skip. You can enter this information later at VMware's Management Interface. If all goes well, you'll finally see the Installation Wizard's Completed screen.

The Installation Wizard will create a desktop shortcut for Workstation. You can test the installation by launching the application. You'll be presented with the Tip of the Day pop-up box. If you want, you can disable this. After closing the tip window, Workstation's Management Interface launches.

As with any software installation, you'll want to check VMware's download Web site, <http://www.vmware.com/download/>, for any service releases regarding your new Workstation installation. Take a moment to browse the menus and skim the help files. You'll want to be familiar with configuration options and be aware of the breadth of help available to you when you install guest VMs in Chapter 4.

INSTALLING VMWARE'S DISKMOUNT UTILITY

If you go to VMware's download Web site, you'll find a cool utility that allows you to mount VMware virtual disks as drive letters for read and write access. If you use disk-imaging products, such as Norton Ghost, DiskMount is similar to the Norton Explorer utility. DiskMount will give you access to all files on any powered-off virtual disk created with VMware Workstation 4, GSX Server 2.5.1/3, and ESX Server 2. The utility is designed for Windows 2000, Windows XP, and Windows Server 2003. Unfortunately, VMware doesn't provide support for DiskMount. Assuming storage space is available, be sure to back up the disk you want to mount first. Then, work with the backup copy. You can make any permanent changes after running tests on the backup first.

When mounting virtual disks, you're limited to the following choices:

- Mounted virtual disks must have a unique drive letter greater than D.
- Only the file-allocated table (FAT)—12/16/32—and NT file system (NTFS) partitions are mountable.
- Permissions on the read-only disks must be changed to read and write.
- Compressed drives must be decompressed.

DiskMount is free, and you don't need a VMware virtualization application installed, such as Workstation or GSX Server, in order to use DiskMount. After installing the utility, you'll need to go to the CLI to perform DiskMount commands. The syntax for DiskMount is as follows:

```
vmware-mount [options] [drive letter:] [\\path\to\virtualdisk]
```

Table 3-1 lists its command options.

Table 3-1. *DiskMount Command Options*

Options	Action Performed
/v:N	Mounts volume N of a virtual disk (defaults to 1)
/p	Displays the partitions/volumes on the virtual disk
/d	Deletes the drive mapping to a virtual disk drive volume
/f	Forcibly deletes the mapping to a virtual disk drive volume
/?	Displays VMware mount information

Using the DiskMount utility is fairly simple. The following are examples of how to parse the command syntax at the Windows CLI:

- **To view mount virtual disks:** `vmware-mount`
- **To mount a virtual disk:** `vmware-mount R: "C:\My Virtual Machines\myvirtualdisk.vmdk"`
- **To mount a second volume on a virtual disk:** `vmware-mount /v:2 R: "C:\My Virtual Machines\myvirtualdisk.vmdk"`
- **To dismount virtual disks:** `vmware-mount R: /d`

Caution If you fail to dismount virtual disks after using the DiskMount utility, guest virtual machines won't be able to gain access to their disks.

Installing VMware Workstation for Linux

To get the most out of VMware Workstation for Linux, you should have similar best-practice minimums available for your host to those listed for Virtual PC. Also, be sure to have X Windows installed. Without X Windows, you won't be able to easily install and manage guest VMs from the console. If your hardware merely meets but doesn't exceed the manufacturer's minimum requirements, your guest VMs will appear to freeze under average load. You'll also need to make sure your version of Linux has the real-time clock function compiled in the kernel and the port PC-style hardware option (`CONFIG_PARPORT_PC`) loaded as a kernel module.

Caution Whether you're installing from a CD-ROM or a file downloaded from VMware's Web site, you need to make sure your file paths are reflected correctly in installation command statements. If you're unfamiliar with the Linux CLI, now is a good time to pick up an Apress book on Linux, such as *Tuning and Customizing a Linux System* by Daniel L. Morrill (Apress, 2002).

To begin installing Workstation for Linux, go to the CLI and make sure you have root permissions. If you're in a testing lab, it will be easier if you're logged in as root. In production environments, you'll need to log in with the account for which the install is intended and then issue the `su -` command.

Note If you're upgrading from a previous version, you'll need to remove the prebuilt modules' RPM package by executing `rpm -e VMwareWorkstationKernelModules` before proceeding with the installation.

You can install VMware Workstation for Linux using RPMs or TAR files. Choose the method that supports your version of Linux. We'll cover both in the following sections, starting with the RPM method. As a small heads up, sometimes the configuration program, `vmware-config.pl`, will appear to hang; you can press Q to advance to the next configuration prompt.

Installing the RPM

To install the RPM, follow these steps:

1. If you're installing from a CD-ROM, you'll need to mount it first. At the CLI or in a terminal window, enter this:

```
mount /dev/cdrom /mnt/cdrom
```

2. Next, you'll need to browse to the installation directory on the CD:

```
cd /mnt/cdrom
```

3. Now, locate the Linux directory containing the install files:

```
find /mnt/cdrom -name VMware*.rpm
```

4. Change your present working directory to the location of the RPM file:

```
cd /mnt/cdrom/<directory_name>
```

5. Now, enter the following for the RPM install:

```
rpm -Uvh /mnt/cdrom/<directory_name>/VMware-<version and build number>.rpm
```

You should see some hash marks advancing across the screen indicating installation activity.

6. After delivering the package, run the configuration program. You can verify the installation of the RPM by querying the package management system:

```
rpm -qa | grep 'VMware'
```

7. The system will respond with the package and version number installed. It should match the RPM you specified with the `rpm -Uvh` command in step 5.

8. With installation out of the way, you'll now need to run the configuration program. Enter the following at the CLI:

```
vmware-config.pl
```

9. You'll be asked to read the license agreement. You'll need to use the space bar to advance through the document. Assuming you agree to its terms, type **yes** and press the Enter key.
10. The configuration program will question if a network is required for your VMs. The default is Yes. If you need networking, simply press the Enter key.
11. If you have multiple network adapters, the program will ask you which adapter should be bridged to VMnet0. If you're happy with the default selection, press Enter to continue.
12. Next, you'll be asked if you'd like to configure any additional bridged networks. No is the default. Being that you can configure more bridged networks later, accept the default by pressing the Enter key.

13. Next, you'll be asked if NAT networking is necessary. Select the default (Yes) for now by pressing the Enter key.
14. The program will then ask to probe for an unused private subnet. The default is Yes. If you're happy with the system scanning your network, press the Enter key. The scan can take a couple of minutes, and upon completion, it will reveal what appears to be available for use.
15. The system will query if you'd like to use host-only networking in your VMs. The default is No. If this is what you want, press the Enter key. However, if you don't select Yes now, bridged and host-only networking won't be available to your VMs.
16. You'll next be asked if you'd like to share the host's file system with the guest VMs. If you have Samba already configured, you can select No; otherwise, you'll need to select Yes to have the configuration program configure it for you. To complete the configuration of Samba, you may be prompted for the username and password of the account that will use VMware.
17. The system will take a moment to configure VMware and then start all related services. The system ends the install by announcing that you can run VMware Workstation by running `/user/bin/vmware`. Take a moment to test your configuration. Conversely, if you want to uninstall the program, issue `rpm -e VMwareWorkstation` at the CLI.

Installing the TAR

If your OS doesn't use RPMs, you'll need to use the TAR installation of VMware for Linux. Follow these steps:

1. Before installing the software, create a temporary directory for the installation TAR file:

```
mkdir /tmp/vmlinux
```

2. Next, copy the TAR file to the newly created temporary directory on your hard drive. Be sure that your file paths correctly reflect the directories on your host system:

```
cp VMware-<version and build number>.tar.gz /tmp/vmlinux
```

3. Next, change to directory where the copied files reside:

```
cd /tmp/vmlinux
```

4. Because the install file is an archive, you'll need to unpack it by issuing the following command:

```
tar xzf /tmp/vmlinux/VMware-<version and build number>.tar.gz
```

5. Once the decompression is finished, find the installation directory:

```
find /tmp/vmlinux -name vmware-distrib
```

6. Next, navigate to the installation directory:

```
cd /tmp/vmlinux/vmware-distrib
```

7. Finally, execute the installation program:

```
./vmware-install.pl
```

8. You'll be prompted several times during the install. Generally, the defaults are sufficient for most installs save a caveat or two. The first prompt asks you to confirm the install directory, `/usr/bin`.
9. Next, the install program will ask you the location of the init directories, `/etc/rc.d`.
10. You're then prompted for the init scripts directory, `/etc/rc.d/init.d`.
11. The install program will then create the installation directory for the library files, `/usr/lib/vmware`.
12. The system will want to create the `/usr/share/man` directory for the manual files.
13. Next, you'll be asked to supply the directory for the VMware documentation files, which is `/usr/share/doc/vmware`. Press the Enter key to continue with the default. You may be prompted to confirm the creation of additional parent directories. Assuming this is okay, press Enter to continue.
14. Unlike the RPM install, the TAR install asks if you'd like to run the configuration program, `/usr/bin/vmware-config.pl`. The default is Yes; press the Enter key to run the VMware configuration program.

Configuring the TAR version of VMware for Linux installation is identical to that for the RPM configuration. You'll be asked to read the license agreement, configure networking, set up VM bridging, set up NAT, and so on. The installation ends the install by announcing you can run VMware Workstation by running `/usr/bin/vmware`. Take a moment to test your install. If you have problems, you can uninstall the program and give it another shot.

Regardless of the installation methodology, RPM or TAR, you can run the configuration program at any time to reconfigure VMware Workstation. You'll also need to run the program any time you upgrade the Linux kernel or when you want to change the character of Workstation, such as removing or adding host-only networks. If you run the configuration program and it doesn't work, it's probably because of a listing issue in the default path statement. You'll need to check one of three locations depending on your needs. Table 3-2 lists the typical path directories.

Table 3-2. *Path Scripts*

User Account	Script to Modify
One user	<code>\$HOME/.bash_profile</code>
All users except root	<code>/etc/profile</code>
Root	<code>/root/.bash_profile</code>

You can always run the program by using the absolute path (the installation script's file-name including its location starting from the / of the file system, `/usr/bin/vmware-config.pl`). After the configuration program has completed configuring your host, be sure to exit from the root account by using the `exit` command.

The help system built into VMware Workstation for Linux depends on a Web browser being accessed from the `/usr/bin/netscape` directory. If your browser is located elsewhere (Netscape, Mozilla, or otherwise), be sure to create a symbolic link to it at that location (for example, `ln -s <browser path> /usr/bin/netscape`).

VM Host Tuning Tips

Demands placed on high-load multiprocessor servers are extremely different from the loads placed on a single-processor notebook. Knowing how to tweak your system to get dramatically better performance is no secret. Many knowledge-base articles, white papers, forum discussions, and magazine articles are littered all over the World Wide Web regarding performance tuning. Surf the Web, and see what can help you the most. Please be advised, though: although you can experience significant performance gains by turning a few software knobs for your OS, you should test any performance tweak in a test environment first. You're better off having a slow server than no server; therefore, thoroughly test all your system changes in the lab first.

The tuning tips in this section cover hardware, Linux OSs, and Microsoft OSs. These tips will give you a general idea of where you can begin to start squeezing horsepower out of your hosts for the benefit of guest VMs. Some of these tips are common sense; not all will apply to you, but the following tips will get you into the mind-set that a custom install can be a whole lot better than the default configurations that manufacturers offer you:

- Download and install the latest BIOS for your host hardware.
- Delete existing partitions, including manufacturer support partitions.
- Create one large host system partition and format using modern partition types—NTFS, ReiserFS, EXT3, or XFS.
- Install a better host OS: Windows XP or Windows 2003/Red Hat Workstation or Advanced Server.
- Install updated manufacturer drivers, and stay current with new versions.
- Install current service release, hotfixes, and security patches (including antivirus).
- Review, stop, and remove any unneeded services.
- Upgrade to Gigabit Ethernet, and team multiple network adapters.
- Perform a clean operating system install from scratch; don't install unneeded packages or applications.
- Stop unneeded applications from initializing by removing them from startup or deselecting them during the setup.

- Religiously remove temporary files and spyware; eliminate command/Web histories and file caches.
- Remove unnecessary networking protocols, and change the protocol binding order as necessary.
- Remove unused TTY virtual consoles from `/etc/inittab`.
- Performance-tune your hard drive with `defrag` or `hdparm`.
- Compile a custom kernel.

Though most of the tuning tips in this section can be used across many of Microsoft's OSs, we'll specifically address Windows XP because of its ubiquity. This section contains a basic performance recipe specifically geared to give you the most bang for the time you spend wrenching on your OS. Before moving ahead, start with a fresh install of Windows XP, and before tweaking any machine in a production environment, always test your performance changes in the lab first. This section touches on some knob turning and switch flipping you can do to optimize an operating system. The Internet also has many reputable sites and forums where you can find valuable tuning information.

Windows Update: Use Microsoft Windows Update to ensure your computer is patched with the latest service and security releases and any critical hotfixes. You may need to run Windows Update several times, and you may need a few reboots as well.

Hardware updates: With the operating system taken care of, you can look to your computer's hardware. If your computer is produced by a major manufacturer, such as Dell or Hewlett-Packard, you can download updates from their support Web sites. If you're your own master, you can download updated adapters and motherboard drivers from component manufacturer Web sites, such as Intel, 3Com, Creative, and Nvidia. Device Manager will list the components installed in your computer and is a good place to begin figuring out what hardware needs updating. You may have to crack your system case open to get chipset identification numbers. If you can't find the manufacturer of your devices, use a good search engine for help, such as Google.

Desktop settings: If you have an extra pretty desktop, you'll need to do with less to get the last bit of performance from your computer. Each time Windows boots, it keeps the background in memory, and you can kiss those resources goodbye. Go with a black background. (it's really easier on the eyes) or limit yourself to a small tiled picture—one that's nearer to 0 kilobytes (KB) than 1KB. If you're convinced it's okay to do away with the pictures, select the Display icon in Control Panel, and follows these steps:

1. From the Themes tab, Select Windows XP Theme.
2. From the Desktop tab, set Background to None and Color to Black.
3. From the Screen Saver tab, select None.
4. Click the Power button, and choose Presentation under Power Schemes.
5. Ensure that the Enable Hibernation setting is disabled on the Hibernation tab.

6. From the Appearance tab, select Effects. Deselect all the check boxes.
7. From the Settings tab, select a resolution that allows for good guest viewing.

Delete unused applications: Windows XP, like many modern operating systems, suffers from application bloat. You should uninstall unneeded or useless applications to recover disk space as well as reduce the amount of things firing off on system boot. The `sysoc.inf` file, generally found in the `c:\windows\inf` directory, is a good place to begin putting XP on a diet. You'll need to edit it with Notepad. Removing the HIDE option allows previously unviewable applications to be displayed in the Add/Remove Windows Components utility in the Control Panel. You can blow away everything you don't think you'll need, such as MSN Explorer, Pinball, and so on. You can always reinstall them later; follow these steps:

1. Select Edit ► Replace.
2. In the Find What field, type **HIDE**, and leave the Replace With field empty.
3. Select Replace All.

Startup applications: You can do without nearly all the applications next to the system clock, except antivirus. If you want to get a good idea of how much space these applications are using, launch Task Manager and look in the Mem Usage column. Add the totals from instant messengers, CD-burning software, personal information devices, sound and video card add-ons, and so on. You really don't need all those applications running, so create a shortcut to these applications in the System menu and then blow them out of the registry and the Startup menu. Before editing, be sure to back it up. The registry key you want to focus on to prevent these applications from starting is `HKEY_CURRENT_USERS\Software\Microsoft\Windows\CurrentVersion\Run`.

Recycle Bin: The Recycle Bin can needlessly chew up disk space with today's hard drives exceeding 300GB. For instance, the default reserved size is 10 percent of the total disk. If you have a 250GB hard drive, your Recycle Bin will be a whopping 25GB in size! If you're worried about losing things you delete, you can back up things you think you may need later to a CD-ROM. Set your Recycle Bin to as close to zero as you're comfortable with, as in 1–3 percent of the total disk space.

Remote desktop: Because virtualization applications give you remote access to your VMs, you probably won't need Remote Desktop Sharing. In addition, if you're your own help desk, you can go ahead and disable Remote Assistance. To disable these two services, follow these steps:

1. Right-click the My Computer icon, and select Properties.
2. Select the Remote tab.
3. Deselect the Remote Assistance option and the Remote Desktop option.
4. Click OK to finish.

Windows firewall: In our highly litigious society, manufacturers are trying to idiot-proof everything. For Microsoft, this means including a bulky firewall. Though most of us will agree that this is a service that's long overdue, you can disable the firewall. Before doing

so, make sure you're aware of the risks involved and you have another firewall (hardware or software) in place. We don't necessarily recommend disabling the firewall and are intentionally not covering the option to disable it, but you can use the Microsoft Knowledge Base, or you can perform a quick search on a good search engine for assistance.

Disable extraneous services: XP fires up many services on boot to make using the computer a bit easier for novices. If you're a long-time computer user, you may just be too tired to worry about all the overhead Microsoft throws on a stock operating system. Take the time to trim a few seconds off your boot time and give your processor more idle time by stopping unnecessary services. You can view the running services on your computer by going to the Control Panel and selecting Administrative Tools and then Services. Before stopping services and setting the Startup Type field to Disabled, you may want to take a screen shot of your current settings and print it out in case it's necessary for you to revert to any previous settings. Assuming you don't need them, you can safely stop the following services and set them to Disabled:

- Alerter
- Automatic Updates (do manual updates instead)
- Background Intelligent Transfer Service
- ClipBook
- Error Reporting Service
- Fast User Switching
- Indexing Service
- Messenger
- Portable Media Serial Number
- Smart Card
- Smart Card Helper
- System Restore
- Terminal Services
- Themes
- Uninterruptible Power Supply
- Upload Manager
- Volume Shadow Copy
- Web Client

During the process of paring down services, sometimes services fail to stop or hang, leaving a slew of orphaned child processes. You'll find yourself performing triage at the CLI with the stock Windows `net start/stop` command and praying you won't have to reboot.

If this happens, you invariably will have to reboot and start the whole process again. Rather than using the stock Windows legacy command, get your hands on `pulist` and `kill`. If they're not installed on your XP system (or other Windows OS), you can download `pulist` from the Internet at <http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/pulist-o.asp> or get it off the Windows 2000 Resource Kit. You can acquire `kill` from many places, such as from the Windows 2000 Server CD-ROM in the Support/Tools folder. You may have to use `setup.exe` to install the tools to get your copy of `kill.exe`. After copying both commands to the root of your hard drive, use `pulist` at the CLI to view the running process. Find the hanging process, and identify its process ID (PID). Then type `kill -f <process id>`. Type `pulist` again to verify your results.

Assuming the process is still hanging on, you have a couple more choices before a reboot. You can use the AT command to stop the service: `at <time> /interactive cmd /c kill -f <process id>`. Lastly, you can use the Process Viewer (`pview.exe`) from the Windows 2000 Server CD-ROM to adjust the permissions of the process so that `kill` will work. Gaining access to `pview.exe` is identical to that of the previous `kill` command.

If you have an application that refuses to quit working after reboot, you can use the Microsoft Windows 2000 Scripting Guide to create a script at terminate processes. You'll find the following link to be invaluable in helping suppress unruly applications: http://www.microsoft.com/resources/documentation/windows/2000/server/scriptguide/en-us/sas_prc_fgbg.mspx.

Listing 14.14, at that site, can suppress any service that launches. The script loops and will eat up some processor time; however, it's good at keeping a wily application under control. All you need to do is type the script into Notepad and save it with a `.vbs` extension. Supply the script with the name of the application executable from Task Manager you want it to monitor, and when it starts, the script will take care of the rest.

Memory tweaking: You can get added performance out of XP by tweaking the memory. You'll need to edit your registry, so back it up first. Make only one change at a time to the system, and then thoroughly test it. You may find that you get negative results. If this is the case, you can reverse course and undo any changes. You can make three changes to the registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management` to squeeze out a bit more performance:

- You can disable the Paging Executive to prevent XP from paging data from RAM to the hard drive; this tweak keeps the kernel-mode drivers and system code in memory. Assuming you have a large amount of RAM, greater than 512MB, the net effect of leaving the data in RAM may yield a performance increase. You'll need to change the `DisablePagingExecutive` key value from 0 to 1. This tweak can negatively impact hardware and is generally reserved for troubleshooting software, such as debugging drivers. If you run into problems, you'll need to revert the setting to 0.
- You can run the Windows XP kernel in memory to provide a modest performance boost. You can accomplish this by changing the value of the `LargeSystemCache` key from 0 to 1. Windows XP will then allocate all memory, except 4MB, to the kernel. The 4MB of memory is reserved for disk caching, and Windows XP is capable of allocating more as needed. This registry tweak allocates memory-to-disk caching and keeps changed pages resident in memory until available pages drop near 250.

- You can increase I/O performance for large file transfers by editing or creating a REG_DWORD value labeled IoPageLockLimit. In general, a setting equal to 12MB will find the performance sweet spot. However, you may have to use trial and error. The value entered for the key needs to be in bytes (1MB = 1,048,576 bytes).

Paging file: For Linux and Windows operating systems, most computer professionals agree that the paging file should be 1.5–2.5 the size of the physical RAM. Having a larger paging file will help you to run more VMs under VMware VM applications, but it will do little for you with Microsoft VM applications. Regardless of the multiple you choose, you can increase performance by placing the paging file on a separate hard drive and disk controller. You can optimize the paging file by opening the System icon in Control Panel and following these steps:

1. In the Performance area, select Settings, and click the Advanced tab.
2. Under Virtual Memory, select Change. Set the paging file on the C drive to 0.
3. Now, select a different hard drive, and set the minimum and maximum sizes to your calculated choices.
4. As a bonus, on the Advanced tab, select Settings from the Performance section. Select Adjust for Best Performance, and then click OK.
5. Reboot your system after this change.

DLLs: Windows XP generally doesn't unload DLLs after programs close in order to speed up subsequent restarts of an application. You can edit the registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer and add the DWORD value, AlwaysUnloadDLL, with a setting of 1 to force DLLs to unload on application close.

File system: Windows XP has the ability to use NTFS partitions. If you're using a FAT-partitioned or FAT 32-partitioned hard drive, convert it to NTFS. You can convert the file system on a hard drive by using the convert command at the CLI. If you need help with it, type `convert /?`. Don't forget to regularly defragment your hard drive. Whether you use a direct-attached single drive, an array of RAID disks, or a logical disk on a SAN, defragment your systems. Don't get caught believing that defragmentation software directly rearranges disk clusters: it doesn't. Disk defragmentation software defragments logical cluster numbers (LCNs) stored in a table. It's possible that the logically contiguous LCNs will be physically fragmented by the RAID or SAN disk controllers; however, performance is realized because the file is in the best possible place for physical access, and the file system can access the file at a single logical location.

Networking: Assuming you have a network adapter with an intelligent processor, you can offload processor tasks to the network adapter by editing or creating DisableTaskOffload as a REG_DWORD value in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters and setting its value to 0.

Some people may have you believe that performance tuning is a secret black art, but it really is just the application of experience. The trick to making sure you're building the fastest and most reliable operating system to host your guest VMs is testing each change before making additional changes and clearly documenting all your changes. If you find a secret formula

that works for your environment and you can't duplicate your success, you've wasted a lot of time and effort. Don't fall into the trap of thinking you'll remember what you've done—you probably won't. The IT field is saturated with so much good information that it's impossible to keep everything mentally organized: a printed work journal will never let you down when it comes time to reproduce a system in a disaster.

Summary

In this chapter, we covered installation and optimization techniques for deploying workstation-class VMs on desktop operating systems, including VMware Workstation and Microsoft Virtual PC. We gave you general tips for tuning VM hosts and specifically covered how to optimize Windows XP for added performance. In the next chapter, we'll cover installing and managing virtual machines for workstation-class applications. This will include prestaging, backing up, and monitoring guest VMs.



Deploying and Managing VMs on the Desktop

In this chapter, you'll build on your knowledge of desktop virtualization application installation from the previous chapter and turn to deploying and managing workstation-class guest VMs. We'll continue to use "how-to recipes" and will cover the basics of backing up VM configurations, configuring VMs to run as services, and monitoring VM performance. We'll also give you a dose of VM command-line administration. If you're looking for the complete treatment of VM backups, you'll want to refer to Chapter 7 for details.

Deploying VMs with VMware Workstation

Creating VMs for VMware Workstation is a straightforward process and is dramatically simplified by virtualization software's well-thought-out wizards. Though you should know what you want in a VM before creating it, you can always edit the VM after it's created. We'll go through the process of creating VMs on Windows and Linux hosts; you can simply skip to the sections you need. After discussing how to create guest VMs, we'll cover individual hardware options and show how to migrate guest VMs between host systems.

Note With the introduction of ACE from VMware, you can bring best-practices enterprise-class management to end users when deploying VMs. You can roll out standardized computing environments that adhere to corporate security policies and application management strategies while simplifying support. ACE achieves its magic by leveraging a VM and the policies you set for it. Visit VMware's Web site to read more and download a demo copy.

Building a Windows VM is identical to building a Linux VM. To start, launch VMware Workstation. For Windows host systems, select Start ► Programs ► VMware ► VMware Workstation, and then select File ► New Virtual Machine to launch the New Virtual Machine Wizard. To launch VMware for Linux, make sure X Windows is started, type `vmware` in a terminal window, and then select File ► New Virtual Machine.

As with all installs, you should customize the installation options. Wizards are great, but you want to be in as much control of all installation procedures as possible. To that end, select Custom as your choice for the initial VM configuration.

You'll be required to select which guest operating system type to install. For the example's install, we'll select the first choice, Microsoft Windows. If you want to install a Linux VM, choose Linux.

From the Version drop-down menu, select the OS you'll be using for the guest VM. As you've probably noticed, you have one heck of a selection for guest VMs (see Figure 4-1). Though it's not listed, you can install DOS 6.22 by selecting Windows 3.1. For the discussion's purposes, we'll be configuring Windows 2003 Server. Regardless of what you choose, these directions will be identical to the choices you'll need to make.



```
Windows 3.1
Windows 95
Windows 98
Windows Me
Windows NT
Windows 2000 Professional
Windows 2000 Server
Windows 2000 Advanced Server
Windows XP Home Edition
Windows XP Professional
Windows Server 2003 Web Edition
Windows Server 2003 Standard Edition
Windows Server 2003 Enterprise Edition
Windows Small Business Server 2003
Longhorn (experimental)
```

Figure 4-1. *VM configuration selection type*

You'll now have to select a name for your VM as well as a storage point for the guest VM's virtual hard disk. Though you can change the name and move the location of the virtual hard disk later, it's important to make wise choices now. First, you don't want to have to create more work for yourself than necessary. Second, changes disrupt workflow and don't always turn out as predicted. Not to be too didactic, but work smarter and plan your work. If you need to take a few moments to figure out what to call your system, take the time. If you need to decide on a virtual disk storage location, take additional time to create necessary directory structures now.

Naming conventions for computer systems should be meaningful. Some network administrators use cryptic host names in an effort to achieve a greater level of security; however, you don't get much security through obscurity. (Fingerprinting the TCP/IP stack of a host will pretty much tell anyone what they want to know anyway!) Conversely, you don't want to give out more information in a host name than necessary. A host name should have enough information to help you manage your systems. For example, naming computers after trees or planets might not be helpful if you're dealing with many systems or working late in a disaster

situation. Many corporations name systems after an office's physical location and include some type of IP addressing scheme information. For instance, a host might be named LAXVM2K10, where LAX stands for the Los Angeles office location and VM2K stands for a Windows 2000 VM. The last two digits, 10, represent the subnet ID. The Web has plenty of opinions and good tips on naming conventions. In the end, settle on something other than chaos, and stick to it.

As you can see in Figure 4-2, the guest VM for this example is named `vmw2k3-1`. We also created a separate storage directory for this VM, `D:\VirtualMachines\vmw2k3-1`. For Linux systems, adjust your pathing constraints as necessary (for example, `/root/VirtualMachines/vmw2k3-1`). As you probably already noticed, this isn't the default directory. Remember from the discussions in previous chapters that you don't want the host OS and guest VMs to use the same hard disks or same controllers. In this example, we're using the secondary SATA controller and an additional hard drive to increase VM performance. If you're in a lab or educational environment and don't have access to expanded resources, simply use the default directory—just create a custom directory for each guest VM. It makes management so much easier by having all the files for each VM stored in one place. Once you're satisfied with your naming conventions, continue with the install.

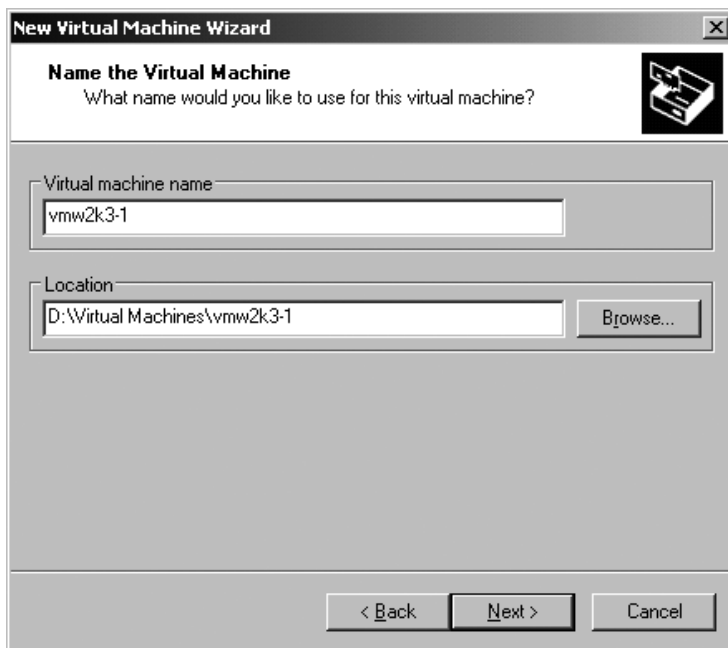


Figure 4-2. VM name and storage directory

The wizard will recommend the amount of RAM to use for the guest VM. Notice that in Figure 4-3, the wizard reveals the minimum, recommended, and maximum amount of RAM to use for the guest. These are merely recommendations. You should perform a bit of research to determine the amount of RAM your guest OS requires. In lab and learning situations where RAM is limited, feel free to throttle memory back. In production environments, be sure to allocate as much RAM as you'd normally provide for a system. As depicted in Figure 4-3, our example will use 256MB of RAM. Select Next to continue.

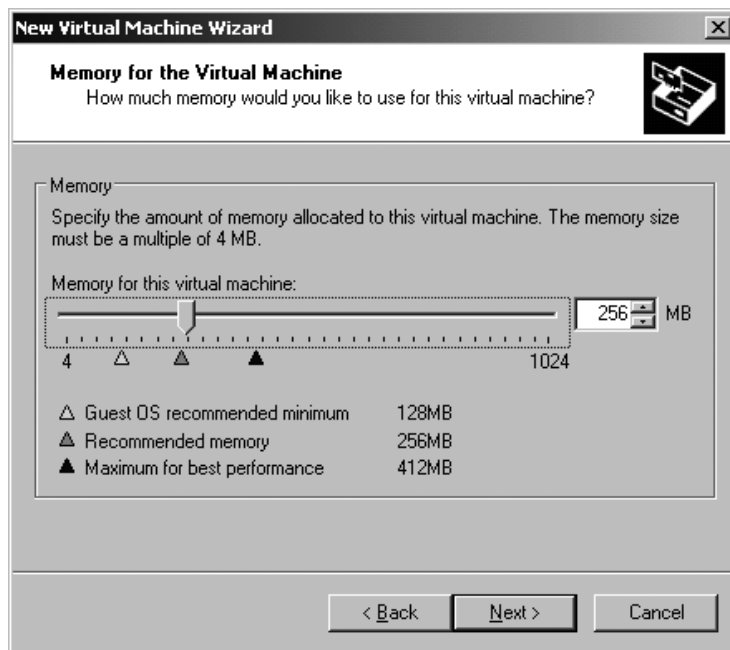


Figure 4-3. VM memory allocation

You'll now be required to select the type of networking you want to make available to your VM. If you want to hook up your guest VM as quickly as possible to your test or production network, select Use Bridged Networking. If you want this guest VM to be in a private network, select Use Host-Only Networking. Choosing NAT will grant you the ability to have a private VM network that has access to your production LAN: this is a one-way relationship. As shown in Figure 4-4, we'll use the default of bridged networking.

You'll now have to select which I/O adapter type to use in your guest VM. An easy way out is to use the BusLogic default. However, you'll get a little better performance from the LSI Logic option. As a word of caution, you'll need to have the driver available for when you install the actual operating system on the guest VM.

Tip It's a good idea to download the LSI Logic driver for your VMware guest VMs. It's well worth the effort. Additionally, you'll want to copy the driver to a floppy disk and make an ISO image of the disk. You can then mount the image as if it were a physical floppy drive and have it available during the guest OS install. You can download ISO image-making software, such as MagicISO from <http://www.magiciso.com> or Undisker from <http://www.undisker.com>, on the Internet for a trial period.

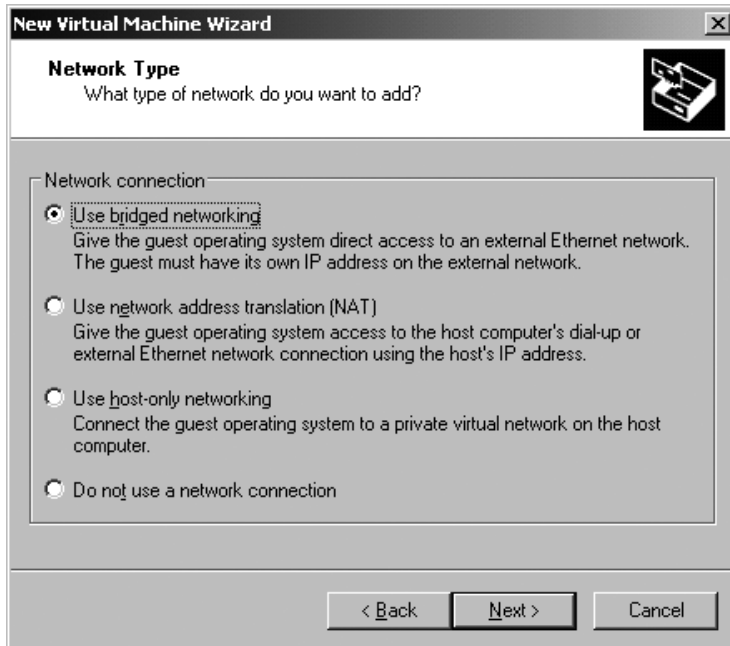


Figure 4-4. *Network type*

Because we don't want our workstation to mount a physical disk in this example, we'll use the default disk type selection of Create a New Virtual Disk. Now, you need to decide which type of disk to use with your VM. You have the option of using a virtual IDE or SCSI disk despite the physical medium in the host. Virtual IDE is limited to two devices per channel, and virtual SCSI is limited to seven. For this example, we'll choose SCSI.

On the Specify Disk Capacity screen, you'll need to decide the total maximum amount of space to be used by the guest VM. You'll also need to decide if you want the disk to be dynamically expanding, which is the default, or fixed. Fixed disks have the space allocated up front, and the space will always be available to the guest. Dynamic disks expand as necessary. This means the disks will grow to the maximum size specified.

You'll get better performance with fixed disks because dynamic disks tend to fragment your hard drive. If you're in an educational or lab environment, dynamically expanding disks tend to work well. For production environments, you'll want to stay with fixed disks. Because our example is geared around a Windows 2003 Server and our test system has sufficient disk space, we'll select the option **Allocate All Disk Space Now**.

As for how many gigabytes to use for the system disk, this is really a matter of preference and experiential knowledge. In the NT days, 4GB was sufficient. Today, with the code bloat involved with Windows 2000 and 2003, you'll find you'll want a disk with 8GB or even 12GB reserved for the virtual hard disk. The value of having ample space is that you can copy the i386 directory from the Windows CD, service packs, and drivers to the hard drive for future use. Having plenty of room also allows you to keep log files longer.

Caution You have the option to split virtual disks into 2GB files. If your host's operating system doesn't support files larger than 2GB, like EXT or VFAT, you'll need to select this option during virtual disk creation. For some Linux operating systems, the wizard will autoselect the **Split Disk into 2GB Files** option.

If you choose to preallocate space for your guest's virtual disk up front, the system will inform you that it will take a moment to create the disk. You'll be informed that the system may appear to hang, so be patient during virtual disk creation.

The wizard will prompt you for the name of the virtual disk and the location where you'd like to have it stored. Choose the same location and naming convention you established earlier in the VM creation process. In our example, we used `vmw2k3-1` as the name of the server, so we'll also use it for the name of our virtual disk. Click the **Browse** button to specify the destination directory for the virtual disk. For this example, our destination directory is `D:\VirtualMachines\vmw2k3-1\vmw2k3-1`. For Linux, your path will look similar to `/root/VirtualMachines/vmw2k3-1/vmw2k3-1`. When it comes time to service your VMs, you'll be glad that all files related to any specific VM are in one folder.

Take a moment to click the **Advanced** button and explore these options. On this screen, you can specify the virtual disk's device node and the disk mode (see Figure 4-5). When using virtual SCSI disks, the wizard creates the virtual machine with one SCSI device, 0. The virtual disk in our example is the first hard disk, 0. You can select the virtual disk mode from the **Mode** section. You can set the disk to **Independent Persistent** or **Independent Nonpersistent**. If you want to easily use snapshots, leave the default disk mode as a persistent disk. Click **Back** to continue, and then click **Finish**.

To help you better understand nodes, take a moment to look at Figure 4-6: this is what the **Specify Advanced Options** screen looks like for virtual IDE disks. Notice it shows only four nodes. You know that VMs support a maximum of two devices on each virtual IDE controller. In Figure 4-6, the virtual disk is bound to the first controller, 0, and is the first device on that controller, 0. The virtual CD-ROM drive is on the second virtual controller, 1, and is the first device on that controller, 0.

VMware will present you with a progress indication during the creation process. Depending on the host's hardware, it will take five to ten minutes to create an 8GB virtual disk.

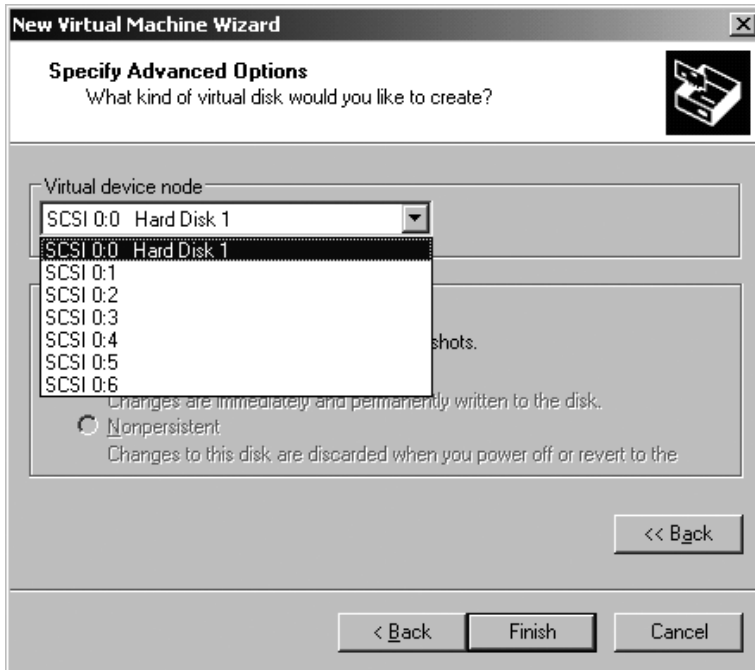


Figure 4-5. SCSI virtual disk advanced options

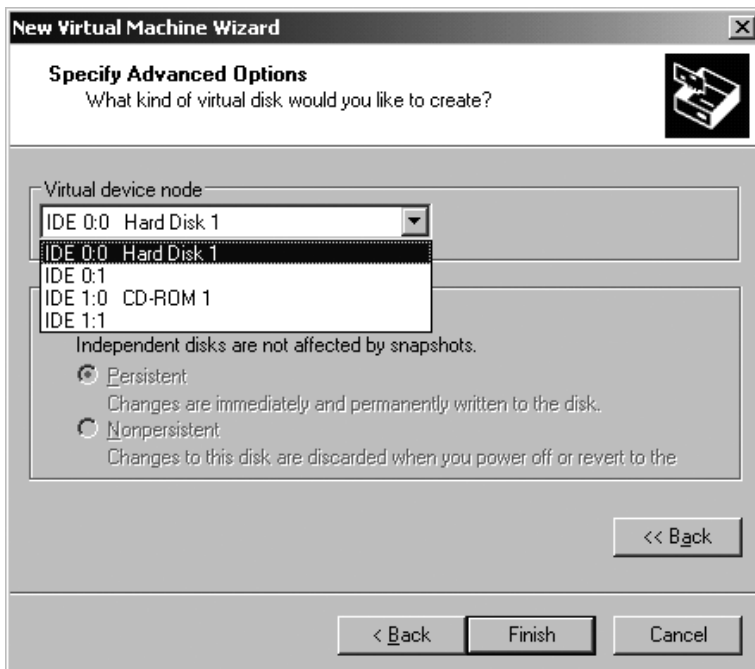


Figure 4-6. IDE virtual disk advanced options

When the disk creation process is complete, you'll be returned to the Workstation Management User Interface (MUI). Notice in Figure 4-7 that our newly created VM appears in the left column under the Favorites heading. Favorites are merely shortcuts to your VMs.

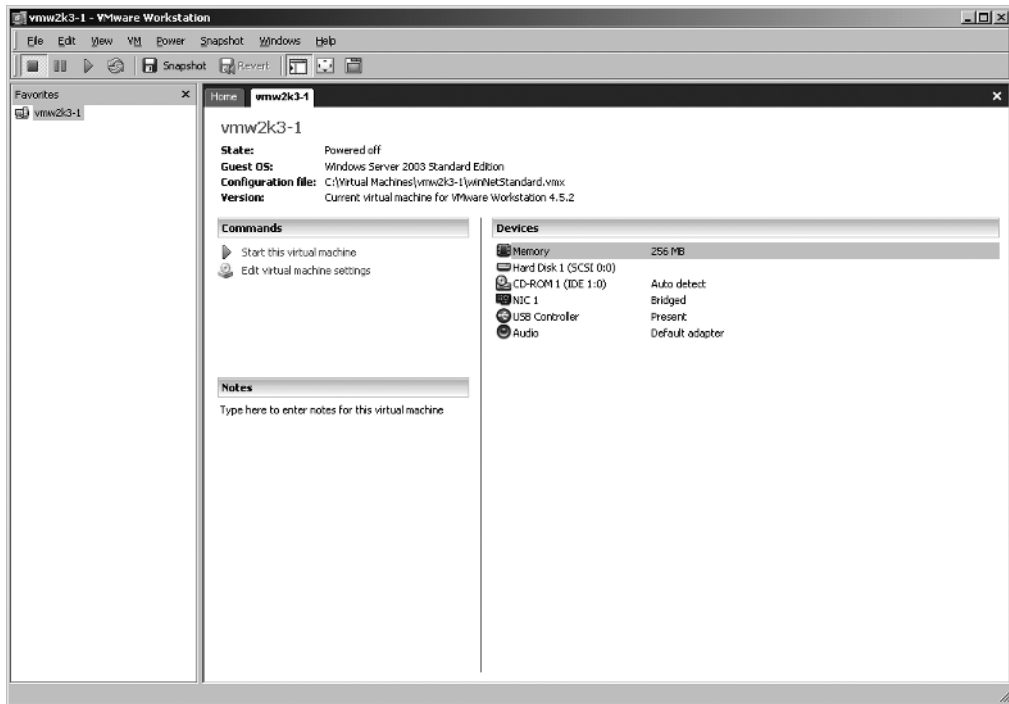


Figure 4-7. *Newly created VM with virtual devices*

Before powering on your new virtual machine, take a moment to look at the VM's hardware settings using the Configuration Editor by selecting Edit Virtual Machine Settings.

Installing VM Tools

After installing an operating system on your guest VM, you'll want to install VMware Tools. The VMware Tools package supplies significant enhancements to a VM and provides extra functionality between a guest VM and the host system. For instance, adding VMware Tools facilitates drag-and-drop support between a guest and the host, increases graphics performance with an optimized video driver, supports time synchronization between guests and the host, and supports seamless mouse transitions between guest VMs and the host system.

VMware Tools is available for you to install by using ISO images compiled in VMware Workstation, and it supports a gamut of Microsoft Windows OSs, from Windows 9x through Windows 2003, as well as many Linux distributions. As a technical note, during the installation process, VMware Workstation temporarily mounts the ISO files as the guest's first CD-ROM drive.

VMware Tools for Windows

In the example that follows, we'll install VMware Tools on a Windows XP guest VM. To initiate the installation of VMware Tools for Windows, start by selecting VM ► Install VMware Tools from Workstation's console menu. You'll be presented with a preinstallation pop-up menu informing you that the VM to receive the tools must be running.

On the Setup Type selection screen, select Custom as your install choice. As a rule of thumb, you really never want an application to automatically make choices for you. Even if you opt for the defaults during a custom install, you'll be aware of the installation points and the services to be loaded in case you need to troubleshoot any problems.

The Custom Setup configuration screen, shown in Figure 4-8, has three main options: Toolbox, VMware Device Drivers, and Shared Folders. If you take a moment to click each option, the Feature Description area will display some details about the option selected. Briefly stated, though, the Toolbox feature improves the usability of a VM by using a group of utilities; the VMware Device Drivers option installs performance-enhancing drivers for the virtual video, mouse, SCSI, and network adapters; and the Shared Folders option facilitates the sharing of files between the guest VM and the host. If this is your first installation, install all the drivers; you'll want to experiment with each. If you're not satisfied with the installation destination of the tools, click the Change button to select a different destination directory. When you're done, select Next to continue.

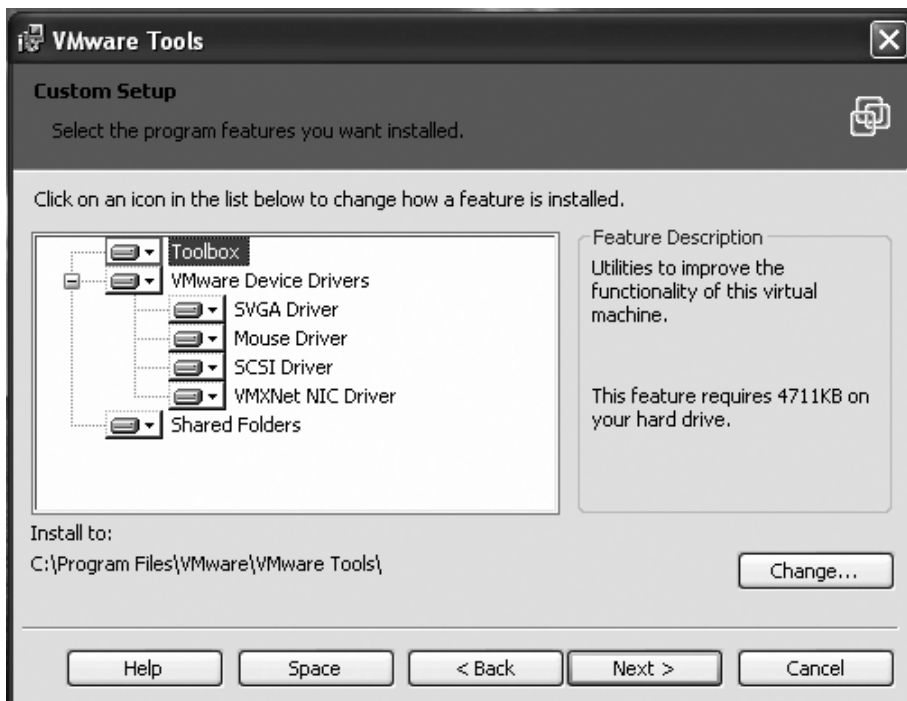


Figure 4-8. Custom setup configuration

The last step to install VMware Tools is to click Install on the Ready to Install screen. If you doubt any selections you previously made, take a moment to review your choices by clicking the Back button (and then click Install to begin the installation).

The wizard will take a moment to deliver the software packages, and during the delivery process you'll receive several alerts, as shown in Figure 4-9. The alerts state that the VMware Tools drivers haven't passed Windows Logo testing. This is okay, so select Continue Anyway on each of the driver alerts. You'll also notice that some of the drivers, such as the mouse driver, activate during the install. You can seamlessly move your mouse between the guest and host windows. After completing the install, be sure to reboot the VM.



Figure 4-9. *Windows driver alert*

If you don't have VMware Tools installed, you'll have to press Ctrl+Alt to release the focus of the mouse from the guest VM and return focus to the host system. The mouse will also underperform in that it will appear to move erratically. Without the video driver, the guest's display will have limited resolution capabilities, and the SCSI drivers simply add better performance characteristics for attached virtual devices. With the tools installed, take a moment to optimize the performance of Windows XP by referring to Chapter 3.

Tip With virtualization application technology storming the IT front, you'll want to read every possible source of information you can. VMware's online documentation and community forums are an excellent place to get valuable information about configuring guest VMs. For instance, you can find detailed information on installing VMware Tools for each Windows OS at http://www.vmware.com/support/ws45/doc/new_guest_tools_ws.html.

VMware Tools for Linux

The VMware Tools package for Linux is built into VMware Workstation and includes all the same great benefits and ease of installation as Windows Tools. You don't need to download any additional software or use physical media. The easiest way to install the tools is to log into the VM as the root user. If you don't, you'll need to invoke the `su` command with a typical user account. Keep in mind that the example code that follows may not work with your particular distribution; therefore, make adjustments as necessary.

To begin the installation, power on the guest VM to receive the packages in text mode; the tools can't be installed if X Windows is running. After the guest VM has completely booted and you've completed the login process, select **File ► Install VMware Tools** from the Workstation console menu.

You'll now need to mount the virtual CD-ROM and then extract the installer files. Before installing the tools, you should also umount the CD-ROM, like so:

```
mount /dev/cdrom /mnt/cdrom
cd /tmp
tar xzf /mnt/vmware-linux-tools.tar.gz
umount /mnt/cdrom
```

You execute the installer by changing to the package delivery directory and executing a Perl script, like so:

```
cd vmware-tools-distrib
./vmware-install.pl
```

Initiate the Linux GUI, and open a terminal session, where you can start the VMware Tools background application:

```
vmware-toolbox &
```

If you want the tools to start automatically when the GUI initializes, add the `vmware-toolbox &` command to your system's startup programs. If after adding the tools your guest VM becomes unstable, you can remove the tools by executing `vmware-uninstall-tools.pl` with root privileges.

VMware Virtual Hardware Options for Windows and Linux

Like physical computers, VMs have an array of peripheral options you can install or remove. Additionally, you can even configure the BIOS of your VM after you power it on. To get the best possible performance out of your guest VM, be frugal with your system's resources. For instance, do you need an audio device or USB device in a server? By paring down the devices in your VM and disabling components in the BIOS, you can really start to ratchet up performance.

Removing devices from a VM is straightforward. Open the MUI, and select **Edit Virtual Machine Settings**. Highlight the device to be removed, and then select **Remove**.

In Figure 4-10, our example VM has a USB and an audio device installed. We have no real need for the devices, so we simply highlight each device, and select **Remove**. Once the device is removed, it won't be available to the VM's operating system. If at any time you need a device, simply power off the VM and add the device.

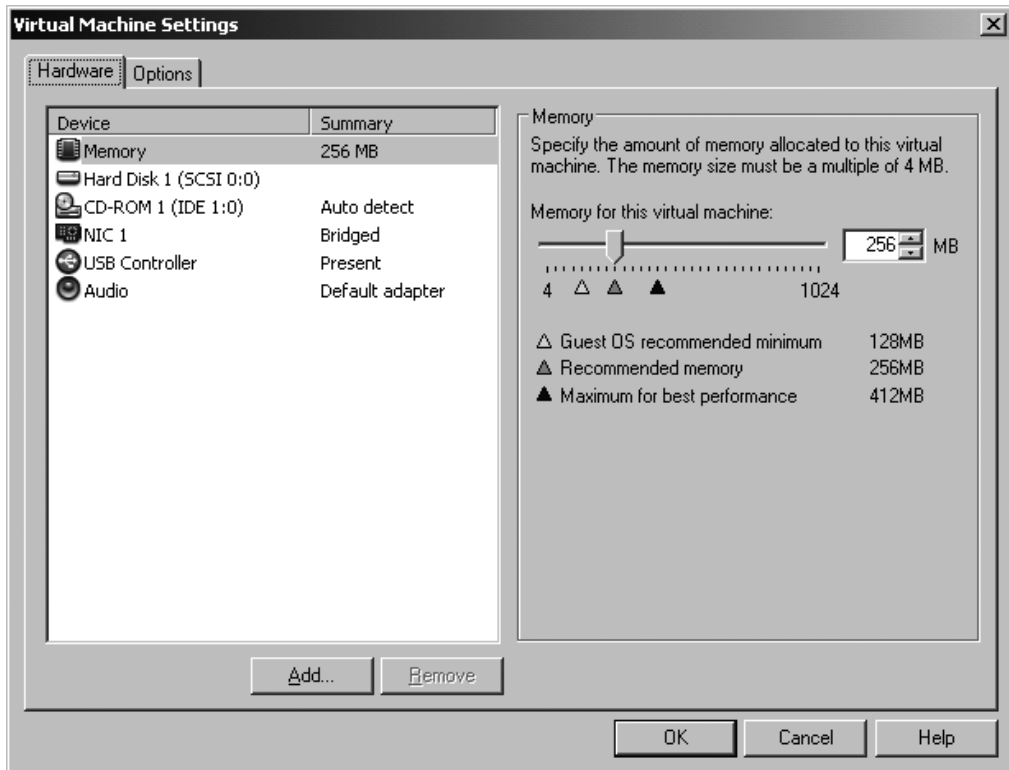


Figure 4-10. Virtual machine hardware settings

If the server you're building is lacking hardware, such as an additional virtual NIC or virtual CD-ROM drive, you can add the components by clicking the Add button. Because VMs are capable of connecting to ISO images as virtual CD-ROMs (or floppy drives), you may find it useful to add a second virtual CD-ROM drive to your system. Before adding a second CD-ROM device to this VM example, look at the Add Hardware Wizard that's presented after clicking the Add button. On the Add Hardware Wizard greeting screen, select Add to continue.

The Hardware Type selection screen, shown in Figure 4-11, presents you with VMware virtual hardware options. You have the ability to add many virtual devices, including hard disks, DVD/CD-ROM drives, Ethernet adapters, and serial ports. If this is your first time creating a VM or adding hardware to an existing VM, take a moment to add one of each device to a VM to get a feel for the process and to see what options are available to you.

Going through each device in the wizard may seem like overkill, and it will be tiresome for some. However, we want to cover the options that may not be apparent. We'll discuss the major points of each hardware option and show relevant figures. Additionally, you can skip to the section you need for information on adding a particular device.



Figure 4-11. *Hardware type selection*

Hard Disk

When you add a hard disk using the Add Hardware Wizard, you'll get three choices: Create a New Virtual Disk, Use an Existing Virtual Disk, and Use a Physical Disk. Adding a new hard disk to an existing VM is virtually identical to the process of preparing a drive while creating a guest VM. If you're creating a VM to connect to an existing virtual hard disk, simply browse to the location of the preexisting virtual disk. Remember that virtual disks will end with a *.vmdk extension.

Using a physical disk for a VM is really the only tricky option. Generally, using physical disks should be left to advanced users. Conversely, you'll never become an advanced user if you don't try to use physical disks. To that end, don't let the warning in Figure 4-12 scare you away from the performance benefits of using physical disks for your VMs.

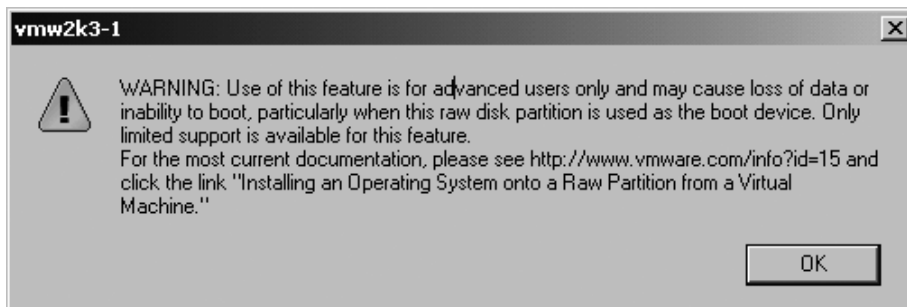


Figure 4-12. *Physical disk usage warning*

On the physical disk selection screen, you'll need to choose between using a whole disk and using a partition of one. If you choose to use the entire disk, you'll be asked to provide a virtual disk filename that will store the partition access configuration information. Remember to make any necessary directories to house the file, and be sure the filename is meaningful.

If you elect to use a physical partition, you'll be presented with a Select Physical Disk Partitions screen. You'll need to select which partition on the physical disk will be used.

Tip If you find you have plenty of space on a single partition hard drive and wish you had partitions available for guest VMs, you can repartition an existing hard drive without losing your data using third-party repartitioning utilities. Norton PartitionMagic is an excellent product to aid you in repartitioning a Windows system. You can learn more about PartitionMagic at <http://www.symantec.com/partitionmagic>.

Remember to make necessary directory name and filename changes for the new virtual disk. As always, make sure your naming conventions are consistent and meaningful. If you need to configure the virtual disk to be independent, click the Advanced button during the creation process on the Specify Disk File screen.

DVD/CD-ROM Drive

When adding a DVD/CD-ROM drive to your guest VM using the Add Hardware Wizard, you have the option of using a physical device or a software image, as noted in Figure 4-13. VMware Workstation has the ability to autodetect a physical device, or you can specify a device. Additionally, you can employ legacy emulation by clicking the Advanced button on the physical drive selection screen. If you use an ISO image, you'll need to specify the location of the ISO file. Like virtual disks, you can specify the device node by clicking the Advanced button on the Choose an ISO Image screen.

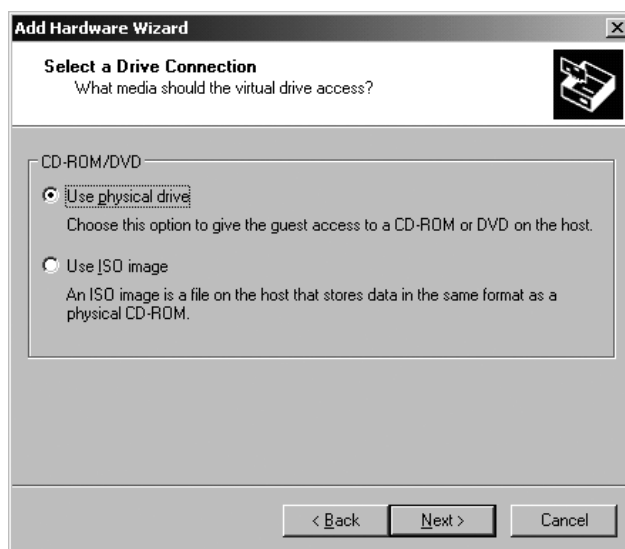


Figure 4-13. DVD/CD-ROM drive connection

Floppy Drive

Creating a virtual floppy drive is similar to creating a virtual disk. The wizard will present you with three options: Use a Physical Floppy Drive, Use a Floppy Image, and Create a Blank Floppy Image. There's no real trick to using any of the three options. You just need to decide what's going to suit your needs and then decide if the guest VM will connect to the VM when powering on. Using floppy images is a great way to keep from having to use sneaker net.

Ethernet Adapter

Adding Ethernet adapters to a VM will also require you to select the network connection type to be used. You have four available options: Bridged, NAT, Host-Only, and Custom. If you're unsure of which type to use, please refer to Chapter 2 where each network type is discussed in detail. When you're done making your choices, select Finish to complete the process.

Sound Adapter

When adding a sound adapter to your guest VM, the only real issue you'll need to be concerned with is if you have more than one sound adapter installed. If this is the case, you'll want to specify the sound adapter by selecting it from the drop-down list under the Connection heading (see Figure 4-14).



Figure 4-14. *Sound adapter installation*

USB Controller

Adding USB support requires you to decide if you want the guest VM to automatically attach to a USB device when it has focus on the host. If you desire this behavior, check the box under Device Status. Select Finish to complete the install.

Serial Port

Creating a serial port for a VM affords you the ability not only to have access to the host's physical serial port but also to have the ability to output information either to a file or to a named pipe. You'll have three options: Use Physical Serial Port on the Host, Output to File, and Output to Named Pipe.

If you elect to create a virtual port mapping to the host's physical port, you'll need to select which COM port you want to connect to using a drop-down menu. If you decide to send the information of a virtual COM port to a file, you'll need to specify the filename and storage location. The named pipes option, as shown in Figure 4-15, asks you to further configure the relationship of the named pipe. The named pipes setting allows you to directly connect two VMs; in addition, it allows you to create a connection between an application on the host VM and the guest VM. It's important to remember that the pipe name be identical on the client and server and takes the form of `\\.\pipe\<pipe name>`. After making your choices, select Finish to complete the install.

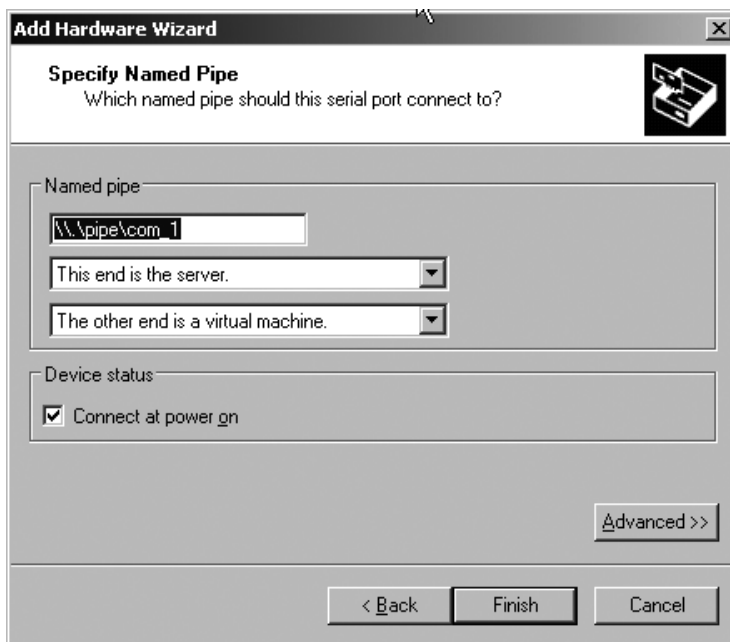


Figure 4-15. *Serial port named pipe configuration*

Parallel Port

Adding a parallel port to a guest VM is similar to adding a serial port. You'll need to select whether you want to use the host's physical port or use VMware's ability to write to a file. If you have multiple parallel ports, the wizard will allow you to specify which port you want to use. If the new virtual parallel port is to write to a file, you'll need to select the appropriate file output destination.

Generic SCSI Device

As shown in Figure 4-16, when adding a generic SCSI device to a guest VM, you'll need to select the physical SCSI device to which you want to connect; additionally, you'll also have to assign a virtual device node to use with the generic SCSI device. You can have the guest VM connect to the physical device when powering on. To do this, select the Connect at Power On option under Device Status.

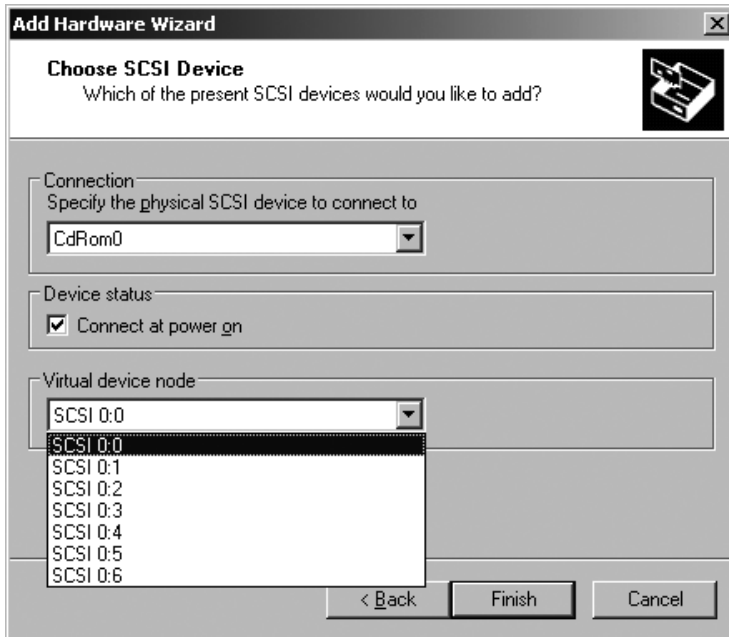


Figure 4-16. *Generic SCSI device configuration*

Note When editing a VM to add or remove virtual devices, be sure to click the OK button on the Virtual Machine Settings dialog box. If you don't, your settings will often not take effect.

Microsoft Virtual PC: Building a Windows VM

Building VMs with Virtual PC is wizard-driven. Select **File ► New Virtual Machine Wizard** to begin creating a guest VM. After acknowledging the splash screen, you have three options to choose from on the Options window:

- Create Virtual Machine
- Use Default Settings to Create a Virtual Machine
- Add an Existing Virtual Machine

The options are relatively self-explanatory and have no real gotchas associated with them. If you choose to use the default setting to create a guest VM, you'll have to create a virtual hard disk after the VM creation process has completed. Looking at Figure 4-17, notice that a default machine has no disk configured for it.

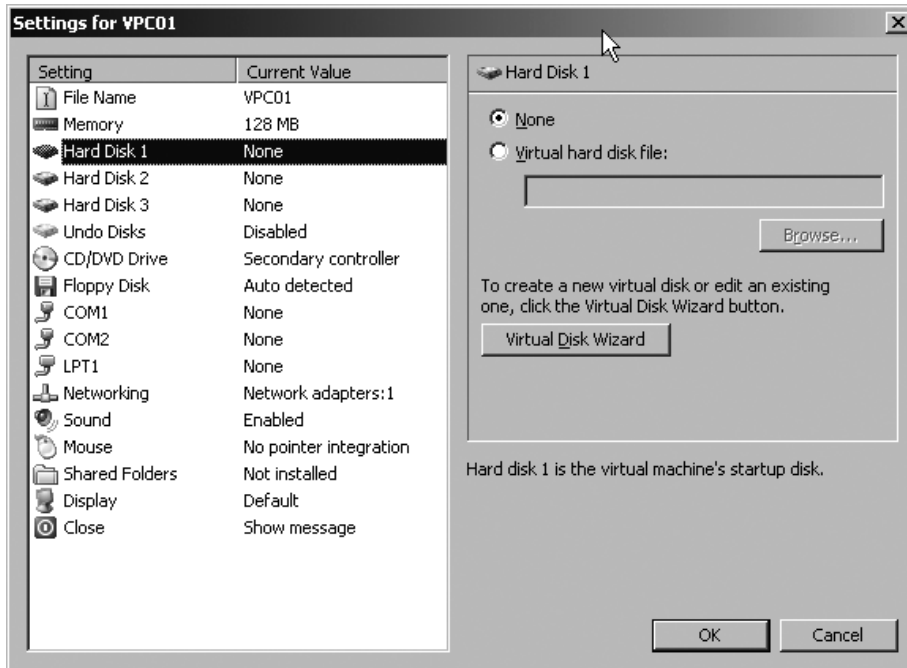


Figure 4-17. Default VM configuration for Virtual PC VM

During the creation process, at the Virtual Machine Name and Location prompt, you'll be asked to supply a name for the VM and a location to store its files. You should create a separate directory for every VM and name the directory after the VM is created. By keeping all the files labeled with the VM's name and in a separate directory, it's easier to manage guests later.

On the Operating System screen, shown in Figure 4-18, simply select the guest operating system you want to install from the drop-down list. All listed systems will be Microsoft branded. If you're installing an OS not officially supported by Microsoft, select Other.

The Memory selection window, shown in Figure 4-19, offers a recommended amount of RAM based on the OS chosen. Select Adjust to change your RAM requirements as necessary. Make sure you adhere to best practices when creating production VMs. In lab environments, you can slide by on minimum memory requirements for the guest OS.



Figure 4-18. OS selection

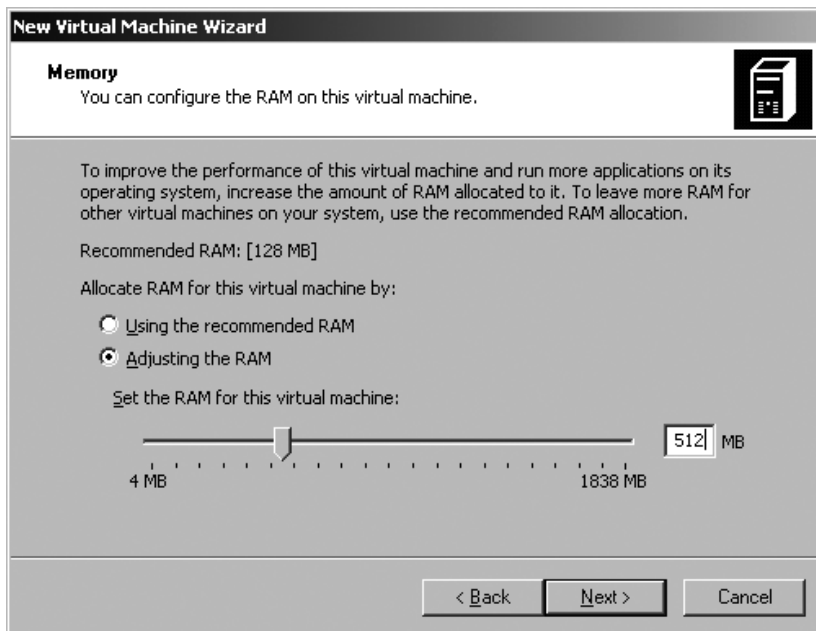


Figure 4-19. VM memory selection

The Virtual Hard Disk Options screen and Virtual Hard Disk Location screen (shown in Figure 4-20) allow you to choose between using an existing virtual hard disk and creating a new one. If you're restoring a VM or using a copy of an existing disk, choose Existing; otherwise, select New Virtual Hard Disk. When it comes time to specify the storage point of the VM, be sure to create a unique directory named after the guest VM and name the virtual hard disk after the VM. Having all the files of a VM in one location makes administration significantly easier.

As shown in Figure 4-21, the wizard will finish the process by presenting you with a summary window. Verify that all selections were registered correctly before completing the process. Your VM should be viewable in the console; you can further configure the VM's virtual hardware by selecting Settings.

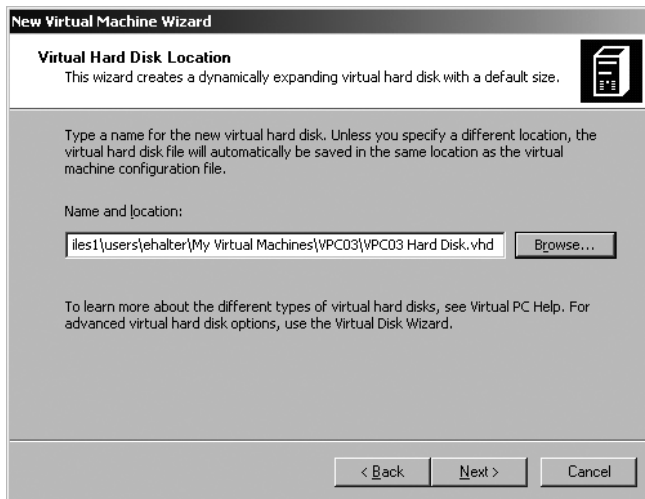


Figure 4-20. *Virtual hard disk location*



Figure 4-21. *Completing the wizard*

Microsoft Virtual PC: Building a Linux VM

We realize that Microsoft doesn't officially support Linux operating systems as guest VMs on Virtual PC, but you may find yourself in a test or an educational environment with the need to have Linux running on Virtual PC. Virtual PC has the wherewithal to run many operating systems: it's just that many operating systems are out of the scope of Microsoft's support services. If you find you want to run unsupported operating systems, just remember you'll be the first line of support, and the Internet will be your next best friend. More important, Microsoft dropped PC Additions for Linux operating systems after purchasing Virtual PC from Connectix; therefore, you'll have to do without the convenience factors and performance benefits it offered.

Just like building Windows VMs, building Linux VMs with Virtual PC is wizard-driven. Rather than repeating a bunch of figures from the previous section, we'll include only the ones important to Linux. To begin, select File ► New Virtual Machine Wizard. After acknowledging the splash screen, you have three options to choose from on the Options window:

- Create Virtual Machine
- Use Default Settings to Create a Virtual Machine
- Add an Existing Virtual Machine

The options for Virtual PC are relatively self-explanatory. If you choose to use the default setting to create a guest VM, you'll have to create a virtual hard disk after the VM creation process has completed. During the creation process, at the Virtual Machine Name and Location prompt, you'll be asked to supply a name for the VM and a location to store its files. You should create a separate directory for every VM and name the directory after the VM being created. By keeping all the files labeled with the VM's name and in a separate directory, it's easier to manage guests later.

On the Operating System screen, select Other for the guest operating system from the drop-down list. The Memory selection window offers a recommended amount of RAM based on the OS chosen. Select Adjust to change your RAM requirements as necessary. Make sure you adhere to best practices when creating production VMs. In lab environments, you can squeak by on minimum memory requirements for the guest OS.

The Virtual Hard Disk Options and Location screens allow you to choose between using an existing virtual hard disk and creating a new one. If you're restoring a VM or using a copy of an existing disk, choose Existing; otherwise, select New Virtual Hard Disk. When it comes time to specify the storage point of the VM, be sure to create a unique directory named after the guest VM and name the virtual hard disk after the VM. Having all the files of a VM in one location makes administration significantly easier. The wizard will finish the process by presenting you with a summary window. Verify that all selections were registered correctly before completing the process.

When you go to install the Linux operating system on the guest VM, you'll have to contend with several issues, including video, time synchronization, and sound. When it comes to the point in the install process to configure X Windows, be sure that the S3 Trio64 card is selected and the memory requirement is set to 8MB. The reason you want to set these values as mentioned is because these are the virtual devices made available to the guest VM. In addition, stick with 8-bit or 16-bit video. Many Linux distributions have difficulty running in 24-bit

mode. Because the time won't synchronize properly from the host system to the Linux guest VM, you'll have to rely on NTP; we'll cover using your host as a time source in Chapter 8. For installation details on NTP, you'll need to reference your distribution's documentation. However, if you're using recently released Red Hat products, you can configure NTP by selecting System ► Date/Time Properties. From the application window, select the Enable Network Time Protocol option, and enter the IP address of the server you want to use. If you don't require sound for your VM, you're best off not enabling it for the VM; however, if you want to configure it, you'll want to configure Linux to use the SoundBlaster SB16 sound card. In a terminal window, execute the `sndconfig` sound configuration utility. If the utility fails to detect the virtual sound card, you'll need to supply hardware configuration parameters to it during the manual configuration process on the Card Settings screen. Specify the following values:

- **I/O Port:** 0x220
- **IRQ:** 5
- **DMA 1:** 1
- **DMA 2:** 5
- **MPU I/O:** 0x330

Virtual PC Virtual Hardware Options

Virtual PC offers an extensive array of virtual hardware options that you can add to leverage the functionality of a VM or that you can remove to increase performance. Whatever your needs may be, you can access hardware options by selecting Settings from the application console. You can get a good idea of how each option is configured for a particular VM by checking its *.vmc file before and after any deletions or additions. If you have any question as to what a particular item in the Settings dialog box does, Microsoft provides a brief and informative explanation in the lower-right corner of the application window.

File Name

The File Name option provides you with the ability to rename the virtual machine. It doesn't modify the names of the original installation directory or the VM's virtual hard disk; however, it does rename the configuration file.

Memory

The Memory option allows you to increase or decrease the amount of available RAM by using a slide control or by entering the memory size. You can make adjustments in 1MB chunks.

Hard Disk (1–3)

You can enable or disable up to three virtual hard disks for any given VM. You have the option of specifying an existing virtual disk or creating one with the assistance of the Virtual Disk Wizard. If you're creating a disk with the wizard, be sure to adhere to good naming conventions.

Undo Disks

Undo Disks is an option that can be enabled; it allows you to commit or discard changes to a VM after you're done using it. You can delete, commit, or save the changes for the next time you use the VM. Undo disks are great for being able to preserve the state of a VM, but they do require additional disk space.

CD/DVD Drive

The CD/DVD Drive option connects to the host's physical drive and is mapped to the guest VM's secondary IDE controller by default. If you clear the option box, the virtual CD drive will be connected to the primary controller of the guest VM.

Floppy Disk

The Floppy Disk option is set by default to detect the host's physical floppy disk. If you don't need access to the physical device, disable it for added performance.

COM Ports

Creating a serial port for a VM affords you the ability to not only have access to the host's physical serial port but also to have the ability to output information either to a file or to a named pipe. If you elect to create a virtual port mapping to the host's physical port, you'll need to select which COM port you want to connect to using a drop-down menu. If you decide to send the information of a virtual COM port to a file, you'll need to specify the name of the file and the storage location, and the named pipes option requires you to specify the named pipe.

LPT1

LPT1 allows the VM to connect to any of the host's available physical parallel ports.

Networking

The Networking option contains the settings for the four available network adapters the guest VM can use. You can elect to use any or all the adapters and set the network connectivity type as well.

Sound

The Sound option is enabled by default. You can disable it for increased performance and to eliminate any potential conflicts with applications.

Mouse

The Mouse Pointer Integration option is available only after the installation of Virtual Machine Additions. Pointer integration is what allows the mouse to flow freely between the guest window and that of the host.

Shared Folders

After the installation of Virtual Machine Additions, the Shared Folders feature enables the guest to access data on the host. You can specify the location by selecting Share Folder.

Display

Like the Shared Folders and Mouse Pointer Integration options, the Display settings can be used only after the installation of Virtual Machine Additions. The settings in Display allow you to manipulate resizing characteristics of the guest VM's window, from full-screen to windowed. You can also set the visibility of the menu and status bars.

Close

The Close setting controls displaying messages when a guest VM is shut down, turned off, or placed in a save state. Messages can be useful for troubleshooting purposes, so you may want to leave them enabled.

Installing Virtual Machine Additions

Microsoft Virtual Machine Additions, available from <http://www.microsoft.com/downloads>, increases the performance and flexibility of guest VMs running on Virtual PC and can be used only after the guest OS is installed. Figure 4-22 shows the installation of Virtual Machine Additions. You can also programmatically install Virtual Machine Additions using an unattended mode with `setup.exe` and its command-line options.



Figure 4-22. *Virtual Machine Additions*

Table 4-1 lists the command-line options you can use during an unattended install when using the setup program. The command should be in this format: `setup.exe -s -v"/qn [Reboot=ReallySuppress]"`. In addition, you can install the software by activating the executable from within Windows Explorer and using the wizard.

Note If you move a VM created with Virtual PC to Virtual Server, be sure to reinstall Virtual Machine Additions. Reinstalling the software ensures that the moved guest VM is using the server version of Virtual Machine Additions and avoids any conflicts related to versioning.

The enhancements you can expect to experience from installing Virtual Machine Additions include seamless use of the mouse from the guest window to the host, increased performance, host-to-guest time synchronization, and optimized video drivers.

Table 4-1. *Virtual Machine Additions*

Option	Action Performed
-s	Runs setup program without a GUI
-v	Passes options to Msiexec.exe
/qn	Runs Msiexec.exe without a GUI
Reboot=ReallySuppress	Prevents reboot after installation

Managing VMs

Managing VMs in a production or test environment is no different from in traditional physical systems; you'll need to continue to adhere to the same best practices you probably already use, such as backing up, installing antivirus software, monitoring log files, renaming VMs, and performing routine maintenance. In the following sections, we'll cover backing up, copying, and moving VMs. In addition, we'll show you how to run VMs as services and begin to cover VM configuration files.

Backing Up and Modifying VM Configurations

Though we thoroughly cover backup strategies for VMs in Chapter 7, we'll briefly introduce you to backing up VMs and their configuration files. In addition, we'll cover modifying VM configuration files.

When you first begin experimenting with VMs, you'll soon realize that at some point it becomes necessary to rename a guest VM, its configuration file, and its virtual disk file. You'll generally find yourself in this position when you're forced to move a VM from a test environment to a production environment. To ensure the integrity of the VM and a smooth transition, you'll first need to back up your VM's virtual disk and configuration files, rename them as necessary, and then move them to the new host. Whether you're renaming a Microsoft Virtual PC or VMware Workstation VM, you can complete the procedure in a few steps.

Renaming VMware Workstation VM

Prepping VMware Workstation to accept a new name for a guest VM is fairly straightforward. The VM must be powered down and not in a suspended state before proceeding with the renaming procedure:

1. Remove the VM from the console inventory by selecting the Management tab of the VM to be removed. Right-click the tab, and select Remove from the favorites; this action won't delete any files related to the VM to be renamed. Close VMware Workstation.
2. Locate the VM directory containing the guest's VM configuration file and virtual hard disk file. If you used the default installation directory and named the installation directory after your VM, the path to your VM will be similar to `C:\Documents and Settings\user\My Documents\My Virtual Machines\<VM_Directory>`. Rename the directory, the *.vmdk file, and the *.vmc file to exhibit the new name.
3. Using a text editor, edit the *.vmx file to reflect the name change. You'll need to modify two lines: the virtual disk configuration line and the display name line. For example, a change may look similar to `scsi0:0.fileName = "<New_Name.vmdk">` and `displayName = "<New_Name>"`.
4. Start VMware Workstation, and open the renamed VM by selecting File ► Open Virtual Machine. You'll be prompted if you'd like to generate a new universally unique identifier (UUID). If you need to maintain the MAC address for your VM, select Keep the Existing Identifier; otherwise, select Create New Identifier.

Renaming Microsoft Virtual PC VM

Renaming a VM for Virtual PC follows approximately the same process as VMware VMs. The VM must be powered down and not in a suspended state, the virtual hard disk file and configuration file will need to be renamed, and edits to the configuration file must be made. Follow these steps:

1. Launch Virtual PC, and highlight the guest VM to be renamed. Select Remove from the Service Console to remove it from the VM list. Close Virtual PC.
2. Locate the VM directory containing the guest's VM configuration file and virtual hard disk file. If you used the default installation directory and named the installation directory after your VM, the path to your VM will be similar to `C:\Documents and Settings\user\My Documents\My Virtual Machines\<VM_Directory>`. Rename the directory, the *.vhd file, and the *.vmc file to exhibit the new name.
3. Using a text editor, edit the *.vmc file to reflect the name change. You'll need to modify three lines, including the virtual disk configuration lines and the display name line:

```
<absolute type="string">C:\Documents and Settings\ehalter\My Documents\
My Virtual Machines\New Virtual Machine\vpctest.vhd</absolute>
<relative type="string">.\vpctest.vhd</relative>
<name type="string">vpctest</name>
```

4. Launch Virtual PC, and select File ► New Virtual Machine Wizard from the menu. The wizard will prompt you through the process of adding your VM. Be sure to select Add an Existing Virtual Machine, and browse to the renamed directory. Select the VM's *.vnc configuration file.
5. Select Start from the console. Virtual PC may complain about the hardware standard being changed, but you can select not to see the notice again.
6. If the VM is a copy of an existing VM, be sure to delete the Ethernet MAC address value from `<ethernet_card_address type="bytes">MAC_ADDRESS</ethernet_card_address>` to avoid having duplicates on your network. On initialization, Virtual PC will create a new unique MAC.

Backing Up VMware Workstation

One method of backing up VMs is to power down the guest VM and the virtualization application. Then copy the entire VM to a new location. For VMware, if you need to recover a specific file, you can mount the copy in the new location and restore files using VMware's DiskMount Utility. If you want to revert to using the state of the VM at the time of the backup, simply copy it back to its running production location. This is a highly manual process, but this approach is fine for personal use or in instances where time permits VMs to be down for the copying process. Rather than firing up a copy of a VM to restore a file, you can restore an individual file with the DiskMount Utility. You can use this utility on Windows 2000, Windows XP, or Windows 2003 operating systems. VMware offers DiskMount as a free download from its Web site, and installing a VMware virtualization application isn't a prerequisite to installing and using it.

DiskMount is a command-line utility and can be accessed by running the command at the CLI from its installation directory. The default directory for DiskMount is `C:\Program Files\VMware\VMware DiskMount Utility`. If you want, you can change the path during the installation process. After downloading and installing the application, go to the CLI; you can take a moment to examine the utility's usage syntax by executing `vmware-mount /?`. You'll see a list of command options to mount a virtual disk, as shown in Table 4-2.

Table 4-2. *vmware-mount Command Options*

Utility Option	Action Performed
/d	Deletes virtual disk drive mapping
/f	Forcibly deletes virtual disk drive mapping
/v:N	Mounts a specific volume on a virtual disk
/p	Lists volumes or partitions located on a virtual disk
/y	Mounts a virtual disk even if it has a snapshot
/n	Doesn't mount a virtual disk if it has a snapshot
/?	Displays command help

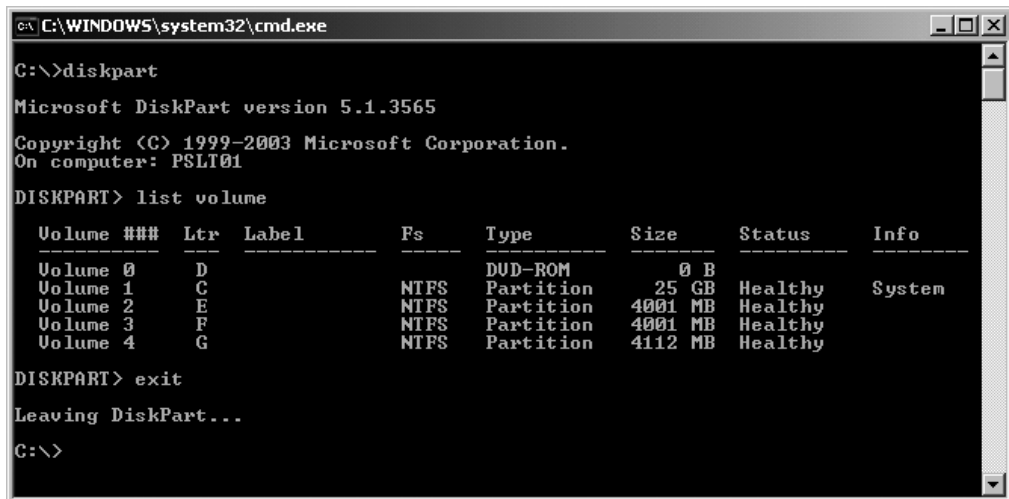
To mount a virtual disk, you'll need to execute the DiskMount Utility as follows:

```
vmware-mount [drive letter:] [path to virtual disk] [options]
```


For example, if you want to mount a disk labeled `VMTest1.vmdk` in the `C:\Documents and Settings\user\My Documents\My Virtual Machines\VMTest1` directory, you'll first need to choose an available drive letter to mount. You can use Windows Explorer or any available command-line utilities to see what drive letters are in use. For instance, if you're using Windows XP, and since you're already at the command line, you can use the `diskpart` utility.

Note If you need further assistance with the `diskpart` command, you can refer to Microsoft's resources Web site at http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/dm_drive_letter.mspx.

Assuming you're using Windows XP, type `diskpart`. This will start the `diskpart` utility and place you at its command prompt. Next, type `list volume`. The output will look similar to Figure 4-23. Terminate the `diskpart` utility by typing `exit`. This will put you back at the normal command prompt where you can continue mounting your virtual disk. Be sure to be in the utility's installation directory before continuing.



```

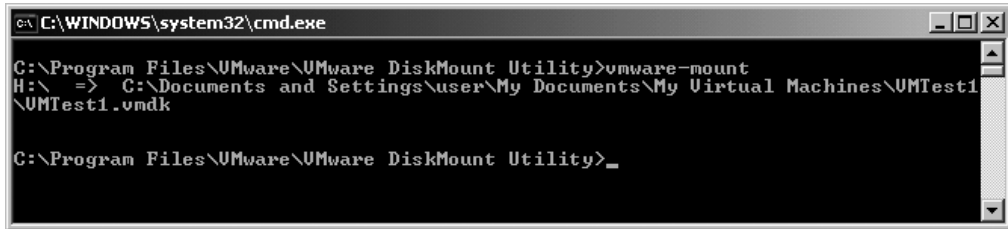
C:\WINDOWS\system32\cmd.exe
C:\>diskpart
Microsoft DiskPart version 5.1.3565
Copyright (C) 1999-2003 Microsoft Corporation.
On computer: PSLT01
DISKPART> list volume

Volume ###  Ltr  Label          Fs      Type          Size      Status       Info
-----
Volume 0      D             DUD-ROM        0 B
Volume 1      C             Partition      25 GB      Healthy      System
Volume 2      E             Partition     4001 MB    Healthy
Volume 3      F             Partition     4001 MB    Healthy
Volume 4      G             Partition     4112 MB    Healthy
DISKPART> exit
Leaving DiskPart...
C:\>

```

Figure 4-23. Windows XP `diskpart` utility output

For the `vmware-mount` utility to work properly, you'll need to select any drive letter not in use that's greater than D. In our example, the next available drive letter is H. With all the variables made known, you can execute the `vmware-mount` command. For our example, we'll need to execute `vmware-mount H: "C:\Documents and Settings\user\My Documents\My Virtual Machines\VMTest1\VMTest1.vmdk"`. Notice the use of double quotes, which help the operating system to correctly interpret spaces. If the command executes correctly, the command will return with nothing. Rather than assuming all went well, run the `vmware-mount` command from another instance of the command prompt. This will list all mounted virtual disks and will look similar to Figure 4-24.



```
C:\WINDOWS\system32\cmd.exe

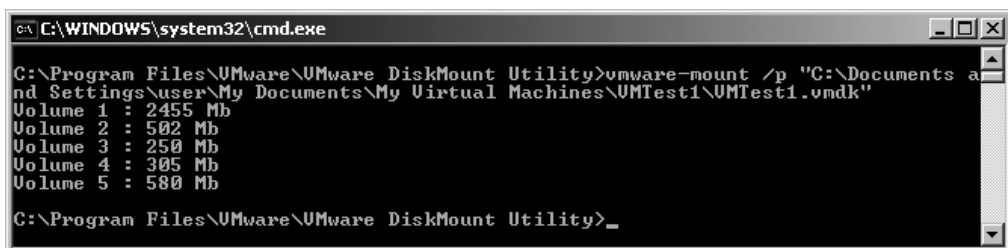
C:\Program Files\VMware\VMware DiskMount Utility>vmware-mount
H:\ => C:\Documents and Settings\user\My Documents\My Virtual Machines\UMTest1\
UMTest1.vmdk

C:\Program Files\VMware\VMware DiskMount Utility>_
```

Figure 4-24. *vmware-mount* output

To restore a file, you'll have to make sure you have the copied version of your VM disk mounted as well as the production VM's disk mounted. Remember that the VMs can't be powered on. Now, browse to the VMs mounted virtual disk, and find the source file you need to restore. Next, copy it over the existing file in the destination virtual disk. When you're done restoring your file(s), unmount both virtual disks with the /d option. For our example, we'd need to run `vmware-mount H: /d`. Sometimes the utility will fail to dismount the virtual disk. If this is the case, use the /f switch to forcibly dismount the disk.

Being able to mount a specific volume on a virtual disk comes in handy when the disk partitions are formatted differently. For instance, you could have incompatible partition types (UFS, EXT2, or EXT3) and FAT/NTFS partitions on a single virtual disk. Being that you can't mount certain types of partitions, you can single out the FAT or NTFS partitions. To mount a specific partition, you'll need to use the /v option with `vmware-mount`. For instance, if you wanted to mount the third partition on a virtual disk, you'd execute `vmware-mount /v:3 h: "C:\Documents and Settings\user\My Documents\My Virtual Machines\UMTest1\UMTest1.vmdk"`. But how do you find out what partitions are available? To determine partitions that are available on a virtual disk, you need to use the /p option with `vmware-mount`. The output will look similar to Figure 4-25.



```
C:\WINDOWS\system32\cmd.exe

C:\Program Files\VMware\VMware DiskMount Utility>vmware-mount /p "C:\Documents a
nd Settings\user\My Documents\My Virtual Machines\UMTest1\UMTest1.vmdk"
Volume 1 : 2455 Mb
Volume 2 : 502 Mb
Volume 3 : 250 Mb
Volume 4 : 305 Mb
Volume 5 : 580 Mb

C:\Program Files\VMware\VMware DiskMount Utility>_
```

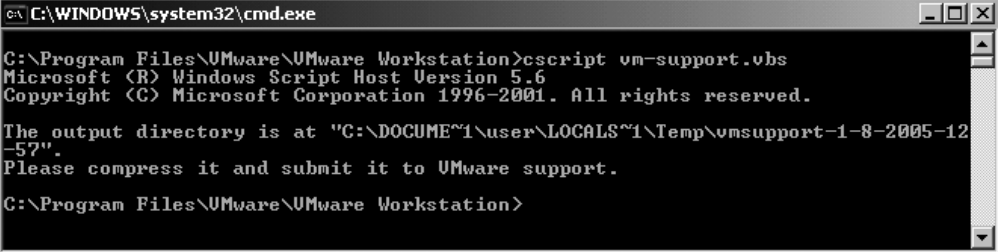
Figure 4-25. *vmware-mount /p* output

The DiskMount Utility is limited to mounting FAT or NTFS volumes, and if any virtual disks are set to read-only or if they're compressed, you'll have to change the disk's attributes to successfully use the DiskMount Utility.

Note VMware has a DiskMount Utility guide available on its Web site. If you need further assistance, you can download it at <http://www.vmware.com/download>.

Another “utility” you can use from VMware to aid in backing up VMware configuration files is the `vm-support` script. You can find it in the VMware Workstation installation directory, and it's titled `vm-support.vbs`.

To run the script, use the command line, and browse to the install directory of Workstation. Then execute the `cscript vm-support.vbs` command. The script can take quite a while to execute, so don't blow it by closing the window because you think the system has locked up. When the script has completed, it will display the directory location of its output. In Figure 4-26, the output is stored in the currently logged on user's Temp directory.



```
C:\WINDOWS\system32\cmd.exe

C:\Program Files\VMware\VMware Workstation>cscript vm-support.vbs
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

The output directory is at "C:\DOCUME~1\user\LOCALS~1\Temp\vm-support-1-8-2005-12-57".
Please compress it and submit it to VMware support.

C:\Program Files\VMware\VMware Workstation>
```

Figure 4-26. *vm-support.vbs* script output directory

Browse to the output directory, and take a look at the cache of files `vm-support.vbs` created. You should have several files:

- Current_User
- Global_Config
- Misc
- SYSTEMP
- TEMP
- VM

Treasure troves of information lay in wait in each directory. Take a moment to browse through the files and get acquainted with what's available to you. You'll find that there's everything from installation log files, installation directory settings, guest VM IP and MAC address information, power-on log data, and user preferences files. In our particular example, because we're interested in creating a backup of the VM's configuration file, we want to point out that it's found in the VM directory, `C:\Documents and Settings\user\Local Settings\Temp\vm-support-1-8-2005-12-57\VM\VM0`. After locating the output of the service script, gather the files into a single zipped file and store it on removable media.

Though the two techniques explored are fairly hands-on and not exceptionally practical, they're useful for individual users and workgroups. You may not use `vm-support.vbs` and `vmware-mount` to back up your virtual machines, but you now have two exceptionally cool and powerful tools to toss into your bag of tricks!

VMware *.vmx Configuration Files

In general, changing VMware configuration files isn't necessary. If you need to make changes, be careful of typos and accidental deletions. A VM can be rendered worthless from a fat-fingered configuration file. Manually modifying configuration files requires using a text editor. In Windows, Notepad is sufficient for this purpose.

Every virtual machine has a *.vmx configuration file that contains the settings for the VM, and you can locate the file with the aid of the Virtual Machine Editor. The Options tab displays the location of the VM's configuration file. Use Windows Explorer to find it and then open it for editing. As shown in Figure 4-27, you'll find that the configuration file looks similar to a traditional ini file.

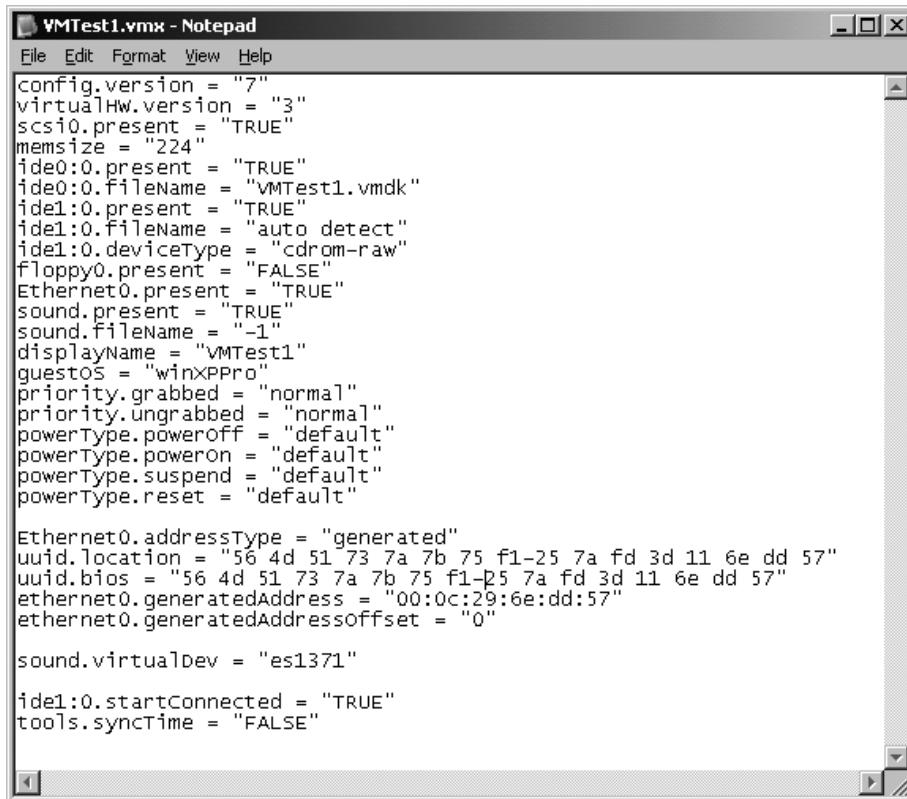


Figure 4-27. VMware *.vmx configuration file

Advantages to being able to add virtual components to a VM using an editor is that it's fast, and you can script changes for a large-scale rollout. We'll touch on some of the common options found in a VM configuration file.

Note You can add many undocumented options to a configuration file. Moreover, if you have a problem with a VM not working, you probably aren't the first person to have that problem. Visit VMware's forums for research and assistance. The Web site is at <http://www.vmware.com/community/index.jspa>. We'll cover the configuration files in more detail in Chapter 6.

You can control most virtual device behavior by changing its value from false to true, or vice versa. For instance, the configuration file will list devices as not being present by setting its status to false. Look at the following snippet from a VM's configuration file where no devices are present:

```
scsi0.present = "TRUE"
scsi0.virtualDev = "lsilogic"
scsi0:0.present = "FALSE"
scsi0:0.fileName = "NoDevices.vmdk"
ide1:0.present = "FALSE"
ide1:0.fileName = "auto detect"
ide1:0.deviceType = "cdrom-raw"
floppy0.present = "FALSE"
usb.present = "FALSE"
```

Referring to the lines beginning with `scsi`, the VM won't have a virtual disk made available to it on boot because the `scsi0:0.present` line is set to `FALSE`. You can determine from the `scsi0.present`, `scsi0.virtualDev`, and `scsi0:0.fileName` lines that the VM was originally configured with a SCSI virtual disk using the LSI SCSI controller and was labeled `NoDevices`. The lines beginning with `ide` tell you that no IDE devices are available to the VM because the `present` setting is `FALSE`. If you want to enable the CD-ROM virtual device, change the setting to `TRUE`. Moreover, if you want to enable the floppy drive or USB device, change their settings to `TRUE`. As you can see, there isn't much to enabling or disabling a device.

In the code that follows, let's look at a few more things in a typical configuration file:

```
memsize = "224"
ide0:0.present = "TRUE"
ide0:0.fileName = "VMTest1.vmdk"
ide1:0.present = "TRUE"
ide1:0.fileName = "auto detect"
ide1:0.deviceType = "cdrom-raw"
```

The `memsize` entry sets the memory size for a VM. If you want to increase the virtual RAM, simply change the line to reflect your needs. The `ide` entries detail the specifications for virtual IDE devices. Notice the two numbers separated by a colon after the `ide` entry. The first number is the virtual controller ID, and the second number is the virtual device ID. You know that an IDE channel can have two devices, 0 and 1. You also know that motherboards generally come with two IDE channels, 0 and 1. Therefore, you can determine that the virtual disk labeled `VMTest1.vmdk` is on the master device on the primary controller, and the CD-ROM device is the master device on the secondary controller. If you had a second CD-ROM on the secondary controller, the `ide` entry would be `ide1:1`.

Like virtual IDE devices, virtual SCSI devices are treated similarly with regard to syntax. Looking at the following example, you can determine that the VM using this configuration has the SCSI disk enabled because the first line is set to TRUE and it's using the LSI Logic driver (line 2):

```
scsi0.present = "TRUE"
scsi0.virtualDev = "lsilogic"
scsi0:0.present = "TRUE"
scsi0:0.fileName = "Windows Server 2003 Standard Edition.vmdk"
```

The third line tells you that the hard disk is configured as the first hard drive (0) on the first SCSI device (0) and is made available to the VM. The fourth line informs you that the default VMware nomenclature was used to label the disk field (*.vmdk).

The last thing we want to touch on in this section is the Ethernet entries in the configuration file. In the following *.vmx code, the two Ethernet adapters are configured as present for the VM with the TRUE setting in the first and fifth lines:

```
ethernet0.present = "TRUE"
Ethernet0.connectionType = "bridged"
ethernet0.generatedAddress = "00:0c:29:6e:dd:57"
ethernet0.generatedAddressOffset = "0"
Ethernet1.present = "TRUE"
Ethernet1.connectionType = "bridged"
ethernet1.generatedAddress = "00:0c:29:6e:dd:61"
ethernet1.generatedAddressOffset = "10"
uuid.location = "56 4d 51 73 7a 7b 75 f1-25 7a fd 3d 11 6e dd 57"
uuid.bios = "56 4d 51 73 7a 7b 75 f1-25 7a fd 3d 11 6e dd 57"
```

The second and sixth lines tell you that virtual Ethernet adapters are bridged to a physical NIC. The third and seventh lines list the 48-bit MAC address for each virtual NIC in the guest VM. Lastly, the ethernet0.generatedAddressOffset lines are used in calculating the UUID for the VM, which generates the MAC address. If a MAC address is generated and a duplicate address is found to exist by VMware, the offset value is added. VMware will continue to check for duplicate addresses and continue to add the offset until a unique MAC address is derived. The MAC addresses are represented in hexadecimal. To generate the next available MAC address, the offset number (represented in decimal) is added to the last byte of the MAC address. In the example, 57 in hexadecimal is equal to 87 in decimal. Add the offset of 10 to 87 to arrive at the next value for the last byte of the second MAC address, 97 decimal or 61 hexadecimal.

Virtual PC *.vmc Configuration Files

Virtual PC's configuration file is stored in a text-editable XML format. It's used to store a guest VM's configuration information and can be manually edited. Before editing the file, be sure to make a backup of the file in case you enter errors during the editing process. Figure 4-28 depicts the layout of the file, which is difficult to edit using Notepad. You may want to use a color-sensitive editor, such as Crimson Editor (<http://www.crimsoneditor.com>).

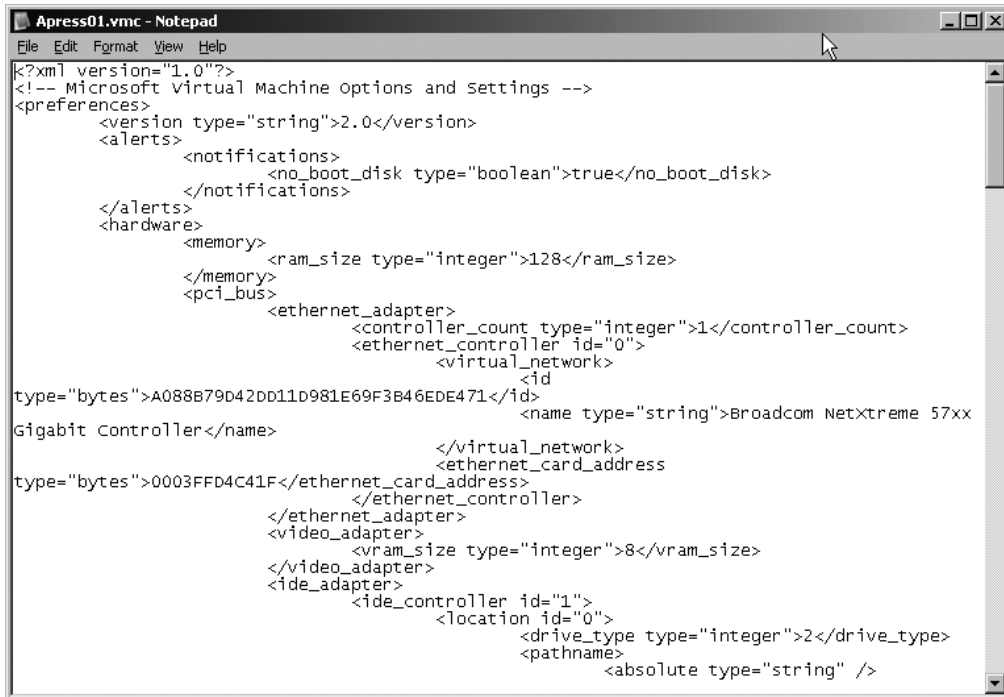


Figure 4-28. Virtual PC *.vmc configuration file

Virtual PC's configuration file is cryptic and can be difficult to edit. However, don't let this deter you from experimenting or making necessary changes. In the configuration file, you'll want to change the VM's MAC address to avoid duplicates on your network. Unlike VMware's use of the UUID to help manage MAC addresses, Virtual PC will passively "allow" duplicates to occur if you're making a copy of a VM to run concurrently with the parent image. If the VM is just being moved, it's not necessary to change the MAC.

To change the MAC address, find the section of the configuration file containing the text string `<ethernet_card_address type="bytes">0003FFD6C41F</ethernet_card_address>`. Remove the 48-bit address from the string, and save the file. In this example, the MAC address is 0003FFD6C41F. On the VM's next boot, Virtual PC will see that the MAC address has been removed, and it will generate a new MAC for the VM.

Other entries in the *.vmc configuration file you may want to experiment with are the settings to change the name and location of the *.vhd configuration file (ide_controller), screen size and resolution (video), RAM allocation (memory), and mouse behavior (mouse). Because the file is a text representation of the Virtual Machine Options and Settings dialog box, you can programmatically add or remove devices on dozens of machines with little scripting effort.

Note If you want to add virtual devices to a guest VM and are unsure of the settings in the *.vmc file, create a VM and remove all the devices. Next, add the device in question, and look at the changes made to the *.vmc file.

Copying and Moving VMware Workstation Guest VMs

Before you begin, make sure you have a good backup of the VM you intend to move to a different host. If something blows out in the process, you'll want to be able to quickly revert to your existing configuration. After confirming you have a good backup, ensure your user account has sufficient privileges to complete all tasks. Next, you'll need to complete a few preliminary tasks before moving the guest VM:

1. Power down the guest VM to be moved. Make sure it's *not* in a suspended state: a suspended VM is a recipe for disaster.
2. Close the VMware Workstation virtualization application.
3. Ensure a transfer medium exists for moving the guest's files. Assuming your VM can fit on a CD or DVD, you can use this type of medium, or you can use a mutually accessible network share: copy the files to the network share and then move them to their final destination. If the guest is going to be placed on a new network host, the easiest way to move the files is over your existing network, going point to point. To connect to the host VM system, you can use the `net use` command to access to the hidden system share to copy the files from an unshared directory. For example, to connect to a server called NEWVMHOST using the default VMware Workstation VM directory, you'll want to run the following at the CLI:

```
net use * \\NEWVMHOST\C$\Documents and Settings\<username>\My Documents\My Virtual Machines /user:local\administrator <password>
```

replacing the following:

- *: Use the next available drive letter.
- <username>: Insert the user account profile ID.
- local: Use the local account IDs for authentication (you could use a domain name instead).
- administrator: Use the local administrator account to log in.
- <password>: Supply the local administrator account password.

With the preliminary steps checked off your task list, you can safely move the guest VM by performing the following steps:

1. Ensure that the new host system has sufficient resources for the added load of the VM to be moved. Next, make sure you have VMware Workstation correctly licensed and configured on the new destination host machine and not running.
2. Create a directory on the new host to house the guest VM files. (If you're just moving the location of the guest VM to a different directory on the current host, simply create the new directory in your desired location and locate the VM's files within the new directory.) If you're using the default directory of `C:\Documents and Settings\user\My Documents\My Virtual Machines`, simply create a subdirectory with the same name as the guest VM; for instance, use `C:\Documents and Settings\user\My Documents\My Virtual Machines\VMXPTEST`. Remember that you want to create directories with

meaningful names for your VMs and then store all related configuration information in said directory.

3. Make sure the guest VM is powered down. Next, you'll need to find *all* the files on the current host and copy them to the new host in the directory you just created. Files you'll minimally be concerned with are as follows:
 - The configuration file, *.vmx
 - The virtual disk files, *.vmdk
4. Start VMware Workstation, select File ► Open, browse to the VM's configuration file (*.vmx) in the directory you created, and then select Open. Take a moment to check the Virtual Machine Setting dialog box to verify that *each* device is properly configured and is pointing to the correct file location.
5. Next, check to make sure the VM name, configuration file, and redo log file are all correctly set. You can find this on the Options tab of the Virtual Machine Settings dialog box, as shown in Figure 4-29.

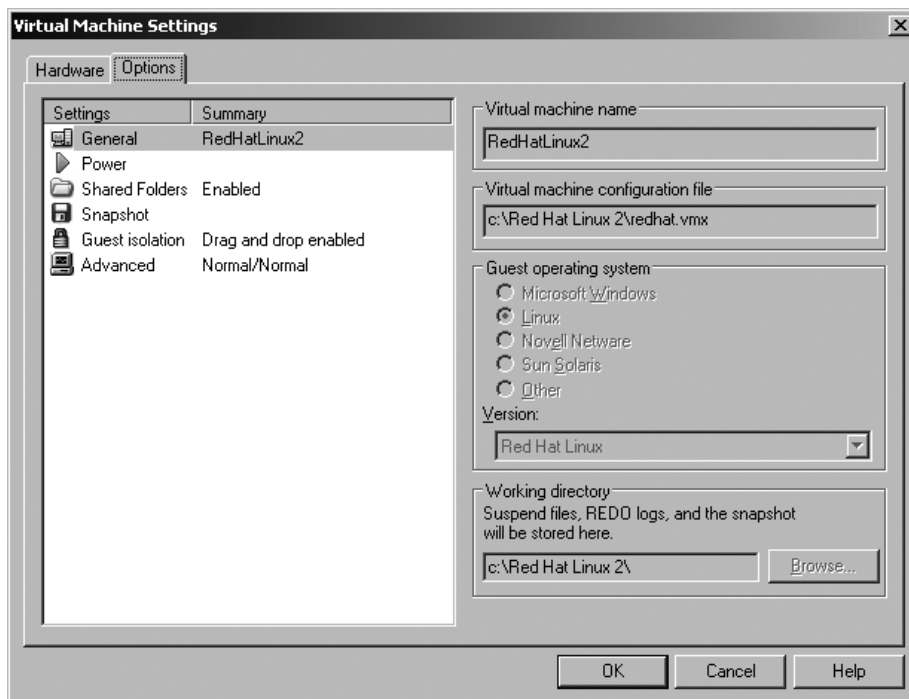


Figure 4-29. VMware Workstation's Virtual Machine Settings dialog box

6. Now, select Power ► Power On. If your guest VM fails to properly initialize, make sure you copied every file from the source host, and then double-check that all settings are pointing to the correct location. If you need to rename your VM, refer to the section "Managing VMs" earlier in this chapter.

VMware Universally Unique Identifiers

When you move a VM, you'll want to concern yourself with the system's UUID. VMware uses the UUID to generate a MAC address for a VM. The UUID is based on the host's System Management BIOS (SMBIOS) UUID and on the installation of the VMware application; it's specifically hooked to the *.vmx file's home directory location the first time the guest VM is booted.

Note SMBIOS presents motherboard management and diagnostic information to a computer system and its software. You can read more about the SMBIOS specification at <http://www.dmtf.org/standards/smbios>.

If you move the location of the guest, VMware will know that the initial install directory has changed because on each boot it compares the value of the UUID.LOCATION entry in the configuration file to the value of the UUID.BIOS. If the directory has changed, the UUID changes, which in turn changes the system's MAC address. Typically, this may not be a problem, but if you're using route maps based on MAC addresses, you may soon find that communications cease. In addition, if you use software that uses SMBIOS (UUID.LOCATION) to identify servers, communications will get interrupted. Lastly, if you're issuing IP addresses based on MAC addresses, your guest VMs will not be able to communicate if the UUID changes. You can find the UUID settings for any given VM in its configuration file. It's generally nested within the Ethernet settings. The following bit of code is from a VM's configuration file. Notice that each 128-bit value matches for uuid.location and uuid.bios.

```
Ethernet0.addressType = "generated"
uuid.location = "56 4d 51 73 7a 7b 75 f1-25 7a fd 3d 11 6e dd 57"
uuid.bios = "56 4d 51 73 7a 7b 75 f1-25 7a fd 3d 11 6e dd 57"
ethernet0.generatedAddress = "00:0c:29:6e:dd:57"
ethernet0.generatedAddressOffset = "0"
```

Foreseeing this issue, VMware configures its applications to ask you if you want to keep the UUID after a virtual machine has been moved. What's the right choice for you? The answer is, it depends. VMware suggests that if the VM is a copy, create a new UUID. Conversely, if the VM has moved to a new location, keep the existing UUID. You just want to make sure no duplicate MAC addresses appear on your network, because hosts identify each other via MAC address. If duplicates exist, network errors will occur. If you want to permanently change the behavior of the UUID warning, you have the option of always keeping or replacing the UUID each time the VM moves.

Copying and Moving Virtual PC VMs to Other Hosts

Microsoft recommends two methods for backing up its VMs. The first method is to treat the VM like a typical physical server or workstation to be backed up. For instance, if you're using QiNetix from CommVault or Backup Exec from Veritas in your backup environment, load the backup agent on the VM and include it in your existing backup process. The advantage to using the first approach is that you won't have to create any new administrative processes to accommodate new guest VMs. The second method for backing up Microsoft Virtual PC VMs is

to back up virtual hard disk and configuration files of a powered-down VM. You'll be treating the VM as if it were a group of normal files, and you'll want to specifically back up the virtual hard disk image file (*.vhd), the configuration file (*.vmc), and the saved state file (*.vsv) if the VM is suspended. The advantage of the second approach is that the entire state of a VM can be restored by using a few files.

Note Be careful when backing up VMs in a suspended mode because the System State is saved to memory in two locations: partly in the *.vhd file and partly in the *.vmc file. Restoring a VM saved in suspend mode can and generally does create an unstable VM.

When it comes time to move a guest VM between hosts, power down the VM to be moved and locate two files: the virtual hard disk file (*.vhd) and the VM configuration file (*.vmc). The default location for Microsoft Virtual PC guest VMs and their associated files is C:\Documents and Settings\user\My Documents\My Virtual Machines\<Guest VM Name>. Before you begin, make sure you have a good backup of the VM to be moved. If the move process fails, you'll want to be able to rapidly revert to your existing guest VM configuration.

After creating and confirming you have a good backup, ensure that your user account has the necessary privileges to complete all tasks. Now, you'll need to complete a few preliminary tasks before moving the Virtual PC guest VM:

1. Power down the guest VM to be moved. Make sure that it's *not* paused: moving a paused VM is a recipe for disaster.
2. Close the Microsoft Virtual PC application.
3. Ensure that a transfer medium exists for moving the guest's files. Assuming your VM can fit on a CD or DVD, you can use this type of medium, or you can use a mutually accessible network share: copy the files to the network share, and then move them to their final destination. If the guest is going to be placed on a new network host, the easiest way to move the files is over your existing network, going point to point. To connect to the host VM system, you can use the `net use` command to access the hidden system share to copy the files from an unshared directory. For example, to connect to a server called NEWVMHOST using the default Microsoft Virtual PC directory, you'll want to run the following at the CLI:

```
net use * \\NEWVMHOST\C$\Documents and Settings\<username> ➤  
\\My Documents\My Virtual Machines /user:local\administrator <password>
```

replacing the following:

- *: Use the next available drive letter.
- <username>: This is the user account profile ID.
- local: Use the local account IDs for authentication (you could use a domain name instead).
- administrator: Use the local administrator account to log in.
- <password>: Supply the local administrator account password.

Having backed up the guest VM to be moved, you can safely move the guest VM by performing the following steps:

1. Make sure the new host system has sufficient resources for the added load of the VM to be moved. Next, make sure Microsoft Virtual PC is correctly licensed and configured on the new destination host machine and not powered on.
2. Create a directory on the destination host to receive the guest VM files. (If you're just moving the location of the guest VM to a different directory on the current host, create the new directory in the desired location and place the VM's files within the new directory.) If you're using the default directory of `C:\Documents and Settings\user\My Documents\My Virtual Machines`, create a subdirectory with the same name as the guest VM. When creating directories for VMs, use meaningful names to store all configuration files and information.
3. Ensure that the guest VM is powered down. Now, find *all* the files for the guest VM to be moved on the current host and copy them to the new host in the directory you just created. Files you'll minimally be concerned with are `*.vmc`, which is the configuration file, and `*.vhd`, which is the virtual disk file.
4. Start Microsoft Virtual PC, and select **File ► New Virtual Machine Wizard**. Complete the wizard by selecting **Add an Existing Virtual Machine** and browsing to the location of the moved VM's `*.vmc` file. After creating the guest VM instance on the new host, take a moment to check the VM's configuration by selecting **Settings** from within the Virtual PC console. Verify that *each* device is properly configured and virtual disks are pointing to the correct file location.
5. Start the guest VM by selecting **Start** from within the Virtual PC console. If your guest VM fails to properly initialize, make sure you copied every file from the source host, and then double-check all settings are pointing to the correct location.

Running VMs As Services

Configuring VMware Workstation and Microsoft Virtual PC guest VMs to run as services isn't natively supported. If you require a guest VM to run in a production environment as a service, you should "buck up" and purchase server-class products. However, if you're on a budget, we'll show you how to cobble together a sound solution with a few tools.

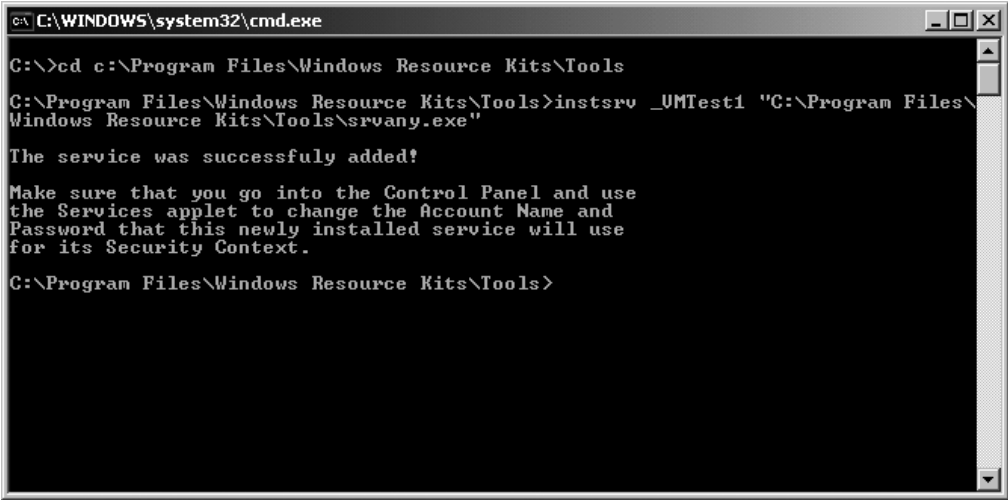
The tools you need to use are `srvany.exe` and `instsrv.exe`; you can find them in the Microsoft Windows 2003 Resource Kit. In particular, you can download the Windows 2003 Resource Kit tools from <http://www.microsoft.com/downloads>. Conveniently enough, the tools can be loaded on a typical Windows XP host; previous versions are even located on legacy resource kits. In addition to the Resource Kit tools, you need two more pieces of knowledge. Specifically, you need to know the location of the virtualization application executable, `vmware.exe` or `Virtual PC.exe`, and the location of the guest's configuration file, `*.vmx` for VMware or `*.vmc` for Virtual PC. Assuming you performed a vanilla install of your virtualization application, you can find the executable for VMware Workstation in the `C:\Program Files\VMware\VMware Workstation` directory and the `*.vmx` configuration file in the guest's VM subdirectory located in the `C:\Documents and Settings\user\My Documents\My Virtual`

Machines directory. For Virtual PC, the executable is located in C:\Program Files\Microsoft Virtual PC, and the *.vmc configuration file is located in the guest's VM subdirectory of C:\Documents and Settings\user\My Documents\My Virtual Machines.

After the tools are installed, you'll need to create the VM service. At the CLI in the directory where you installed the Resource Kit tools, C:\Program Files\Windows Resource Kits\Tools, execute the following commands, where *Service_Name* is what you want to call the VM service and *Srvany_Directory_Path* is the directory location of the *srvany* executable:

```
instsrv <Service_Name> <Srvany_Directory_Path>
```

Referring to Figure 4-30, you can see that we labeled our service *_VMTest1*. Beginning a service with an underscore will cause it to be listed first among all services. Listing your custom services first is a good way to keep track of any custom service you create. As a small aside, notice that we specified the absolute path of *srvany.exe* and placed the entire string in quotes; you'll need to do the same.



```
C:\WINDOWS\system32\cmd.exe

C:\>cd c:\Program Files\Windows Resource Kits\Tools
C:\Program Files\Windows Resource Kits\Tools>instsrv _VMTest1 "C:\Program Files\Windows Resource Kits\Tools\srvany.exe"

The service was successfully added!

Make sure that you go into the Control Panel and use the Services applet to change the Account Name and Password that this newly installed service will use for its Security Context.

C:\Program Files\Windows Resource Kits\Tools>
```

Figure 4-30. *Creating the VMware Workstation service*

You now need to take a moment to configure the newly created service so it can interact with the desktop. Use the Services icon in the Control Panel to complete the task, and use Figure 4-31 as an aid. Notice that the box you want to check is on the Log On tab.

You now need to do a bit of registry hacking. Locate the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<VM service name> subkey, and add a new key labeled Parameters.

Next, add the string value called Application to the new Parameters subkey, and enter the path to *vmware.exe* or *Virtual PC.exe* as its value, such as C:\Program Files\VMware\VMware Workstation\vmware.exe for VMware or C:\Program Files\Microsoft Virtual PC\Virtual PC.exe for Microsoft Virtual PC.

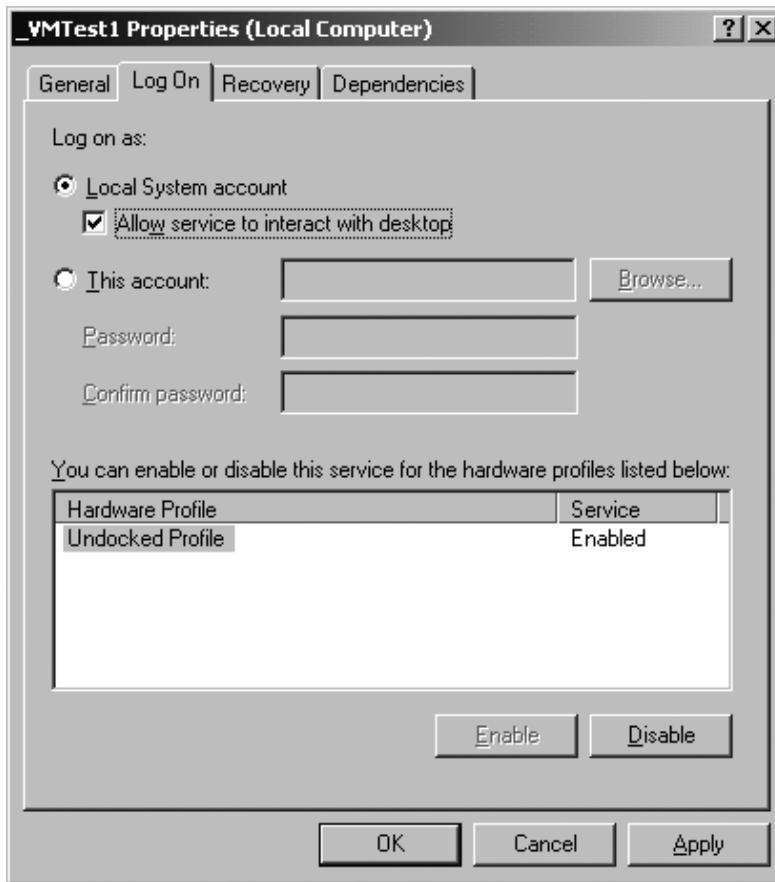


Figure 4-31. Enabling the _VMTest1 service to interact with the desktop

You'll next need to add the path to the VM's configuration file; for example, use C:\Documents and Settings\user\My Documents\My Virtual Machines\VMTest1\VMTest1.vmx for VMware Workstation or C:\Documents and Settings\user\My Documents\My Virtual Machines\VMTest\VMTest1.vmc for Microsoft Virtual PC. The entire syntax for the string should look like one of the following strings, including the quotes:

- **VMware Workstation:** "C:\Program Files\VMware\VMware Workstation\vmware.exe"-x "C:\Documents and Settings\user\My Documents\My Virtual Machines\VMTest1\VMTest1.vmx"
- **Microsoft Virtual PC:** "C:\Program Files\Microsoft Virtual PC\Virtual PC.exe " -pc VMTest1 -launch

Figure 4-32 details what your registry might look like for a VMware Workstation service after completing the editing tasks.

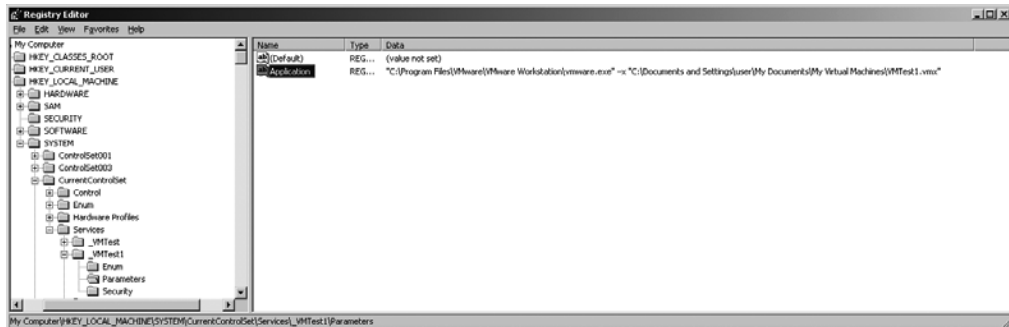


Figure 4-32. *VMware service registry editing*

You may find that these exact instructions won't work for your particular operating system. For VMware, you may get an error similar to Figure 4-33. This error is generally caused by extra spaces being passed to the `vmware.exe` command from the registry with the `-x` option. For Windows XP in particular, make sure there's no space between the double quotes after `vmware.exe` and the `-x`. If it fails, try it again with a space. Virtual PC can be picky about letter case, so type directory and VM names exactly how they appear.

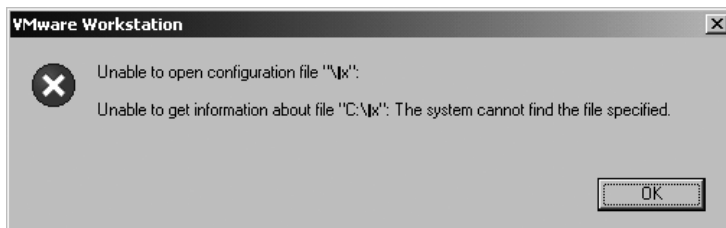


Figure 4-33. *VMware VM service error*

Note When editing the `Application` subkey's `Value` data string to make syntax changes, you may need to reboot between each registry edit to successfully commit your changes to your operating system.

If for some reason you can't get your VM service to function, it's okay. Some operating systems just have difficulty parsing command-line options, so we have a secondary approach to offer you. For VMware, if you have problems creating and running a VM as a service, run the `vm-support.vbs` script. Browse to the `SYSTEMP` directory. Check the log files for anything unusual. If you have typos in your registry settings, or if your OS is acting up, your log will have some details. In the following output, the log notes that VMware is having difficulty interpreting spaces:

```
MAR 07 09:54:28:
vmui| Log for VMware Workstation pid=228 version=4.5.2
build=build-8848 option=Release.4.5.2
MAR 07 09:54:28:
vmui| Unhandled VMUI exception: There is a space character
in your options. Perhaps you are trying to pass two separate options
(such as -q -x) in the first line of your configuration file.
If so, you need to merge them (-qx).
```

If your VM is having extensive issues trying to run as a service with the current approach, we have a workaround. You can create system shortcuts and use them with the `cmd` command. When using the shortcut method, create a folder to house the shortcuts so they don't litter up your desktop. Inside your VM service shortcut folder, create a shortcut by right-clicking your mouse and selecting **New ► Shortcut**. The Create Shortcut Wizard will appear. In the field provided, type the absolute path to your VMware Workstation or Microsoft Virtual PC executable, and select **Next**.

The wizard requires you to provide a name for the shortcut. For the name, simply use the name of the VM that's to be run as a service. Select **Finish** to complete the shortcut creation.

You'll now need to edit the shortcut to reflect that you want a VM to start when you activate the shortcut. Right-click the shortcut, and select **Properties**. In the **Target** field, as in Figure 4-34, you need to append the `-x` command switch and the location of the VM's *.vmx file. For Microsoft Virtual PC, you'll need to append the `-launch` switch and the location of the VM's *.vmc file.

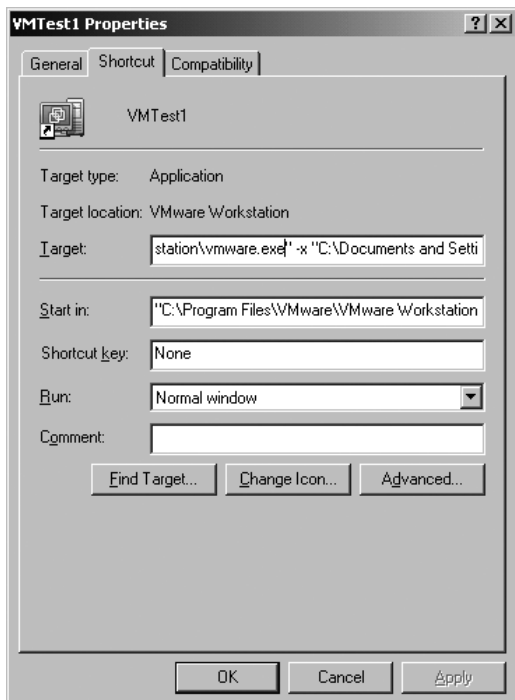


Figure 4-34. *Modifying the shortcut target*

In this example of creating a VMware Workstation service, the entire target line reads as follows (notice the spaces before and after the `-x` option). Select Finish when you're done, and take a moment to test the shortcut and make sure it launches your VM as you intended it.

```
"C:\Program Files\VMware\VMware Workstation\vmware.exe" -x➡  
"C:\Documents and Settings\user\My Documents\  
My Virtual Machines\VMTest1\VMTest1.vmx"
```

If you were creating a Virtual PC shortcut, the target line should look like the following:

```
"C:\Program Files\Microsoft Virtual PC\Virtual PC.exe" -pc VMTest1 -launch
```

To complete the process of creating a VM service using this second method, follow the previous steps. The only caveat is that you'll want the Value data for the Application string to point toward your shortcut. For instance, it should look similar to the following snippet:

```
cmd /c "C:\Documents and Settings\user\Desktop\Service_VMs\vmtest1.lnk"
```

The `cmd /c` command and option launches a command window and executes the shortcut. The `/c` option tells the command window to close upon execution completion. The `.lnk` part is the file extension for a shortcut. Be sure to append it to your command string, or your VM won't launch.

Whether you use the first or second option to create a VMware Workstation or Microsoft Virtual PC VM service, the guest VM will launch automatically in its own window on boot. In addition, the VM will continue to function even after you log off the host.

Note If you're just interested in having a VM power on when the host operating system boots, create a shortcut with the appropriate command options and place it in the Startup folder. These VMs will terminate when you log out of the host.

Things you'll want to watch out for when running guest VMs as services are any pop-up dialog boxes requiring your input, such as the VMware Tip of the Day. Assuming you're using best performance practices by not allowing VMs to automatically attach to floppy disk drives or CD-ROM drives, you won't have to worry about multiple VMs vying for device contention and the hassle of dealing with pop-up windows. All such dialog boxes will prevent your VMs from successfully running as unattended services. Lastly, if you can't get either of the previous approaches to work, take a moment to get help with the `cmd` command using `/?`.

Introducing VM CLI Administration and Keyboard Shortcuts

We'll begin to touch on using the `vmware.exe` and `Virtual PC.exe` commands in this section and cover some basic keyboard shortcuts. We'll cover other command-line utilities in Chapter 6, including `vmware-vdiskmanager`, `vnetstiffer`, and `vnetstats`. For Virtual PC, we'll discuss its `setup.exe` file.

VMware Workstation CLI

On Windows systems, to get an idea of the full range of options available with the `vmware.exe` command, execute `vmware.exe /?` at the CLI. If your host is based on Linux, execute `man vmware` for command help. If you're going to be using command-line switches on a regular basis, you may find that it's easier to use the options in a shortcut. For instance, on a Windows host, all you have to remember to do is add the option to the Target field and add the absolute path to the VM's configuration file. For example, on a Windows host, if you wanted to start a VM in a maximized window, the Target path should read similar to the following:

```
"C:\Program Files\VMware\VMware Workstation\vmware.exe" -X "
C:\Documents and Settings\user\My Documents\My Virtual Machines\VMTest1\VMTest1.vmx"
```

The `-X` option instructs the VMware Workstation executable to start the VM and maximize the application window. When you're modifying the Target field, be sure to enclose your text strings in double quotes. If you don't, your operating system may have difficulty parsing the spaces in filenames.

Let's take a moment to touch on each command option individually before moving onto keyboard shortcuts. For the most part, the Help window is self-explanatory.

- `-v` instructs `vmware.exe` to reveal the current VMware Workstation version and application specifications. You can get the same information by launching VMware and selecting Help ► About VMware Workstation.
- `-x` powers on a given virtual machine and launches the VMware application in a normal window.
- `-X` is similar to `-x`: the difference is that `-X` will not only power on a virtual machine but will also enter full-screen mode by taking over the desktop.
- `-q` closes a virtual machine upon power off. What this means is that when you launch a VM with the `-q` option, it will close the virtualization application when the guest VM is powered off.
- `-s NAME=VALUE` sets the given variable name equal to the given value.
- `-p` sets the correct relationship between the parent disk and the child disk by correctly pointing the child.

You can use multiple command-line options by concatenating them. On a Windows host, for example, if you want a VM to launch in full-screen mode and close the VM application on guest VM power down, you'll need to type something like this: `vmware.exe -Xq "C:\Documents and Settings\user\My Documents\My Virtual Machines\VMTest1\VMTest1.vmx"`.

If you fail to string the option together correctly, VMware will respond with a warning asking you to correct your code. Moreover, the command and its parameters are case sensitive in a Linux environment.

With the startup options explained, turn your attention to the keyboard. Like most applications, VMware has included keyboard shortcuts for your convenience. You'll find you can perform operations more quickly from the keyboard than by jockeying the mouse around. VMware Workstation controls the nature of the keyboard from the Hot Keys tab under

Edit ► Preferences. As depicted in Figure 4-35, you can change the hotkey combinations to your liking. If you have VMware tools installed, make sure your mouse doesn't place keyboard focus on the guest VM—don't hover the pointer over the guest VM's running window.

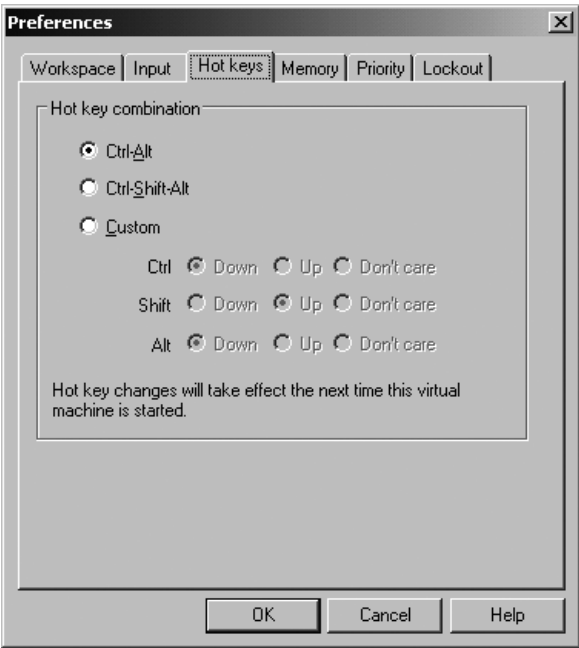


Figure 4-35. VMware hotkeys

Table 4-3 details several useful keyboard shortcuts to aid in the administration of VMware Workstation.

Table 4-3. VMware Keyboard Shortcuts

Keyboard Shortcut Keys	Action Performed
Ctrl+Alt	Releases mouse cursor from guest VM to host OS or returns guest to normal screen mode.
Ctrl+B	Powers on guest VM.
Ctrl+R	Resets guest VM.
Ctrl+Z	Suspends guest VM.
Ctrl+E	Powers off guest VM.
Ctrl+Alt+Enter	Allows guest VM to enter full-screen mode.
Ctrl+O	Opens a VM by locating its *.vmx configuration file.
Ctrl+F4	Closes currently visible VM.
Ctrl+Alt+Fx	Fx represents a function key to which a virtual machine is assigned. Using this key combination you can toggle between guest VMs running in full-screen mode. The full-screen mode title bar reveals the Fx key assigned to a particular guest VM.

Using the `vmware` command on a Linux host is similar to on a Windows host. Command-line usage takes this format: `vmware [-x] [-X] [-q] [-s <variable>=<value>] [-m] [-v] [/<path_to_config>/<config>.vmx] [X toolkit options]`. Short of adding the `-m` option, which starts VMware in quick switch mode, you'll have to contend only with the directory structure and command-line usage of Linux.

Virtual PC CLI Administration

Virtual PC offers a few executables that are useful for scripting batch files, installing Virtual PC using group policies, and running straight from the command line. We'll discuss two commands: the first command, `Setup.exe`, is valuable for installing Virtual PC, and the second command, `Virtual PC.exe`, is effective for launching VMs in a specific mode.

Note The CLI executable arguments are case sensitive, but the options for each aren't. As a rule of thumb, maintain the letter case of the commands, arguments, and options to avoid problems.

Let's first look at the syntax and options for a working example of `Setup.exe`. The trick to getting this command to do what you want is to strictly observe the spacing between the command and its available options and quotation marks. In the following example, notice that a space precedes the parameters `installdir` and `pidkey` and that all information following the `-v` parameter is enclosed in double quotes. Also, notice that there's no spacing between the other options and parameters. On some operating systems, you may find that you need a space after defining the product key if you use additional options:

```
Setup.exe -s -v"-qn pidkey=<productkey>allusers=2"
```

Now, let's look at the entire offering of `Setup.exe` using Table 4-4 and the complete command syntax before moving onto a couple more examples. The command and all possible options that can be used with it are `Setup.exe [-s] -v"[-qn] pidkey=<?>[allusers=2][username=?][organization=?][installdir=?]"`.

Table 4-4. *Setup.exe Command Options*

Option	Action Performed
<code>-s</code>	Silently installs without GUI interface.
<code>-v</code>	Passes parameters to <code>Msiexec.exe</code> .
<code>-qn</code>	Instructs <code>Msiexec.exe</code> to run without the GUI.
<code>pidkey=<?></code>	Supplies Microsoft Virtual PC product key code. Dashes aren't required.
<code>allusers=2</code>	Installs Virtual PC for all users. Exclude to install only for the current user.
<code>username=<?></code>	Supplies installation user account name. Exclude to use current user.
<code>organization=<?></code>	Supplies organization name. Exclude to use organization name from registry.
<code>installdir=<?></code>	Supplies custom installation directory. Exclude to use default.

The following example will install Virtual PC in unattended silent mode for all users with all possible customizations:

```
Setup.exe -s -v"-qn pidkey=12345123451234512345"
allusers=2username=administratororganization=Apress"
installdir=c:\apress\testnetwork"
```

You can use the next example to install Virtual PC with the GUI for the current user:

```
Setup.exe -v"-qn pidkey=12345123123456712345"
```

The second command we want to explore is `Virtual PC.exe`. The command is located in the installation directory and can not only be used to launch the Virtual PC application but can also set asset tag numbers, set BIOS serial numbers, set screen resolution, and set the launch methods of guest VMs. Before using the command, be sure to make a backup of the configuration file so you can compare the results before and after command execution. The gotchas that will keep you from successfully using this command are double quotes around the command; also, some options work on VM launch only. Before jumping into the raw command syntax, let's look at a couple of examples.

You probably have an idea of how the command works from the "Running VMs As Services" section. Here, we'll expand on the capabilities of `Setup.exe`. The example to follow starts a guest VM from the CLI titled `Apress01` without the Service Console:

```
"Virtual PC.exe" -pc Apress01 -singlepc -launch
```

In the next example, the Virtual PC VM, `Apress02`, is configured to launch in full-screen mode:

```
"Virtual PC.exe" -pc Apress02 -launch -fullscreen
```

After copying a VM to a new workstation, and you want it to appear in the list of available VMs in the console, you'll need to use the `-registervm` option:

```
"Virtual PC.exe" -registervm Apress03.vmc
```

Table 4-5 lists the available options for `Virtual PC.exe`, and the entire syntax with possible options follows:

```
"Virtual PC.exe" [-singlepc] [-quiet] [-usehostdiskcache] [-VMName
[-disableclose] [-disableopt] [-s3bitclip] [-setbiosno <?>] [-setassettag <?>]
[-launch] [-extnetworking] [-fullscreen|-window] [-geometry widthxheight{+|-}x
offset{+|-}y offset] [-minimize|-restore] [-pause|-resume]] [-help]
[-registervm <VM.vmc>] [-startvm <VM.vmc>]
```

Because the usage of some of these options may modify the contents of a VM's configuration file, be sure to back up the VM's configuration file first.

Table 4-5. *Virtual PC.exe Command Options*

Option	When Useable	Action Performed
-singlepc	Launch	Starts VM without the console
-quiet	Launch	Disables autostarting VMs
-usehostdiskcache	Launch	Enables host disk caching to improve disk performance
-pc virtual_machine_name	Launch or off	VM to be acted on or modified
-disableclose	Off	Deactivates the close button
-disableopt	Off	Disables optimizations
-s3bitclip	Launch	Clips S3 bit coordinates to 12
-setbiosno bios_serial_number	Off	Sets BIOS serial number
-setassettag asset_tag	Off	Sets chassis asset
-launch	Launch	Starts specified VM
-extnetworking	Launch	Restricts access to external network resources
-fullscreen	Running	Places windowed VM in full-screen mode
-window	Running	Places full-screen VM in window mode
-geometry widthxheight{+ -}➡ x offset{+ -}y offset	Running	Specifies location and size of VM window in relation to upper-left corner of monitor
-minimize	Running	Minimizes VM
-restore	Running	Places focus on background VM to now be in foreground
-pause	Running	Pauses VM
-resume	Running	Resumes paused VM
-help	Launch	Displays help
-registervm filename.vmc	Off	Registers VM with console
-startvm filename.vmc	Launch	Registers VM with console on

With the command-line executables explained, we'll now cover Virtual PC's use of keyboard shortcuts. You'll find you can perform operations more quickly from the keyboard. Table 4-6 details several shortcuts you can use in the administration of Virtual PC. If you don't want to use the right Alt key to perform the shortcuts, switch the hotkey by selecting File ► Options ► Keyboard from the Virtual PC Console. Highlight the Current Host Key box, and then press the hotkey you want to use.

Table 4-6. *Virtual PC Keyboard Shortcuts*

Keyboard Shortcut Keys	Action Performed
Rt. Alt+Del	Signals VM Ctrl+Alt+Del
Rt. Alt+P	Pauses or resumes VM
Rt. Alt+R	Resets VM
Rt. Alt+F4	Closes VM
Rt. Alt+Enter	Toggles full-screen/window mode
Rt. Alt+C	Copies selected item(s)
Rt. Alt+V	Pastes copied item(s)
Rt. Alt+I	Installs Virtual Machine Additions
Rt. Alt+down arrow	Minimizes VM
Rt. Alt+left arrow	Switches to previous windowed VM
Rt. Alt+right arrow	Switches to next windowed VM

Monitoring and Configuring VM Performance

You can keep guest VMs and their host computer systems optimized for performance and reliability through good configurations and the maintenance of both. In addition, maintenance includes routine monitoring. To get the most out of your systems, start with the foundation: you can tweak every line of code in a VM, and it will perform miserably if the host is underpowered and poorly managed. Moreover, you can pore over reams of management wallpaper, system logs, and network traces, and it will be for nothing if you don't start off with a host system configured with fast hard disks, powerful processors, and large quantities of RAM. If you want to review some no-nonsense approaches to building a good foundation for your VM, be sure to read Chapters 1 and 2.

Note A computer system that performs adequately as a non-VM host may quickly crash and burn when it's loaded up with VMs. Even if you have a cutting-edge processor and have maxed out the motherboard's RAM capacity, you'll need to contend with potentially more heat added by the increased CPU load. Make sure your system isn't suffering from inadequate cooling by checking heat sinks, fans, and ventilation.

To keep both the host and its guests running optimally, you can start with a few basics:

- Defragment virtual and physical hard drives. Fragmentation affects the general performance of virtual disks, snapshots, and the suspend/resume operations of VMs.
- Make sure your host operating system is optimized for performance and not convenience. For instance, Microsoft operating systems poll the CD-ROM drive frequently (about every second) to check for the insertion of media. This behavior isn't needed in a production environment. You can get additional ideas from Chapter 5.
- Ensure the guest VM has the correct setting for the guest operating system selection on the Options tab of the Configuration Editor. Having the correct operating system set ensures that the best possible running environment is employed for the VM. VMware optimizes internal configurations based on the choice you make for the guest.
- Using snapshots with VMs decreases performance. If you don't need the snapshot feature, disable it. From the application menu, select Snapshot ► Remove Snapshot.
- Verify the running mode of the VM. VMware provides two running modes: normal and debug. Debugging mode is significantly slower than the normal running mode. Check the setting of your guests by selecting VM ► Settings ► Options from the application menu. Select Advanced, and make sure Run with Debugging Information isn't selected under settings.

To help monitor your VMs, you can use Microsoft's Performance console. The Performance console is an application that consists of three management tools: System Monitor, Performance Logs and Alerts, and Task Manager. You can use data gathered from these tools to observe trends in workflow and resource utilization, test performance tweaks to the computer system, and troubleshoot system components. You'll look at VMware's performance counters and Virtual PC's performance options in the next sections. In Chapter 6, we'll cover how to use the Performance console with VMs.

VMware Performance Counters

VMware uses performance counters that can be managed through the host's System Monitor interface (formerly known as Performance Monitor). Unfortunately, the counters are available only for Microsoft hosts. You can, however, monitor the performance of a Linux VM. The performance counters included with VMware Workstation allow you to monitor memory usage, virtual hard disk access, and VM network traffic. To track the performance of a VM, it must be powered on. At the time of this writing, Microsoft Virtual PC doesn't include performance counters to specifically monitor its VMs. The VMware performance counters track the status of the virtual machine and *not* that of the guest OS.

To use the counters in Windows XP, launch the Performance console from the Control Panel. You can find it located in Administrative Tools, and it should look similar to Figure 4-36.

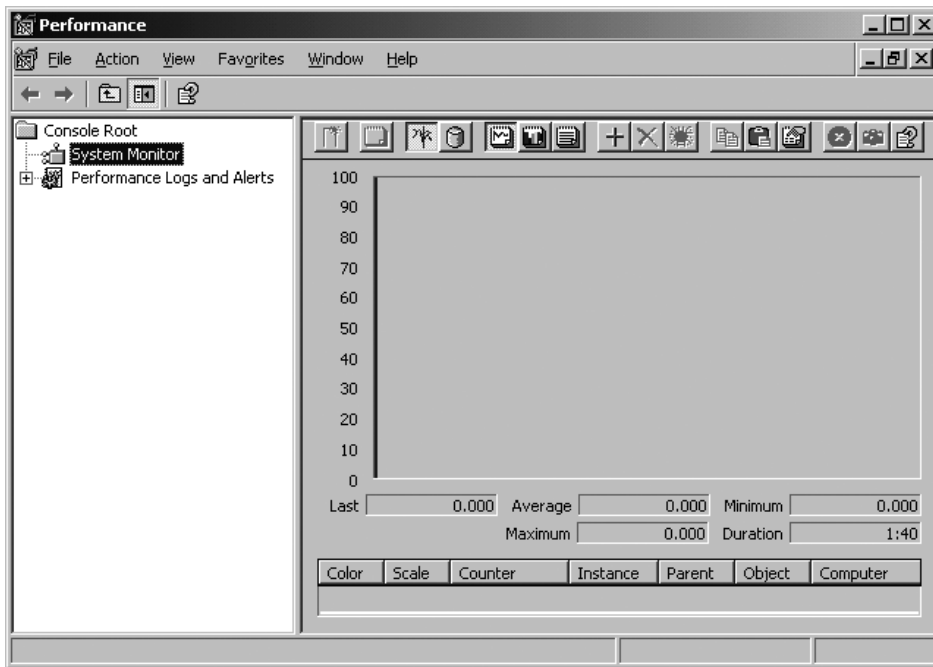


Figure 4-36. *Microsoft Performance console*

By default, it will open with three counters installed. To remove the counters, highlight each one, and press the Delete key on your keyboard. To add a counter, right-click in the gray graph display area, and select Add Counters. You can also add counters by selecting the plus button or by pressing Ctrl+I on the keyboard. The Add Counters dialog box will appear. You can monitor remote hosts as well as the local host. Connecting to hosts over a network can inflate certain counters and increase latency.

Though it's convenient to get an idea of what's going on with a particular system over a network, use counters locally if you're performing any type of baseline work. Make sure you've selected Use Local Computer Counters, and then select VMware from the Performance Object drop-down box. Now, to get an idea of what each counter reports, highlight it from the list, and then select Explain. Take a moment to go through each counter, so you'll know what's available to help you measure guest performance.

If you have several VMs running, the Performance console has the ability to monitor all VMs or a VM of your choice. If you want statistics from each running VM, select All Counters and All Instances. If you desire to monitor a specific counter on a specific VM, choose Select Counters from the list and then choose Select Instances from the list. When you use the Performance console, you really should be looking for something specific. If you add all counters from all VMs, you'll generally wind up with too much data and an unnecessary increase in overhead.

Virtual PC Performance Options

Microsoft Virtual PC has the ability to let you quickly tune performance through its console. To gain access to the Performance option, launch Virtual PC and select File ► Options ► Performance. As in Figure 4-37, you'll see two categories with a few selections to aid in managing performance.

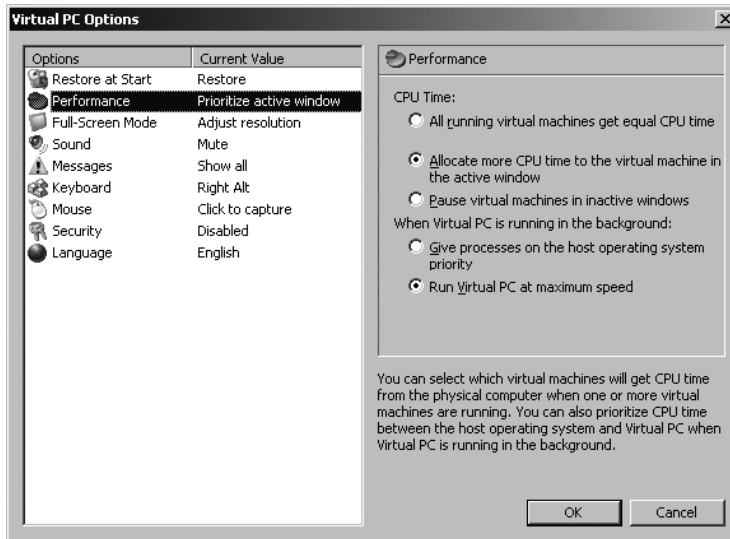


Figure 4-37. *Virtual PC performance options*

The two categories basically give you the ability to place your overall performance focus either on the host or on the guest VMs. When determining priority for the guests, you can favor the VM in the active window, distribute CPU resources equally, or pause VMs not in the active window. When Virtual PC is running in the background, you can choose to give running processes on the host priority or run Virtual PC at maximum speed all the time. Granted, the choices aren't real flexible, but they come in handy in an educational/test environment or on hosts with limited resources. Table 4-7 details each option.

Table 4-7. *Virtual PC Performance Options*

Option	Performance Action
All Running Virtual Machines Get Equal CPU Time	Equally distributes resources across running VMs
Allocate More CPU Time to the Virtual Machine in the Active Window	Distributes about 70 percent of resources to active VM
Pause Virtual Machines in Inactive Windows	Pauses all inactive window VMs, releasing CPU capacity to active VM
Give Processes on the Host Operating System Priority	Reduces background VM performance by 50 percent
Run Virtual PC at Maximum Speed	Allocates all CPU resources to all running VMs

Deciding when to use what option in which category is something you should consider when your needs change or when a system moves from a test environment to a production environment. The first category, CPU Time, grants you the ability to manage the host's CPU time. Minimal considerations for selecting a particular option are based on the following:

- If you're running several VMs in a production environment on Virtual PC, you may want to use the option to distribute resources equally. This ensures you won't experience excessive latency with nonactive window VMs.
- If you're in a lab or educational type environment, you'll probably want to use the 70/30 split option; choosing this option will ensure you're not waiting for the active window VM to catch up with your keystrokes and mouse movements.
- If you're running more than one VM, Microsoft suggests you use the pause VM option. This option is useful if you need to access only one VM at a time and need to allocate all possible CPU resources to the active window VM.

The second category, When Virtual CPU Is Running in the Background, affords you the opportunity to configure the background performance of Virtual PC, and minimal considerations for selecting a particular option are based on the following two choices:

- If you plan on using the host system and guest VMs concurrently, you should choose to give the host operating system priority.
- If you're going to use the host system as a platform for guests only, then you'll want to allow Virtual PC to run at maximum speed.

Summary

We covered the deployment and management of guest VMs in this chapter and discussed the available hardware options in Microsoft's and VMware's virtualization applications. You now know how to install, back up, monitor, and run guest virtual machines as services. With client-side virtualization products covered, you're now ready to move onto enterprise-class virtualization products. In Chapter 5, you'll investigate the three server virtualization applications: Microsoft Virtual Server, VMware GSX Server, and VMware ESX Server. We'll continue taking a step-by-step approach with the installation of each virtualization server application as you move through the next chapter.



Installing and Deploying VMs on Enterprise Servers

Now that your foundation in VM technology is firmly established, you'll learn about running VMs in production environments. Specifically, you'll investigate three server virtualization applications: Microsoft Virtual Server, VMware GSX Server, and VMware ESX Server. VM server products will be a vital piece of your infrastructure; therefore, revisit the details regarding the requirements of each product outlined in Chapter 1. Moreover, pay particular attention to the suggested best practices in this chapter. You don't want to deploy a server that impedes the workflow of your customers and coworkers. In the end, you're better off overbuilding a VM server than you are meeting the minimum requirements.

Being that we already showed how to install the workstation products for Microsoft and VMware, you'll see many similarities between Microsoft Virtual PC and Microsoft Virtual Server, as well as between VMware Workstation and VMware GSX Server. Some information may be redundant, but we'll cover all the installation information to be thorough. Like Chapter 3, we'll take a step-by-step approach when showing how to install each virtualization application in this chapter. You may not need to read each section; therefore, simply skip the sections you don't need.

Installing Microsoft Virtual Server

Because it's supported only on Windows 2003, our example installation of Microsoft Virtual Server will be on Windows 2003. If you have large memory requirements, make sure you use the version of Windows 2003 that meets your needs. Don't fall into the category of people who think they can save a few dollars by using Windows 2003 Standard when you require 6GB of RAM, because Standard supports only 4GB of RAM. If you're going to install Virtual Server on an existing Windows system, take a moment to mentally compare the host's hardware configuration with the following suggested best-practice minimums to ensure a successful Virtual Server deployment:

- Two RAID controllers (RAID 1 for the Windows 2003 host and RAID 5 for the Windows 2003 guests)
- Three 1Gb NICs
- Two 3GHz processors
- 4GB of ECC DDR RAM
- Five 15,000RPM 146GB SCSI hard disks
- 500–800MHz FSB

We also need to point out that Microsoft suggests disabling hyperthreading on Virtual Server hosts. Microsoft's rationale is that a load-stressed server may perform poorly during workload spikes. You may want to make this call on your own. Try running your server with and without hyperthreading, conduct some tests, and implement what's good for your environment. Many servers do just fine with hyperthreading enabled. If you've installed Virtual PC, then you'll be getting ready to experience a bit of *déjà vu* with the install of Virtual Server.

If you're installing from a CD-ROM and autorun begins the installation process, take a moment to disable it now. You can disable it by editing the registry and changing the value of `AutoRun` to 0 in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdrom`. If autorun is already disabled, you'll need to look for the `setup.exe` file, which is about 20MB in size.

Follow these steps to install Virtual Server:

1. On the setup screen, select **Install Microsoft Virtual Server** to begin the installation process.
2. The **Customer Information** screen requires that you enter a valid product key to continue. Enter your information and key, and then click **Next**.
3. The **Setup Type** screen prompts you to select between **Complete** and **Custom**. The **Custom** setup feature allows you to remove the following options: **Virtual Server Service**, **Documentation and Developer Resources**, **Virtual Machine Remote Control Client**, and **Virtual Server Web Application**. Leave the default selection of **Complete**, and click **Next**.
4. The setup program will then configure Virtual Server's **Administration Web site**. The default configuration port is 1024 using TCP. You can change the port number to meet your needs, but don't use the common ports less than 1024 (for instance, 25, 80, 110,

443, and so on). Using a port already slated for an existing service will cause problems for the existing service and for the Virtual Server Administration Web site. The Web site is automatically added to your IIS configuration. You'll also have to decide if the Administration Web site should run as the local system account or as the authenticated user. The default is to run as the authenticated user and is generally sufficient for most installs.

5. The Ready to Install screen is the end of installation configuration. Select Install to begin the installation.
6. If you don't have IIS installed and running, the install program will warn you to change your install choice to reflect the status of IIS. You can continue the installation, but the Administration Web site won't install. You'll need to rerun the installation program after installing and configuring IIS to get the Administration Web site to function.
7. If you elect to install IIS now, cancel the install of Virtual Server first. If you don't, the installer won't recognize the installation of IIS. You can install IIS by going to Add or Remove Programs in the Control Panel. Select Add/Remove Windows Components. Next, highlight Application Server, and select Details. But don't select the box to the left, as this will install all subcomponents. Highlight Internet Information Services (IIS), and select Details; again, don't select the box to the left, as this will install all subcomponents. Highlight World Wide Web Service, and select Details. Once more, don't select the box to the left, as this will install all subcomponents. Finally, check the World Wide Web Service box, and then click OK. You'll need to select OK a few more times to get back to the Windows Components Wizard's main screen.
8. Assuming all goes well, the install finishes with the Setup Complete screen. Select Finish to complete the installation.
9. The summary window will open and detail several points, such as the location of the installation, the address of the Administration Web site, and the location of documentation.
10. To test your installation of Microsoft Virtual Server, from the summary window, select the hyperlink to the Administration Web site. You'll be prompted to log in. By default, only local administrators can log in. Therefore, be sure to supply a username and password from the Local Administrators group.

If you don't have SSL enabled for your IIS server, you'll see a warning at the bottom of the login page stating that SSL isn't enabled for the Administration Web site. It's important you use secure communications with remote administration because you'll be passing passwords and configuration information to the system. You don't want to be the victim of someone unscrupulously using a network analyzer. Take a moment to configure SSL for your server.

To get a certificate for the Administration Web site, you'll have to set up Certificate Services from the Add/Remove Windows Components in the Control Panel, use a preexisting certificate or certificate authority, or generate one using the `makecert.exe` self-signing certificate utility, using SelfSSL, or using OpenSSL. Microsoft recommends you use a certificate server rather than using `makecert.exe`.

makecert.exe is simple to use and saves time in a lab or testing environment; you can download it by browsing to <http://download.microsoft.com/download/platformsdk/Update/5.131.3617.0/NT45XP/EN-US/makecert.exe>. Because makecert.exe more closely approximates the installation of a certificate obtained from a public certificate authority, we'll use it for securing our Administration Web site for this example.

Note The IIS 6.0 Resource Kit Tools also includes the easy-to-use self-signing certificate utility called SelfSSL. You can download the IIS 6.0 Resource Kit Tools at <http://www.microsoft.com/downloads/details.aspx?FamilyID=56fc92ee-a71a-4c73-b628-ade629c89499&displaylang=en>. SelfSSL is a bit easier to use because it automatically installs the certificate in IIS. To create a certificate, select SelfSSL from the IIS Resources program menu. A command window will open showing all the options of the SelfSSL command and will reveal the default behavior.

After obtaining makecert.exe, place it in an easy-to-find location from the command line, such as the root of your hard drive (C). Follow these steps:

1. Navigate to the location of the executable, and create a self-signed certificate with the following command:

```
makecert -r -pe -n "CN=<fully qualified domain name>" -b 01/01/2005  
-e 01/01/2006 -eku 1.3.6.1.5.5.7.3.1 -ss my -sr localMachine -sky  
exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12
```

If you don't know the fully qualified domain name (FQDN) of your server, run ipconfig/all from the command line. Concatenate the host name line and the primary DNS suffix line. Moreover, if you need to set the primary DNS suffix without joining your host to a domain, open System in the Control Panel. Click the Computer Name tab, select Change, and then click the More button. Enter the second-level and top-level domain names. You'll have to reboot your server for the change to take effect.

2. View the certificate using the Microsoft Management Console (MMC). You'll have to add the Certificates (Local Computer) snap-in. The certificate should be stored in the Computer Account certificate under Personal.
3. After verifying the creation of the certificate, you'll need to install it by going to Web Site Properties at the Administration Web site in IIS.
4. Select the Directory Security tab followed by Server Certificate.
5. The Web Server Certificate Wizard will start. Select Next to begin the configuration.
6. From the Server Certificate options screen, select Assign an Existing Certificate.
7. On the Available Certificates screen, select the certificate generated by makecert.exe.

8. You'll be prompted to supply the port for SSL communications. The default is 443.
9. The Web Server Certificate Wizard will generate a summary screen. Check the data, and make sure it's correct.
10. The Completing Installation screen indicates the certificate assignment was successful. Select Finish to end the install.
11. On the Virtual Server Properties window, select Edit under Secure Communications. On the Secure Communications screen, check Require Secure Channel (SSL) and Requires 128-Bit Encryption.
12. Close the Virtual Server Properties window, and then restart IIS to make the changes take effect (by selecting Action ► All Tasks ► Restart IIS).

Test the Administration Web site to verify secure communications. When you're testing your site, make sure to change the port number in the uniform resource locator (URL). The default install port uses 1024 and can no longer be used because of the security settings you made.

Note You may find that the Mozilla and Netscape Web browsers complain about certificates that are self-signed, such as in the `makecert.exe` and `selfssl.exe` applications.

Like Virtual PC, Virtual Server installs seamlessly. Now, browse to Microsoft's Web site at <http://www.microsoft.com/downloads>, and check for any patches or updates. If system patches exist, install them before continuing. With post-installation maintenance completed, you're now ready to start installing guest VMs, but before you begin, you should familiarize yourself with the Management User Interface and skim the information available in the help files. Knowing what options are available in Virtual Server and the scope of built-in assistance will be invaluable during deployment.

Installing VMware GSX Server for Windows

Installing VMware GSX Server is similar to installing VMware Workstation. GSX Server is licensed for unlimited user access and is a solid hosted virtualization product for small and large network workgroups. You can't upgrade software packages from Workstation to GSX Server. So, if you're upgrading a server from Workstation to GSX Server, be sure to uninstall Workstation first. You don't have to worry about losing data during the uninstall process. By default, VMware (and Microsoft) virtualization applications don't delete virtual hard disks during the uninstall process. You'll be prohibited from installing multiple instances of GSX Server on a single host; in addition, you may not install it on a host already running the Virtual Machine Console.

VMware provides for that ability to silently install GSX Server using version 2.0 of the Microsoft Windows Installer runtime engine. This is useful if you'll be rolling out GSX Server to several servers. You can verify the version of Windows Installer by running `msiexec.exe`. Assuming you've extracted the install packages for GSX Server by running `VMware-gsx-server-installer-<version number>.exe /a /s /x /d C:\temp\gsx` at the CLI, you can begin a silent install (including the license key and disabling autorun) by entering the following code at the CLI:

```
msiexec -i "C:\temp\gsx\VMware GSX Server.msi" ADDLOCAL=ALL /qn  
SERIALNUMBER=<serialNumber> DISABLE_AUTORUN = 0
```

You can use the `ADDLOCAL` and `REMOVE` options with the following qualifiers:

- All: Installs all options
- Network: Installs all networking options
- NAT: Installs the NAT service only
- DHCP: Installs the DHCP service only

If you don't want to install the DHCP and NAT services, you'll need to change the previous installation command to the following:

```
msiexec -i "C:\temp\gsx\VMware GSX Server.msi" ADDLOCAL=ALL  
REMOVE=DHCP,NAT /qn SERIALNUMBER=<serialNumber> DISABLE_AUTORUN = 0
```

Unlike workstation virtualization applications, GSX Server requires substantially more hardware to ensure that network availability and performance needs are met for guest VMs. That's why the following list of best-practice minimums for a VMware GSX Server deployment shouldn't be shocking. You should have a server equipped with a configuration that approximates the following:

- Two RAID controllers (RAID 1 for the Windows host and RAID 5 for the ESX Server guest VMs)
- Three 1Gb NIC interfaces
- Two 3GHz processors
- 4GB of ECC DDR RAM
- Five 15,000RPM 146GB SCSI hard disks
- 500–800MHz FSB

If you download your copy of GSX Server from VMware's Web site, it's about a 90MB file that will decompress itself before running. Follow these steps to install it:

1. Use Windows Explorer to find the installation executable, `vmware-gsxserver-<version>.exe`, and launch it. You'll first be presented with the GSX Server Master Installer Wizard, so click Next to continue.
2. The Setup screen appears; you must choose between Complete or Custom for the installation. The Custom selection allows you to remove the following:
 - VMware GSX Server
 - VMware Management Interface
 - VMware VmCOM Scripting API
 - VMware VmPerl Scripting API

The default value is Complete. Because the default is normally sufficient for most installations, click Next.

3. The Destination Installation screen preselects a default directory to deliver the install files. You can modify the directory path by selecting Change.
4. The Ready to Install screen gives you the option of making changes if necessary. If you're satisfied with your previous choices, select Install to begin the software delivery to the host.

The install process will take several minutes. VMware takes a moment to secure the Web-based Management Interface for you and creates a couple of desktop shortcuts. Upon completion, you'll be presented with the Installation Completed screen. Test your installation by launching the VMware GSX Server console.

Note You can change the default port GSX Server for Windows uses for the MMC by editing the `config.ini` file in `C:\Documents and Settings\All Users\Application Data\VMware\VMware GSX Server`. Add `authd.port = <portNumber>` to the file, and change the port to the one you want to use. You may need to reboot your host server.

After completing the install of GSX Server, don't forget to check VMware's Web site, http://www.vmware.com/download/gsx_security.html, for patches and security updates. As of this writing, a security update exists for GSX Server to correct a handful of issues with OpenSSL. The patch is smaller than a megabyte; you should install it in order to maintain the integrity of secure communications between the Management Interface and console sessions.

Installing VMware GSX Server for Linux

Installing GSX Server for Linux is similar to installing Workstation for Linux and can be performed using a TAR or an RPM file. When using GSX Server for Linux, be sure to install the virtualization application on an enterprise-class host OS, such as Red Hat Enterprise Server or Advanced Server. Regardless of vendor and version, you need to make sure your Linux platform support conforms to the following:

- The real-time clock function must be compiled in the kernel.
- The port PC-style hardware option should be loaded as a kernel module.
- The `inetd` process must start on boot.
- The `gcc` package must be installed.

As compared to a Microsoft server operating system, you can typically get similar performance with less hardware using a Linux server operating system; however, it isn't necessarily a good idea to use less of a server in production. Average hardware choices yield below-average performance. Your GSX Server Linux installation should be on hardware that's similar in nature to the following list of best-practice minimums:

- Two RAID controllers (RAID 1 for the Linux host and RAID 5 for the guest VMs)
- Three 1Gb NIC interfaces
- Two 3GHz processors
- 4GB of ECC DDR RAM
- Five 15,000RPM 146GB SCSI hard disks
- 500–800MHz FSB

Don't attempt to install Workstation and GSX Server on the same host system. If you attempt to install GSX Server on a preexisting Workstation installation, it will be upgraded to GSX Server. You'll also want to configure X Windows to manage guest VMs in a GUI console session. GSX Server for Linux consists of three packages. You can choose to install as few or as many as you need. If you're looking to approximate the same functionality as a Windows host, you'll want to install all available packages:

- GSX Server (`VMware-gsx-<xxx>.tar.gz`) for hosting VMs (Perl API installed by default)
- VMware Management Interface package (`VMware-mui-<xxx>.tar.gz`) for remote control
- VMware Virtual Machine Console (`VMware-console-<xxx>.tar.gz`) for local control

When managing systems remotely, you'll have to concern yourself with more than just physical firewalls getting in the way. For instance, default installations of Red Hat Linux and Windows XP SP2 use software firewalls. On the client side, firewalls can block you from using the VMware Management Interface and the Virtual Machine Console by prohibiting connections to the GSX Server host machine. Port 902 must be opened for Virtual Machine Console to access a GSX Server host. The Management Interface requires port 8333 to be open and 8222 if SSL is disabled.

Note To change the port GSX Server for Linux uses for the Virtual Machine Console, you must first determine if your system uses `xinetd` or `inetd`. For `xinetd`, edit the `/etc/xinetd/vmware-authd` file and change `port = 902` to reflect the new port number. If your system uses `inetd`, look for `/etc/inetd.conf` and change `902 vmware-authd` to reflect the new port number.

Whether you're installing from a CD or from a file downloaded from VMware's Web site, you need to make sure your file paths are reflected correctly in installation command statements. To begin the installation of GSX Server for Linux, go to the CLI and make sure you have root permissions. If you're in a testing lab, it will be easier if you log in as root. In production environments, you'll need to log on with the account for which the install is intended and then issue the `su -` command. As a small note, sometimes the `vmware-config.pl` configuration program will appear to hang during installation; you can press `Q` to advance to the next prompt. The step-by-step instructions for RPM and TAR installations follow. Refer to the section that supports your version of Linux.

Installing the RPM

Follow these steps to install the RPM:

1. If you're installing from a CD, you'll need to mount it first. At the CLI or in a terminal window, enter this:

```
mount /dev/cdrom /mnt/cdrom
```

2. Next, browse to the CD-ROM mount point:

```
cd /mnt/cdrom
```

3. To make sure the media is accessible, perform a directory listing:

```
ls -l
```

4. Now, locate the setup RPM:

```
find /mnt/cdrom -name VMware-gsx*.rpm
```

5. The system will respond with the location of the file. Change your present working directory to the RPM's directory:

```
cd /mnt/cdrom/<directory name>
```

6. Now, enter the following for the RPM install:

```
rpm -Uvh /mnt/cdrom/<directory name>  
/VMware-gsx-<version and build number>.rpm
```

7. You'll see advancing hash marks across the screen that indicate installation activity. After delivering the package, run the configuration program. Verify the installation of the RPM by querying the package management system:

```
rpm -qa | grep 'VMware'
```

8. The system will respond with the package and version number installed. It should match the RPM you specified with the `rpm -Uvh` command in step 6.
9. With installation out of the way, you'll need to run the configuration program. Enter the following at the CLI:

```
vmware-config.pl
```

10. Press Enter to read the license agreement. You'll need to use the space bar to advance through the document. Assuming you agree to its terms, type **yes** and press Enter.
11. The configuration program will question if networking is required. The default is Yes, so press Enter.
12. If you have multiple network adapters, you'll be asked which adapter should be bridged to VMnet0. If you're happy with the default selection, press Enter.
13. Next, you'll be asked if you'd like to configure any additional bridged networks. No is the default. Being that you can configure more bridged networks later, accept the default by pressing Enter.
14. Next, you'll be prompted if NAT networking is necessary. Select the default (Yes) for now by pressing Enter.
15. The installation script will ask to probe for an unused private subnet. The default is Yes. The scan can take a couple of minutes, and it will reveal what appears to be available for use.
16. The system will query if you'd like to use host-only networking in your VMs. The default is No. If this is what you want, press Enter. However, if you don't select Yes now, bridged and host-only networking won't be available to your VMs. You'll need to rerun the configuration script to enable the feature. If you select Yes, the script will probe for unused network IDs to use. After the probe, you'll be asked to configure any additional host-only networks. The default is No. Press Enter to continue.
17. The script will prompt you for the port to use for remote console sessions. The default is port 902. Press Enter to continue.
18. The script will then provide a location to store VM files. You can store virtual disks on any Virtual Machine File System (VMFS) partition. The default is generally a safe choice. Press the Enter key to accept the suggested path. If necessary, the path will be created. You'll be prompted to select the default Yes answer. Press Enter to continue.
19. Now, the install script will ask you for your 20-character serial number. Enter your validly licensed product key, and press Enter to continue. If you choose not to enter a serial number, the install will abort.
20. You'll be presented with some registration suggestions; simply press Enter to complete the install.

After the installation is complete, you can test the install by restarting the VMware service. At the CLI, run this:

```
service vmware restart
```

The system should respond by stopping the GSX Server's services and then starting them. Upon the reinitialization of the services, an "OK" should appear at the end of the restart dialog box.

Installing the TAR

If your OS doesn't use RPMs, you'll need to use a TAR to install VMware for Linux. The following instructions are generally suitable for several versions of Linux, including Red Hat. However, please remember that your system path structure may vary, so make changes as necessary.

1. Before installing the software, create a temporary directory for the installation TAR file:

```
mkdir /tmp/vmlinux
```

2. Next, copy the TAR file to the newly created temporary directory on your hard drive. Be sure your file paths correctly reflect the directories on your host system:

```
cp VMware-<version and build number>.tar.gz /tmp/vmlinux
```

3. Next, change to the directory where the copied files reside:

```
cd /tmp/vmlinux
```

4. Because the install file is an archive, you'll need to unpack it by issuing the following command:

```
tar xzf /tmp/vmlinux/VMware-<version and build number>.tar.gz
```

5. Once the decompression is finished, find the installation directory:

```
find /tmp/vmlinux -name vmware-distrib
```

6. Next, navigate to the installation directory:

```
cd /tmp/vmlinux/vmware-distrib
```

Finally, execute the installation program:

```
./vmware-install.pl
```

7. You'll be prompted several times during the install. Generally, the defaults are sufficient for most installs save a caveat or two. The first prompt asks you to confirm the install directory `/usr/bin`. Press Enter to accept the default.
8. Next, the install program will ask you the location of the init directories, `/etc/rc.d`. Assuming the default is correct, press Enter.
9. You're then prompted for the init scripts directory, `/etc/rc.d/init.d`. If the default is correct, press Enter.
10. The install program will then ask an installation directory for the library files, `/usr/lib/vmware`. Press Enter to accept the default.

11. The system will want to create the `/usr/share/man` directory for the manual files. Press Enter to confirm the default setting.
12. Next, you'll be asked to supply the directory for the VMware documentation files, `/usr/share/doc/vmware`. Press Enter to continue with the default. You may be prompted to confirm the creation of additional parent directories. Assuming this is okay, press Enter to continue.
13. Unlike the RPM install, the TAR install asks if you'd like to run the configuration program, `/usr/bin/vmware-config.pl`. The default is Yes; press Enter to run the VMware configuration program.

For both the RPM and TAR install, you can run the configuration program at any time. You'll also need to run the program any time you upgrade the Linux kernel or if you want to change the character of Workstation (for instance, to remove or add host-only networks). If you go to run the configuration program and it doesn't work, it's probably because of an error with the search path.

The use of the search path in Linux is similar to that in Microsoft Windows. It's a variable configured during system initialization. Linux uses the `PATH` command to search for the program that corresponds to input at the CLI. You can add a search directory to the path by issuing `PATH=${PATH}:/<directory>`. You can use the command `which` to verify the existence of a file in the search path, such as `which vmware-config.pl`. If you want to see what's currently listed in the variable, type `$PATH`. If you need additional help configuring the search path, you should refer to the distributor of your operating system.

Wrapping up this section, we want to point out that the help system built into VMware GSX Server for Linux depends on a Web browser being accessed from the `/usr/bin/netscape` directory. If your browser is located elsewhere (Netscape, Firefox, Mozilla, or otherwise), be sure to create a symbolic link to it at that location (for example, `ln -s <browser path> /usr/bin/netscape`).

Installing the VMware Management Interface

To be able to manage guest VMs on the local host with a GUI, you'll want to install the VMware Management Interface. The Management Interface requires that the `libdb.so.3` library be installed first. You can search for it by issuing `find / -name libdb.so.3` at the CLI. The operating system should respond with its location. The Management Interface comes as a TAR and is readily downloadable from VMware's Web site. Follow these steps:

1. To begin the install, go to the CLI or open a terminal window. If your user account doesn't have superuser permissions, you'll need to issue the `su -` command.
2. Next, browse to the location of the installation file. If it's on a CD-ROM, be sure to mount the CD-ROM first:

```
mount /dev/cdrom /mnt/cdrom
```

3. Create a temporary directory to extract the file:

```
mkdir /tmp/vminterface
```

4. Now, copy the TAR to the temporary location:

```
cp VMware-mui-<versionnumber>.tar.gz /tmp/vminterface
```

5. Change to the directory holding the TAR archive:

```
cd /tmp/vminterface
```

6. Extract the archive:

```
tar xzf VMware-mui-<versionnumber>.tar.gz
```

7. Change your present working directory to the location of the installation directory:

```
cd /tmp/vminterface/vmware-mui-distrib
```

8. Finally, run the installation script:

```
./vmware-install.pl
```

9. You'll be asked to read and accept the license agreement. Press Enter to display the agreement, and use the space bar to scroll through the agreement. Type **yes**, and then press Enter.
10. The script will ask for a directory to install the binaries. The default is normally sufficient.
11. You'll next be asked to supply the destination location of the initd directories. You can change the default if necessary.
12. Next, you'll be prompted for the location of the init scripts. Press Enter if the default is sufficient.
13. The install script will then provide a default location to install the Management Interface files. Press Enter to continue. If the directory doesn't exist, the script will ask if it's okay to create the directory.
14. You'll have to supply a location for the documentation files. The default is normally sufficient. Press Enter to continue. If the directory structure doesn't exist, the system will ask if it can create the directory. Select the default of Yes.
15. The script will finally end the installation by evoking the configuration script for the Management Interface.
16. The configuration script will ask you to set a limit for session timeouts. It defaults to 60 minutes. Press Enter to complete the install.
17. When finished installing the Management Interface, be sure to log off with exit.

With the Management User Interface (MUI) installed, it's time to test its functionality. Start X Windows, and launch the Management Interface.

Working with the VMware Virtual Machine Console

You can download the VMware Virtual Machine Console from VMware's Web site, from the Status Monitor tab of the Management Interface, or from the GSX Server CD-ROM. After gaining access to the VMware-console-*<version number>*.exe program, you're ready to begin the installation of the Virtual Machine Console. The following steps are designed for installation from a CD-ROM:

1. The install requires you to use the CLI or a terminal window. If your user account doesn't have superuser permissions, you'll need to issue the `su -` command.

2. If necessary, mount the CD-ROM:

```
mount /dev/cdrom /mnt/cdrom
```

3. Next, find the Virtual Machine Console file:

```
find / -name VMware-gsx-server-linux-client-<versionnumber>.zip
```

4. Browse to the directory with the file:

```
cd /mnt/cdrom/<directory_name>
```

5. Create a temporary location to copy the file to for extraction purposes:

```
mkdir/tmp/vmconsole  
cp /mnt/cdrom/<directory_name>  
/VMware-gsx-server-linux-client-<xxxx>.zip -d /tmp
```

6. Decompress the archive in the /tmp/vmconsole directory:

```
unzip VMware-gsx-server-linux-client-<version_number>.zip -d /tmp
```

7. Navigate to the /tmp directory:

```
cd /tmp/vmconsole
```

8. Based on your needs, install the RPM or TAR file as needed:

```
rpm -Uhv VMware-console-<version_number>.i386.rpm  
tar xzf VMware-console-<version_number>.tar.gz
```

9. Finally, locate the installer directory:

```
find / -name vmware-console-distrib
```

10. From the console-distrib directory, execute the installation script. If at any time during the installation the script appears to hang, press Q to advance the script:

```
./vmware-install.pl
```

11. You'll first be asked to read and accept the license agreement.

12. If you used the RPM version of the install, you'll need to run the configuration program for the Virtual Machine Console by running this:

```
vmware-config-console.pl
```

Note If you used a TAR file to install GSX Server for Linux, the script will automatically launch the `vmware-config-console.pl` script.

13. You'll be prompted as to what port you want to use for the remote console. Port 902 is the default. If the server you're going to manage uses a different port, change the port to match. If you used the defaults, press Enter to continue.
14. After the installation completes, be sure to log off the system with the `exit` command.

With the installation of all three packages completed, test your host's ability to set up and manage guest VMs locally. Execute `/user/bin/vmware` at the CLI. Take a moment to explore every menu, including the built-in help. VMware has excellent documentation for its virtualization software, and its Web site support is even better.

Changing GSX Server's Remote Console Port Number

If you're wondering if you can force the Remote Console authentication daemon (`vmware-authd`) to use a different port number, you can. In addition, if you want to take a minute to beef up security on your GSX Server, you should change the default port. Though a ping-sweep on your network may enumerate open ports on your server, changing the default port of GSX Server's Remote Console port will keep someone from specifically targeting the default port of 902. Don't select a common port, such as ports less than 1024. These ports are already assigned to common services, such as ports 25 (SMTP), 80 (HTTP), and 443 (HTTPS). To change the port, you'll need to follow these steps at the CLI:

1. Type `vi /etc/xinetd.d/vmware-authd`.
2. Find and change the port number to what you want. We'll use 9876 in this example.
3. Type `vi /etc/vmware/config`.
4. Change the `authd.client.port` entry to match your desired port number (9876).
5. Restart `xinetd` with `service xinetd restart`.
6. Restart the Management Interface with `httpd.vmware restart`.

GSX Server guests will be managed from the port you specified. Now, you'll need to use the VMware Remote Console application to connect to guest VMs. When connecting to the server, you'll need to specify your port number in the Connection field in addition to the IP address (for example, `10.10.5.136 9876`). After entering your information, select Connect to proceed.

After connecting to your GSX Server through the Remote Console application, you can verify the port change. For instance, from within a Windows operating system such as Windows XP, you can execute the `netstat` command at the command line with the `-n` option, `netstat -n`. Under the Foreign Address column, locate your server's IP address and verify that the port being used is correct.

If you don't want to have to specify the port number to use when using Remote Console, you can configure it to use a specific port automatically. For Windows systems, create a file and label it `config.ini`. Place the file in `C:\Documents and Settings\All Users\Application Data\VMware\VMware Remote Console`. The file should have one line of code:

```
authd.client.port = <portNumber>
```

If you're using a Linux client to connect to your host, you'll need to modify either `/etc/vmware-console/config` or `/usr/lib/vmware-console/config`. Add the following line of code:

```
authd.client.port = <portNumber>
```

Installing VMware ESX Server

We'll spend a bit more time on ESX Server than the other VM applications because it's a stand-alone product, which means an operating system isn't required to host the ESX Server virtualization application. ESX Server directly communicates with your system's hardware and is said to *run on the metal*. Additionally, we'll cover two configuration scenarios. The first scenario covers what might be involved with a troubled installation, and the second scenario details what you'll experience with a typical configuration.

If you escaped having to use Linux operating systems, you'll have to learn them if you want to take advantage of the power and scalability ESX Server offers. If you're familiar with Red Hat Advanced Server 2.1, you'll feel right at home with VMware ESX Server 2.1. If you're not really familiar with Linux, we'll give you a crash course in survival commands later in this chapter. You will, however, need to get your hands on a good Linux book for further investigation, such as *Beginning SUSE Linux: From Novice to Professional* by Keir Thomas (Apress, 2005).

ESX Server can be expensive to implement and is designed for enterprise-class internet-working needs. Though you don't need to be in a Fortune 500 company to use ESX Server, you'll want to install as much RAM and as many processors as you can; in addition, you'll need SMP licensing to run guest VMs with multiple virtual processors. The host will need at least two RAID controllers, one for ESX Server and one for guest VMs. Preferably, you'll want to store your guest VMs on a SAN. The network throughput needs of your ESX Server will be no different from the number of guests you'll host plus the needs of ESX Server. You'll want to have at least two network adapters, but install as many as possible—you'll want to take advantage of ESX Server's teaming features. This lofty rhetoric boils down to the following for best-practice minimums for ESX Server deployment:

- Two SCSI RAID controllers (RAID 1 for the ESX Server host and RAID 5 for the ESX Server guests)
- Four 1Gb NIC interfaces
- Two 3GHz processors
- 4GB of ECC DDR RAM
- Five 15,000RPM 146GB SCSI hard disks
- 500–800MHz FSB

Short of an extra RAID controller and a few NICs, you may notice that best-practice specifications aren't much different from what you'd specify for a new file server. At 60 percent utilization, you can expect to minimally run four to ten servers on the hardware listed.

If you're bent on using IDE devices for ESX Server, be aware that you can store VMs only on IDE devices with VMFS. It isn't that ESX Server can't run on IDE devices; it's just that VMware forces you to host on SCSI devices. With SCSI devices, you're assured the best possible performance for your guest VMs. Though SATA is clearly as good as some SCSI drives, and though it doesn't have the heat problems associated with extremely high spindle speeds, you won't get the ESX Server VMkernel to start after an IDE-only install. If SCSI drives are cost prohibitive and you need to leverage existing IDE drives, such as test lab equipment, you can invest in a SCSI-to-IDE bridge, such as the one available from Acard. It allows you to run IDE devices on a SCSI controller, so ESX Server will install just fine. Additionally, ESX readily installs on IDE Fiber Channel-attached storage devices.

Now that you know what you can expect from your hardware based on how it measures up to best-practice minimums, you can turn to the actual install. ESX Server has two installation modes, graphical and text. During the graphical mode installation of ESX Server, the setup wizard may utilize extremely low video resolution for your install. If this happens, you won't be able to view the contents of the entire install screen because online help is consuming a third of your viewable real estate. This will become a particular problem when it's time to enter your product serial numbers. You can close the help window by selecting Hide Help in the lower-left corner of your display. As a word of caution, don't close the help just to close it; if you keep it open, you'll be presented with helpful information that will answer many questions and refresh your memory on install topics.

VMware doesn't support running nested virtual machines, which are virtual machines running within virtual machines. The complexity introduced by nested virtual machines can cause complications with regard to the host's resources. Attempting to run a nested virtual machine will end in failure.

Though ESX Server is extremely quick to install and VMware recommends reading the entire installation guide, just skim the guide prior to installation. You'll find it especially helpful if you're upgrading an existing ESX Server, installing in text mode, or performing a scripted installation. We'll point out typical installation mistakes as we walk you through the setup process. Whether you use graphical or text mode installation, you'll have to learn to navigate the installation menus with a keyboard. The graphical install supports a mouse, but sometimes the install wizard won't correctly probe your hardware, only the keyboard functions. If you already jumped ahead and booted off the installation CD-ROM, you may have noticed the Red Hat logo during boot. If you're familiar with a typical Red Hat OS install, then you're already familiar with the navigation keys outlined in Table 5-1.

Table 5-1. *Red Hat Linux Installation Navigation Keys*

Key	Function
Tab	Navigates options
Space bar	Selects current option
Enter	Selects current option
Arrow keys	Navigates options

With navigation out of the way, go ahead and boot from the CD-ROM. You may need to adjust your host machine's BIOS boot-device priority from a floppy disk or the hard drive to the CD-ROM. If you have newer hardware, you can often change the primary boot device by using the system's boot menu: generally it's a function key, such as F12.

Upon booting from the CD-ROM, you'll be presented with the first installation screen that requires you to choose a "noapic" and/or "text" installation. Normally, it's sufficient to just press Enter and use the default, APIC/graphical mode. Advanced Programmable Interrupt Controller (APIC) mode avoids interrupt conflicts at the time of installation. APIC mode is helpful to use because devices may fail to share interrupts gracefully. After installation, errors related to devices contending for the same interrupt are generally resolved because one of the devices is often dedicated to a virtual machine. If interrupt conflicts persist, you'll have to resolve the issues by reconfiguring the hardware IRQs, swapping cards between different bus slots, or removing the device altogether.

To install, follow these steps:

1. Once installation is underway, the installation wizard will present you with the welcome screen; click Next.
2. The Installation Options screen presents you with three choices:
 - **Install Default:** Complete ESX Server install, formatting installation hard disks
 - **Install Custom:** Complete ESX Server install, allowing for keyboard and mouse customization
 - **Upgrade:** Preserves existing data and upgrades the host software

Since you're performing a new install and want to explore the available options, select Install Custom.

3. The Keyboard customization screen allows you to select your exact keyboard (if listed), layout, and use of dead keys. When *dead keys* are pressed, no visible character output is generated; they're generally used to make special characters (characters requiring accent marks). If you're inclined to create passwords with nonprinting characters, you'll want to enable dead keys. For this install, choose the defaults.
4. The Mouse customization screen allows you to select your mouse type and, if you want, emulate three buttons on a two-button mouse. Selecting the default is a good choice. You can always change these settings after installation.
5. The License Agreement screen asks you to agree to the terms. Check the box; if you don't, you might as well turn off your computer now.
6. To install guest operating systems, you'll have to obtain a serial number. You can request a demo license directly from VMware if you haven't purchased one. Enter your license number information in the fields provided. If you want to use virtual SMP in your guest systems, you'll have to obtain that license separately from the ESX Server license. Depending on your physical hardware, the SMP license will allow you to configure your guests with one to four processors. If you choose not to enter license information at the time of installation, that's fine. You'll be presented with a warning. You'll have to enter license information prior to installing guest virtual machines.

7. On the Device Allocation screen, you need to dedicate and/or share your physical hardware between the ESX Server console and guest VMs. You'll also need to divvy up your SCSI controllers and network adapters between the host system and guest VMs. Remember that one network adapter must be dedicated to the Service Console. If you share this network adapter with VM guests, you won't be able to remotely administer the server.
8. You'll be required to specify the amount of RAM to be reserved for the console. VMware makes suggestions on the amount of memory to be reserved based on the total number of guest VMs. You can select the minimum if you're going to host only a few VMs; however, to accommodate future needs and quell any issues regarding performance, select a memory reservation setting one or two steps higher than you need.

If you didn't install two network adapters in the host computer, this is where your actions will catch up to you. To manage ESX Server through the Service Console, one Ethernet adapter must be dedicated for it to use. Once again, ESX Server minimally requires two network adapters: one for the Service Console and one (or more) for the guest VMs. Now is a good time to install any additional NICs you want to use.

Tip To gain remote command-line access to an ESX Server or a GSX Server Linux host machine, use secure shell (ssh). This is a lot like Telnet except that communications are encrypted. If you're running a Windows-based operating system, any number of ssh programs are available for use, such as PuTTY. PuTTY is a Telnet and ssh client written for Windows operating systems and supports ssh v2. You can download it at <http://www.chiark.greenend.org.uk>.

If you insist on using one network adapter during your initial install, that's okay. ESX Server will automatically dedicate the only adapter to the Service Console. If you had multiple adapters, you'd be given a choice of how to assign each of the cards—to the console or to VMs. If you add more adapters after the install, don't reassign the original NIC dedicated to the console; ESX Server can experience trouble using the correct driver for the new Service Console network adapter.

Though it isn't necessary to have multiple SCSI controllers, your VMs will experience better performance if you have one SCSI adapter dedicated to the Service Console and one (or many) dedicated to your VMs. If you have one SCSI adapter, you can share it with the console by dedicating it to the Virtual Machines option and selecting Shared with Service Console. The Service Console will be installed to the highlighted drive. It can be either an IDE drive or an SCSI drive. However, VMs must be installed on SCSI drive. The VMkernel will fail to start otherwise. When you're done allocating your hardware, click Next.

9. If you receive a warning indicating that ESX Server has to initialize your hard drive, you'll lose all the data on the disk. Assuming you don't need any of the data, select Yes to initialize the drive. If you select No, that's the end of your install and the beginning of your data salvage project.

10. On the Disk Partitioning Setup screen, you can choose Manual or Automatic partitioning. If you're experienced and comfortable with Linux-like operating system disk partitioning, select Manual and have at it. You'll be required to create a minimum of three partitions:
- **/boot partition:** The mount point of the boot partition and location of the OS loader
 - **swap partition:** The space reserved for the memory swap file
 - **/ partition (read: root):** The location of file system and Service Console files

Assuming you're manually partitioning the system, you'll be limited to creating four primary partitions (three of which ESX Server will consume). In the fourth partition, you can create multiple logical partitions as your needs dictate.

If you want a guaranteed way to have ESX Server running in no time, choose Automatic Partitioning. If you have a single SCSI device, sda, the autopartitioning process will create the partitions in Table 5-2.

Table 5-2. *The Six Partitions of the Autopartitioning Process*

Partition	Label
/dev/sda1	/boot
/dev/sda2	/
/dev/sda3	swap
/dev/sda4	extended
/dev/sda5	vmkcore
/dev/sda6	vmfs2

When selecting Automatic, you'll be presented with the option to save any existing VMFS partition. VMFS partitions are where VMs are stored. If you're reinstalling ESX Server and want to save preexisting VMs, select Remove All Partitions Except VMFS. Because all partitions will be removed during automatic partitioning, the ESX Server installation wizard will present you with a warning. Assuming you don't mind losing all the data stored on your hard drive(s), including manufacturer-installed utility/management partitions, select Yes to continue.

11. The Defining Disk Partitions screen follows the Disk Partitioning screen. You can either edit the choices made by the installation wizard or click Next to continue. Before you continue jumping through the install, let's take a moment to tweak a few partitions using best practices. Adjust your /boot partition from 50MB to 100MB. Then highlight /boot from the Mount Point column, click the Edit button, and enter **100** in the Size (MB) field. Leave all other options as they are, and click OK.

The maximum any single swap partition can be is 2048MB, so you may need to create multiple swap partitions to achieve your calculated partition size. In addition, don't create a partition less than 256MB for performance reasons. Now, reconfigure the memory swap file and create more as needed: highlight Swap from the Type column,

and select Edit. Change the size of the partition to two times that of your installed RAM. For instance, if you have 512MB installed, enter **1024** in the Size (MB) field. Leave all other options as they are, and select OK.

12. With partitioning out of the way, the installation wizard provides you with the Network Configuration screen. We highly advise you to use a statically assigned IP address for your ESX Server. You really don't want to have to guess the IP address of the Service Console from week to week, so select Static IP and enter the required host information. Be sure to use an FQDN for the Hostname field (for example, `www.apress.com`). If you don't use an FQDN, you'll encounter host name resolution problems, and other systems will have difficulty finding your new ESX Server. After entering the requisite information, or if you're opting to use DHCP, click Next to move on with the installation.
13. On the Time Zone Selection screen, select the correct city and location where the ESX Server will reside. Time synchronization between networked hosts is important for authentication and logging purposes. For instance, some two-factor authentication systems will fail to work if the two hosts involved have the incorrect time. If your system clock uses Universal Coordinated Time (UTC), select the System Clock Uses UTC check box.

If you want your system to account for daylight saving time, click the UTC Offset tab. The Use Daylight Saving Time (U.S. Only) option is at the bottom-center of the screen; choose it if necessary, and then click Next.

14. The Account Configuration screen is where the superuser/root (administrator) password is configured. Be sure to do the following:
 - Choose a password you can remember.
 - Use an alphanumeric sequence.
 - Create a password greater than the six-character minimum, with eight characters optimally.

Security zealots may want to use nonprintable characters in passwords as well.

Tip If you have difficulty creating and remembering strong passwords, create a simple algorithm you can use on common words to generate solid passwords. For instance, you can substitute vowels for prime numbers and turn the word *football* into *f12b3l*. You can make your passwords even stronger by starting and ending them all the same way. For instance, you could start all passwords you create with a # and end them with a !; so *football* is now *#f12b3l!*, *baseball* is now *#b1s2b3l!*, and *password* is now *#p1ssw2rd!*.

The root user account allows for the complete administration of ESX Server. It's a powerful account and can do anything to ESX Server you want—unlike Windows OSs, the root administrative account can delete protected system files. Rather than using the root account for administration, you should use a separate account and

employ the `su` command to gain root-level commands only when it's required. Though using the root account in a training or test environment to help avoid complications is conducive to learning, habitually operating ESX Server as the root user in a production environment will inevitably result in damage to the file system. This happens because the superuser account overrides system safeguards, and mistakes generally go unnoticed. During initial installation, create an additional account for general system usage by selecting Add on the Account Configuration screen. When you're done, click OK and then Next.

15. The system is now prepared for installation. Click Next to begin the package delivery.
16. The final installation screen points out some important information. It notes the location of the setup log files:
 - **Install log:** `/tmp/install.log`
 - **Kickstart log (file containing preinstall choices):** `/root/anaconda-ks.cfg`

You can refer to the log files after the setup for troubleshooting or documentation purposes. Once the install begins, don't interrupt the process. If you do, you earned yourself a reinstall.

Tip Like the Blue Screen of Death, ESX Server has the Black Screen of Death. Any number of reasons can cause the system to black-screen, including incompatible hardware, interrupted installs, and bugs in the code itself. Make sure you do plenty of research before installing ESX Server to avoid black-screen headaches. For instance, if you're installing Citrix on ESX Server, you'll receive the Black Screen of Death if you don't disable the COM ports in the guest VM's BIOS. Moreover, you'll also experience the Purple Screen of Death in the event of serious hardware failures.

If you hang out for the install, you can get an idea of what's being delivered to your system by scanning the package summary. In total, it takes only five to ten minutes to install and only about 700MB of disk space. The Congratulations screen declares the installation is complete.

Before selecting Next to reboot, let's take a look at what's going on behind the scenes. The installation process has several sessions running that you can view by pressing Ctrl+Alt while simultaneously pressing one of the 12 function keys (F1–F12). Cycle through the different sessions saving F12 for last, because it may reboot your system. If your system doesn't restart, press Ctrl+Alt+F7: this will return you to the Congratulations screen where you can click Next to reboot. On reboot, you'll see the LILO Boot Menu screen. If you do nothing, the menu timer will expire, and the system will start. If you're familiar with how Linux or Novell OSs boot, you'll feel right at home while ESX Server initializes all its necessary processes; it ends in the Welcome to VMware ESX Server screen.

Now that the installation is complete, you're ready to configure ESX Server to host virtual machines. To configure the server, you'll need to have access to an additional networked computer with a Web browser. Is it beginning to feel a little like NetWare? VMware recommends that your Web browser be Internet Explorer 5.5 or 6.0, Netscape Navigator 7.0, or Mozilla 1.x.

Verifying ESX Server Configuration Information

When ESX Server is booting, you can get a good idea of what services are being loaded. If you've used Linux-type operating systems, you'll encounter a familiar boot process. Before you start the configuration of ESX Server, let's make sure the system is ready to communicate on the network. For Unix and Linux users, this section will seem basic, but it's essential for Microsoft administrators. If you're familiar with the information, skim this section and use it as a review.

You'll want to check the server's basic configuration, so you'll need to gain shell access by pressing Ctrl+Alt+F2. You can switch between shell sessions by pressing Ctrl+Alt+F1 through F12. So you can get an idea of how it works, toggle between the Welcome screen and the shell Login screen.

At the shell Login screen, enter the administrator's username root, and use the password you configured earlier for the account.

Caution Unlike Microsoft OSs that are generally case insensitive, every command in Linux-like operating systems is Case Sensitive. Everything! For instance, ROOT isn't the same thing as Root or root. These are all three different user accounts, and only one is the superuser administrator account—root.

After logging into the ESX Server CLI, the first thing you'll want to check is the system's IP address information. Type `ifconfig | less` at the command line. (You could use the pipe with the `more` command, but `less` allows scrolling up and down with the arrow keys. Type `q` to quit.) You should have information for the loopback address (lo) and a minimum of two Ethernet controllers (eth0 and eth1). Remember that you need at least two NICs to bridge or NAT VMs to your network. If you dedicate a single NIC to VMs, you'll lose the ability to connect to the Service Console with a Web browser.

Confirm that your server's information is correct, and then test the following using the ping command. You can use Ctrl+C to break out of the looping echo requests and echo replies.

- Test the server's TCP/IP stack by pinging the loopback address with `ping 127.0.0.1`.
- Test the server's default gateway with `ping <your_gateway_router_IP_address>`.
- Test DNS by pinging a Web site with `ping www.apress.com`, for example. Some Web servers block ICMP traffic, so you may get only the domain name to resolve to an IP address.

Viewing Configuration Files

If you have a problem with the ping tests listed previously, you'll have a problem configuring ESX Server through a Web browser. If no IP address information exists, you can attempt to use the `setup` command to configure your IP address information. A better solution to verifying configuration information is to check the configuration files from the command line. Though we could easily turn this into a Linux tutorial, we'll just walk you through the basics.

Editing configuration files starts with a good text editor and enough knowledge to be dangerous, so now is as good a time as any to discuss the Vi editor: some of us hate it, and some of us love it. It comes with nearly every version of Linux; you can even find it in the Microsoft NT Resource Kit. Whether you're a fan of Vi or not, it's an excellent text editor that gets the job done. Table 5-3 lists the commands you'll need to memorize for ESX Server administration.

Table 5-3. *Vi Survival Commands*

Command	Action
vi	Starts the Vi editor
vi <filename>	Starts Vi and opens the given file for editing
:q!	Exits the Vi editor without saving changes (quit!)
:wq!	Exits the Vi editor and saves changes (write and quit!)
:set nu	Turns on line numbering
esc	Exits text insert mode
i	Enters text insert mode
Arrow keys	Navigates the cursor
Backspace key	Deletes previous character
Delete key	Deletes selected character

Vi has two different running modes: command mode and edit mode. In command mode, every keystroke is interpreted as a command and performs file functions such as editing and saving. These commands are referred to as *colon commands* because they start with a colon. You know you're in command mode because you won't see "INSERT" in the lower-left corner of your display. The lower-left corner should display a colon when executing colon commands. If you have any doubts as to what mode you're in, press the Escape key (several times) to enter command mode.

In text mode, typed characters are echoed to the screen and temporarily become part of the open document. Enter text mode from command mode by typing a lowercase i (for insert mode). Use the arrow keys to navigate the cursor, and use the Backspace and Delete keys to remove characters. To quit without saving your current work, or to save your current work and quit, refer to Table 5-3.

Now that Text Editing 101 is out of the way, you can verify your ESX Server configuration information using the filenames listed next. For each file, open and view your system's information. If anything is incorrect, take a moment to fix any errors and save your changes.

You can find configuration files relating to the host name (FQDN) of your system in three places:

- vi /etc/sysconfig/network
- vi /etc/hosts
- vi /usr/lib/vmware-mui/apache/conf/httpd.conf

IP address configuration information for the first network adapter is located in /etc/sysconfig/network-scripts/ifcfg-eth0.

The contents of the `ifcfg-eth0` file will look similar to the following:

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=x.x.x.x
IPADDR=x.x.x.x
NETMASK=x.x.x.x
NETWORK=x.x.x.x
ONBOOT=yes
```

IP address information for the second network adapter is located in `/etc/sysconfig/network-scripts/ifcfg-eth1`.

The contents of the `ifcfg-eth1` file will look similar to the following:

```
DEVICE=eth1
ONBOOT=yes
```

You can find DNS server configuration information in the `resolv.conf` file. Review the contents of it by executing the following command:

```
vi /etc/resolv.conf
```

Default gateway information is located in the network file. Make sure the default gateway is properly set. If not, you'll spend needless time troubleshooting your host. Verify your host's settings with the following:

```
vi /etc/sysconfig/network
```

After verifying and making any necessary changes, restart the networking service by entering the following command:

```
service network restart
```

Continuing at the CLI, execute `hostname <FQDN>` (for example, `hostname www.apress.com`) to make the host name change effective without a reboot.

Next, repeat the ping tests, and verify your system's host name with `uname -a`.

With positive results from network connectivity and name resolution testing, the ESX Server installation is complete. You can move onto your post-installation housekeeping. You'll perform these tasks by configuring and using the MUI.

Using Linux Survival Commands

Now that the ESX Server installation is out of the way, you need to look at a few Linux survival commands. If you're accustomed to Linux operating systems, this section will be a snooze; however, if you're new to the world of Linux, this section should offer some much needed information. Table 5-4 is a quick reference for the commands you can use at the Service Console. A complete discussion of Linux and its wealth of commands are beyond the scope of this book.

Table 5-4. *Linux Survival Commands*

Command	Purpose	Example
man	Displays extended help	man reboot (the letter <i>q</i> quits help)
--help	Displays basic help	reboot --help
clear	Clears console screen	clear
su	Switches to superuser or between users	su or su <userid>
ls	Lists files in a directory Lists long listing with hidden files Lists scrollable long listing	ls / ls -al ls -l less
pwd	Reveals present working directory	pwd
cd	Changes to new directory	cd <directory>
mkdir	Makes a directory	mkdir <name>
rmdir	Removes an empty directory	rmdir <directory>
rm	Removes a file (or directory)	rm <filename>, rm -r <directory name>
mount	Attaches to media, as in CD-ROM Mounts CD-ROM Mounts floppy drive	mount <device> <destination> mount /dev/cdrom /mnt/cdrom mount /dev/fd0 /mnt/floppy
eject	Ejects CD-ROM	eject
cp	Copies a file or directory	cp <source> <destination>
mv	Moves a file or directory	mv <source> <destination>
find	Locates a file or directory	find / -name <name>
tar	Creates a compressed file Extracts a compressed file	tar -czvf <source> <destination> tar -xzvf <source> <destination>
umount	Disconnects from media or partition Unmounts floppy drive Unmounts CD-ROM	umount <directory or device> umount /mnt/floppy umount /mnt/cdrom
fdisk -l	Lists disk partitions	fdisk -l
service	Controls system services and stops Starts system services Restarts system services	service <service> stop service <service> start service <service> restart
passwd	Manages passwords	passwd <user ID>
ps	Lists system processes	ps -efa less
free	Displays memory statistics	free -m
ifconfig	Reveals IP addressing information	ifconfig -more
uname	Reveals host name	uname -a
ping	Stands for Packet Internet Groper	ping <IP address>
reboot	Restarts the ESX host server	reboot

Working with the Management Interface

Now that you've confirmed and tested basic ESX Server functionality, let's move onto configuring the system to host VMs. You'll conduct post-configuration tasks through the MUI. In a nutshell, the MUI is ESX Server's primary front-end management tool and is delivered by an Apache-powered Web page. Because it requires a Web browser to function, be sure to turn off any pop-up stoppers you may have running. Failing to allow pop-ups from the ESX Server will result in you not being able to see VMware ESX Server's configuration windows. In addition, you'll need to be able to contact the ESX Server via its host name from your workstation, so configure DNS or edit your local host file for name resolution. We'll cover five main post-configuration tasks:

- Customizing the MUI port number
- Configuring initial MUI configuration settings
- Creating the VMFS swap file
- Assigning network cards
- Constructing virtual switches

Understanding MUI and SSL

You can configure ESX Server to use a certificate from your certificate authority or continue to use the one ESX Server generates for itself during the install. If after installing ESX Server you find it necessary to change the host name of the system, you should re-create the system's SSL certificate to avoid getting SSL security alerts while accessing the administrative interface of ESX Server from your Web browser. An SSL alert proclaims that the name on the security certificate is invalid. This occurs because the host's name isn't the same as the one on the certificate. If you didn't supply an FQDN during the install, you'll find that the name on the certificate is *localhost.localdomain*. You can manually reconfigure ESX Server, or you can use the `enableSSL` script provided by VMware to generate another self-signed certificate with the correct host name. You can download the `enableSSL` script from <http://www.vmware.com>. To use VMware's script, you need to follow these steps:

1. Create a file with Vi, and name it `enableSSL.pl`:
2. Type the text from the `enableSSL.pl` script into `enableSSL.pl`, and save it.
3. Rename the old SSL directory:

```
mv /etc/vmware-mui/ssl /etc/vmware-mui/ssl-old
```

4. Run the script you just created:

```
perl enableSSL.pl localhost root <root_password>
```

5. Verify that the new SSL directory was created:

```
ls /etc/vmware-mui/ssl
```

6. Restart the Apache daemon:

```
service httpd.vmware restart
```

7. When all proves to function, delete the old directory:

```
rmdir -R /etc/vmware-mui/ssl-old
```

Test your new SSL certificate by browsing to your host's MUI. View the certificate, and verify its name is the same as the host's.

Configuring the ESX Server Installation: Part One

Sometimes things don't go as planned during the ESX Server installation. If something goes awry, the System Configuration Wizard will present itself after logging into the MUI. The wizard generally launches for one of three reasons: the ESX Server has been installed without being configured, ESX Server has been upgraded from a previous version, or the VMkernel failed to load. If ESX Server proclaims, "Your system has not yet been configured, or the existing configuration is incomplete; the wizard will guide you through the system configuration process," hold onto your keyboard: ESX Server is throwing you a curveball. If you run into problems with a new installation, repeat the install to see if you can duplicate the configuration process. You don't want any anomalies related to software or hardware sneaking into production.

After connecting to your ESX Server through a Web browser, select Yes to the SSL security alert. If you want, you can view it and install the certificate. You'll next be presented with the VMware Management Interface login screen. For now, enter **root** as your username and the appropriate password. Check your Caps Lock key, and remember that everything is case sensitive. Once you've logged into ESX Server's MUI, you'll need to complete several post-install tasks if something went wrong during installation, which may include the following:

- Accepting the license agreement
- Configuring the startup profile
- Setting up the storage configuration
- Configuring the swap configuration
- Setting up network connections and switches
- Adjusting security settings

If your install fails to provide the initial configuration pop-up—it's a good thing! Keep reading, though, because we'll cover a few things here that aren't covered if you have an uneventful installation of ESX Server. If something went awry during the installation, the ESX Server Configuration screen will open in a new window.

License Agreement

The first thing you must do is accept VMware's license agreement on the End User License Agreement screen. Be sure to check the box signifying your compliance and enter your serial numbers. If you didn't purchase Virtual SMP, leave the relative fields empty. If you don't have the proper licensing information, you won't be allowed to proceed with the server configuration. You can get a demo license by registering with the VMware Web site.

View your license rights, and confirm the correctness of the license expiration, number of supported host machines, number of supported host processors, and number of processors supported for each virtual machine. If all is okay, click Next to continue.

Startup Profile

Before being presented with the Startup Profile screen, there will be a brief pause while the startup profile is loaded. You'll need to specify how much memory is to be reserved for the system console. This number will be based on how many VMs you intend to host on your server. Being that RAM is cheap and you want to err on the side of performance, under System Startup, pick a setting one greater than you need. For instance, if you plan to run eight virtual machines, use the settings for sixteen virtual machines.

Under Hardware Profile, you now need to assign the network adapters to the Service Console or to VMs. If you have only one NIC, you can assign it for VMs to use, but you'll lose the console's network connection. For ease of administration, you should install two NICs. If not, you'll be forced to configure everything from the CLI or be left to hack together a locally running copy of X Windows. Though running X Windows locally is possible, it consumes valuable resources and isn't officially supported; moreover, it's a good way to cause your VMs to quit functioning. After dedicating memory and NICs as necessary, click Next to continue your installation.

Storage Configuration

ESX Server may take a moment to save your startup profile and reboot. On reboot, log back into the MUI. You may receive an error message stating that the system isn't configured or it has an incomplete configuration. You have a choice to view the system logs, so take a moment to explore them by selecting the System Logs hyperlink. Cycle through the four tabs, and make note of any pertinent information. The logs can get quite long and be cryptic; if you're not a Linux guru, be prepared to break out your Linux books to decipher the contents. Whether or not you're required to troubleshoot the install process, you'll eventually see the Storage Management Disk and LUN configuration screen, assuming all the necessary partitions were created during the install. You can take the opportunity to repartition the system, or you can click Next to continue. If you have to create partitions, you'll need to know that guest VMs can be stored only on VMFS partitions or physical disks. You must have a least three partitions: /boot, swap, and /. The partitions created during the install that contain the Linux file system and the Service Console swap partition can't be changed.

Swap File

On the Swap Configuration screen, select Create to make a memory swap file. You'll be presented with another swap file configuration screen. Generally, ESX Server does a good job of determining the filename, size, and activation policy. If you'd like to change the storage volume or filename, now is a good time to do it. Make sure the swap file is a minimum of 256MB and no larger than twice the server's physical memory. If desired, you can direct ESX Server to allow you to manually start the swap file. If you choose this as the activation policy, you'll have to start it manually by entering `vmkfstools -w` at the CLI. Take a moment to activate your swap file now. If you do, you won't have to reboot.

Network Configuration

After completing the Swap Configuration screens, you'll be presented with the Network Connections/Virtual Switches dialog boxes. If necessary, you'll need to configure network adapters that guest VMs will use. Before setting your speed and duplex, know how your switch is configured. If you decide to team network adapters, know the team bonding type of your connecting switch. As a rule of thumb, avoid autonegotiation and reconfigure your switch if necessary. You can experience high latency intervals while the NIC and switch attempt to hash out their respective settings.

If you have one NIC, you'll need to configure the adapter after completing the Configuration Wizard. You'll need to perform the following steps:

1. Use Vi to edit `/etc/module.conf`. Comment out the alias `eth0` line by placing a pound, hash, square, gate, or number sign at the beginning of the line:
2. Connect to ESX Server from a Web browser, `http://<hostname>`. Select Edit, and under Device Allocation, reassign the NIC to Virtual Machines. Reboot.
3. Log in as root at the console. Use Vi to edit `/etc/rc.d/rc.local`, and append the NIC's name. If you're unsure of your NIC's name, you can use the `findnic` command as an aid.

```
insmod vmxnet_console devName="vmnic0"
ifup eth0
```

4. Reboot, and the single NIC sharing is complete. You'll now have to perform all administration of ESX Server at the console.

Binding NICs together to form a bond is the same as teaming NICs. When creating bonds, be sure to use functionally similar physical NICs because ESX Server will support only what they have in common, such as VLAN tagging, TCP segmentation offloading, and checksum calculations. You're limited to bonding a total of ten physical adapters for any given VM. You may also be required to create and configure a virtual switch. You'll need to supply a name for the switch in the Network Label fields. Next, look at Outbound Adapters, and assign an available NIC from the list to the switch. Then click Create Switch.

Similar to a physical switch, a virtual switch has 32 ports; it can be used to load balance and team NICs and provide redundant links. You can connect one VM to each port. Virtual switches have labels and can be renamed only when no VMs are using them, so take precautions when doing this (double-check everything). If you rename a switch with a programmatically attached VM, it won't boot. If you're creating a network of host-only VMs, deselect all NICs from the configuration of the switch. In addition, you can create *port groups*, which are similar virtual local area networks (VLANs). VLANs logically group switch ports into a single network and allow for secure communications amongst the hosts. Before creating a port group, you must have an existing network created. You can have IDs between 1 and 4095. When you're done configuring the switch and NIC, click Next to continue.

ESX Security

The ESX Server installation process will now present you with the Security Settings screen. By default, the system selects high security for you. If it's necessary to make changes, choose Custom to enable the minimum number of services you need. Though this should go without being written, the more you open up ESX Server, the more vulnerable it is to attacks. When you're done selecting your security settings, click Next to continue.

After completing the ESX Server Configuration screen, you should be presented with the VMware Management Interface Login screen. Should your system continue to fail the configuration process, double-check your hardware configuration, and make sure it meets the minimum requirements for ESX Server.

If you're still having problems, make sure you have at least one NIC and one SCSI hard drive available for guest VMs: both are minimum requirements. You'll also notice that the host name isn't properly configured. After triple-checking your system against the minimum requirements and VMware's HCL for ESX Server, give the installation another shot. You'll find that it really is seamless, and you can move onto what you should expect to see from a successful installation.

Caution Installing X Windows on ESX Server for educational or test purposes is okay. Don't install X Windows on a production ESX Server system—the overhead will ravage resources and render a production server useless. Given the aforementioned warning, you can install X by installing it as well as its dependencies by using RPMs from Red Hat 7.3.

Configuring the ESX Server Installation: Part Two

In Part One, we touched on several problems you'll run into with the ESX Server install and how to negotiate fixes with the aid of the Configuration Wizard. Now let's look at what you should expect to see when things go well. Connect to the MUI using your Web browser. You'll be presented with two warnings, one stating that "No swap space is configured or none is active" and the second stating that "No virtual Ethernet switches found." Select Reconfigure at the end of the first warning, and follow these steps:

1. Under the heading Configured Swap Files, select Create. The swap space on ESX Server affords VMs the ability to use more memory than physically available. ESX Server does a good job of recommending the amount of space to make available, but you should adjust the File Size field to be at least 2GB or two times the amount of physical memory in the host system. Be sure that Activation Policy is set to Activate at System Startup. If you don't, you'll have to manually activate the swap space on each boot. When you're done making changes, select OK to continue. You'll be returned to the initial Swap Configuration screen. On this screen, under Configured Swap Files, select Activate. If you don't activate the swap space now, it won't be active until the system is rebooted. Select OK to continue.
2. On the Status Monitor page of the MUI, you should have one warning remaining. Select Reconfigure at the end of the warning. The system will present you with the Virtual Switches tab from ESX Server's Network Connections settings. Take a moment to read about virtual switches on this screen, and then select Click Here at the bottom of the screen to create a virtual Ethernet switch. Under New Configuration, the system will provide a suggested network label for the switch; you can make changes if necessary. Under Bind Network Adapters, you'll then have to select which outbound adapter should be bound to the new virtual switch. After making any changes, select Create Switch to continue.
3. Next, select the Physical Adapters tab. You'll need to set the speed and duplex of the network adapter bound to the new virtual switch in the Outbound Adapter settings section. You can run into problems if you leave the setting on Auto Negotiate. When you're done making changes, select OK to continue. The system will return you to the Monitor Status tab of the MUI. All warnings should now be gone.

Take a moment to explore the Memory tab. It displays a summary of physical and reserved memory. In addition, it will summarize the memory consumed by any configured and running guest VMs. Take a moment to explore the Options tab. The Options tab details all configurable information as related to your ESX Server. Click through each field, and read all available information. You'll want to be somewhat familiar with these settings as you install guest VMs and make changes to support enterprise servers.

Summary

Building on your knowledge of how to physically size and prepare enterprise-class servers from previous chapters, you should now feel comfortable installing, configuring, and deploying server-class virtualization applications. You should also be able to prepare and install both Windows and Linux hosts for running multiple production virtual machines, and you should be able to troubleshoot basic installation issues for each of the server-class virtualization applications. In the next chapter, you'll focus on managing production VMs on enterprise servers. You'll look at management and monitoring issues regarding VMs running on production servers and learn about VM scripting techniques.



Deploying and Managing Production VMs on Enterprise Servers

Now that you know how to size hardware for VM servers and install VM server application software, in this chapter you'll install guest VMs and learn about management and monitoring issues regarding production servers. We'll use a step-by-step approach to expose techniques for monitoring resource utilization, monitoring performance, scripting fault monitoring, and scripting fault-tolerant VM recovery. We'll leave enterprise-class virtualization topics to Chapter 14, where we'll discuss virtualization techniques to help manage the enterprise.

Deploying VMs with VMware GSX Server and ESX Server

Creating VMs for VMware GSX Server and ESX Server is similar to creating them for VMware Workstation. Though you should know what you want in the way of hardware and naming conventions with a VM before creating it, you can always edit the VM after it's created. We'll show how to create VMs on Windows and Linux hosts for GSX Server and install a guest VM on ESX Server. To make the most of this topic, skip to the section that interests you. After discussing how to create guest VMs, we'll cover individual hardware options, and you'll learn how to migrate guest VMs between host systems.

Building VMware GSX Server VMs

Building a Windows VM is identical to building a Linux VM. The major difference between the two is the host's operating system. To start, launch VMware Server. For Windows host systems, select Start ► Programs ► VMware ► VMware GSX Server/VMware GSX Server Console, and select File ► New Virtual Machine to launch the New Virtual Machine Wizard. To launch VMware for Linux, make sure X Windows is started and type `vmware` in a terminal window; then select File ► New Virtual Machine.

During the installation process, you have the option of performing a default or custom install. As with all installs, you'll want to customize your installation, so select Custom. Wizards are great, but you want to control as much of all installation procedures as possible. Then select which guest operating system type to install.

For this sample install, we'll select Microsoft Windows. If you want to install a Linux VM, choose Linux. From the Version drop-down menu, select the OS you'll be using for the guest VM. As you've probably noticed, you have one heck of a selection for guest VMs. Though it's not listed, you can install DOS 6.22 by selecting Windows 3.1. For discussion purposes, we'll be configuring Windows 2003 Server. Regardless of what you choose, these directions will be identical to the choices you'll need to make. As we explained in Chapter 4, you'll have to select a name for your VM and a storage point for the guest VM's virtual hard disk that adheres to best practices.

For example, a guest virtual machine named `vmlinux2` would be stored in `D:\VirtualMachines\vmlinux2`. For Linux systems, adjust your path statements as necessary (for example, `/root/VirtualMachines/vmlinux2`). Remember, you don't want the host OS and guest VMs to use the same hard disks or same physical controllers. If you're in a lab or educational environment and don't have access to expanded resources, simply use the default directory—just create a custom directory for each guest VM. It makes management much easier by having all the files for each VM stored in a unique directory.

Caution If you use the default location for VM guests and use folder redirection for My Documents, you may experience adverse performance when large virtual disks are copied to the redirected folder.

The Access Rights selection screen requires you to decide if you want the VM to be available to everyone using the machine or only under the account from which it's being created. If this will be a production server, be sure to uncheck the private selection.

On the Startup/Shutdown Options screen, you have the ability to manage the startup and shutdown characteristics of the new guest and the account under which the VM will run. VMs can run under the user account that powers it on, a local system account, or a preconfigured user account. If you're building a server to be a LAN resource, it's often best to set the VM to run under the local system account. Additionally, you can select to have the guest start up and shut down with the power cycles of the host.

Determining the power-up or power-down characteristics of a guest VM will be based on individual circumstances. If you have VMs power on at host startup, you won't have to manually start each one. Conversely, it may make servicing the host more difficult. Having guest VMs automatically power down with the host is convenient, and this plays nicely into any UPS management shutdown strategies. Moreover, to prevent the loss of data, it's a good idea to power down all VMs before bouncing the host.

The wizard will recommend the amount of RAM to use for the guest VM. Generally, the wizard performs an okay job of selecting the amount of RAM for you. But before "mashing" next, notice that the wizard reveals the minimum, recommended, and maximum amounts of RAM to use for the guest. These are merely recommendations. You should have performed a bit of research to determine the amount of RAM your guest OS actually requires.

In lab and learning situations where RAM is limited, feel free to throttle memory back. In production environments, be sure to allocate as much RAM as you'd normally provide for a system. If you have the ability to cycle your servers often, you can start with the minimum amount of RAM and work your way up until you reach a satisfactory level of performance.

The Network Type Selection screen lists the networking characteristics you can make available to your VM. If you want to connect guest VMs as quickly as possible to your test or production network, select Use Bridged Networking. If you want this guest VM to be in a private network, select Use Host-Only Networking. If you have second thoughts as to the networking types, please refer to Chapter 2, which covers the networking types thoroughly.

The I/O Adapter Types selection screen designates whether a BusLogic or an LSI adapter will be used in the VM.

An easy way out of this choice is to use the BusLogic default. However, you'll get a little better performance from the LSI Logic option. As a word of caution, you'll need to have the LSI driver available when you install the actual OS on the guest VM.

Tip It's a good idea to download the LSI Logic driver for your VMware guest VMs. It's well worth the effort. Additionally, you'll want to copy the driver to a floppy disk and make an ISO image of the disk. You can then mount the image like a physical floppy drive and have it available during the guest OS install. You can download ISO image-making software from several vendors on the Internet for a trial period; for instance, try MagicISO from <http://www.magiciso.com> or Undisker from <http://www.undisker.com>.

If you're creating a new VM, on the Select a Disk screen, leave the default choice to create a new virtual disk. If you're setting up a VM to use physical disks, refer to the "Configuring VMware GSX Server and ESX Server Virtual Hardware Options" section in this chapter.

The Select a Disk Type screen provides two options: IDE and SCSI. You'll generally want to match the virtual hardware with the physical hardware.

You have the option of using a virtual IDE or SCSI disk despite the physical medium in the host. Virtual IDE is limited to two devices per channel, and virtual SCSI is limited to seven. For IDE devices, you're limited to four.

On the Specify Disk Capacity screen, you'll need to decide the total maximum amount of space to be used by a guest VM. You'll also need to decide if you want the disk to be dynamically expanding, which is the default, or fixed. Fixed disks have the space allocated up front, and the space will always be available to the guest. Dynamic disks expand as necessary. This means they'll grow to the maximum size specified. If you're building a server, be sure to allocate all the space up front. Dynamic disks fragment the host's hard drive and will eventually impact performance.

You'll get better performance with fixed disks because dynamic disks tend to fragment your hard drive. If you're in an educational or lab type environment, dynamically expanding disks tend to work well. For production environments, you'll want to stay with fixed disks. Being that our example is geared around a Windows 2003 Server and our test system has sufficient disk space, we'll select the Allocate All Disk Space Now option.

As for how many gigabytes to use for the system disk, this is really a matter of preference and experiential knowledge. In the NT days, 4GB was sufficient. Today, with the code bloat involved with Windows 2000 and 2003, you'll find you'll want a disk with 8GB or even 12GB reserved for the virtual hard disk. The value of having ample space is that you can copy the i386 directory from the Windows CD, service packs, and drivers to the hard drive for future use. Having plenty of room also allows you to keep log files longer.

Caution You have the option to split virtual disks into 2GB files. If your host's operating system doesn't support files larger than 2GB, like EXT or VFAT, you'll need to select this option during virtual disk creation. For some Linux operating systems, the wizard will autoselect the Split Disk into 2GB Files option.

If you chose to preallocate space for your guest's virtual disk up front, the system will inform you it will take a moment to create the disk. The system may appear to hang, so be patient during virtual disk creation.

The wizard, on the Specify Disk File screen, will prompt you for the name of the virtual disk and the location where you'd like to have it stored. Choose the same location and naming convention you established earlier in the VM creation process.

Take a moment to click the Advanced button and explore these options. On this screen, you can specify the virtual disk's device node and the disk mode. Nodes reference a virtual disk's relationship to its bus and device ID. For instance, the first IDE device on the first channel is 0:0, and the first device on the second channel is 1:0. Specify the virtual disk mode from the Mode section, and set the disk to Independent Persistent or Independent Nonpersistent. If you want to easily use snapshots, leave the default disk mode selection.

When the disk creation process is completed, you'll be returned to the VMware Server MUI. The newly created VM appears in the left column under Favorites, which are merely shortcuts to your VMs. Before powering on your new VM, take a moment to look at the VM's hardware settings using the Configuration Editor by selecting Edit Virtual Machine Settings.

Building VMware ESX Server VMs

Building ESX Server VMs is similar for both Windows and Linux VMs. It's a wizard-driven process, and the initial setup completes as quickly as ESX Server can format the virtual fixed disks. During the installation process, you have the option of performing a default or custom install. As with all installs, choose to customize it. Wizards are great, but you want to be in as much control of all installation procedures as possible.

To start, open your Web browser, browse to the ESX Server MUI login page, and log in with an account that has superuser privileges, such as the root user account. Next, select Add Virtual Machine from the bottom-right corner of the window, and follow these steps:

1. On the Add Virtual Machine screen, select Guest Operating System and supply a Display Name option for the VM. In the Location field, remember to create a unique directory for the guest VM and name it after the VM. Take caution not to delete the configuration file's extension.
2. The Virtual Machine Configuration screen gives you the opportunity to select the number of virtual processors to make available to the VM and the amount of memory. If you'll be running Terminal Services for Citrix Metaframe, be sure to select that option under Workloads. This option allows ESX Server to reserve and allocate memory for the VM.
3. On the Virtual Disk selection screen, select Blank, Existing, or System LUN/Disk. Assuming you're creating a simple virtual disk, select Blank. Chapter 14 covers SAN-attached disks.

4. The configuration of the VM's virtual disk ends at the Edit Virtual Disk Configuration screen. When supplying a name for the Image File Name option, name the *.vmdk file after that of the VM; it reduces administrative overhead in the future by making maintenance easier. Set the virtual SCSI node, and select the disk mode you want to use. Persistent mode acts like a typical physical disk.

ESX Server will take a moment to create the virtual disk for the VM. Once the creation process has completed, the MUI displays the VM's configuration via the Hardware tab. Verify each device. If you don't need a device, remove it. If you need additional devices, click Add Device.

Mounting ISO Images

If you have ISO images of your OS installation media, you can mount the ISO by selecting Edit under DVD/CD ROM Drive. For the Device setting, select ISO Image, and specify the ISO directory in Location. As a rule of thumb, the default installation of ESX Server doesn't leave much room for ancillary files such as ISO images to be hanging out. You can work around this by using the mount command with the samba option to connect to network locations and mount them as a local directory. Follow these steps:

1. Connect to the ESX Server console using a terminal session. You can use ssh from within Linux, or you can use connectivity programs, such as PuTTY, to attach Windows OSs.
2. Create a mount point in the /mnt directory for the ISO:

```
mkdir /mnt/iso
```

3. Use the mount command to connect to the remote Server Message Block (SMB) share. The command statement that follows is similar to what you'd need to run. All the parameters are passed to the remote server to allow connectivity. Be careful when sending passwords; they will be in plain text.

```
mount -t smbfs -o username=administrator,workgroup=<domain>,  
password=123456 //RemoteServer/RemoteShare /mnt/ISO
```

4. Test the mount completed successfully by listing the contents of the /mnt/iso directory:

```
ls /mnt/iso
```

5. To install the guest OS, connect to the ESX Server MUI and edit the properties of the CD/DVD-ROM drive. Set the mount location to /mnt/iso/<cdrom.iso>, and install the guest's OS.

If you want, you can put the mount command in a small Perl script to mount the remote location on demand, and you can create a second script to unmount (umount) the remote share when you're finished using the ISO. This is the Perl script to mount the remote share:

```
#!/usr/bin/perl  
use strict;  
` mount -t smbfs -o username=administrator,workgroup=<domain>,  
password=123456 //RemoteServer/RemoteShare /mnt/ISO`;
```

This is the Perl script to unmount the remote share:

```
#!/usr/bin/perl
use strict;
`umount -f //RemoteServer/RemoteShare`;
```

Installing VM Tools for GSX Server and ESX Server VMs

After installing an operating system on your guest VM, you'll want to install VMware Tools. The VMware Tools package supplies significant enhancements to a VM and provides extra functionality between a guest VM and the host system. For instance, adding VMware Tools facilitates drag-and-drop support between a guest and the host, increases graphics performance with an optimized video driver, supports time synchronization between guests and the host, and supports seamless mouse transitions between guest VMs and the host system.

VMware Tools is available for you to install by using ISO images compiled in VMware GSX Server and ESX Server, and it supports a gamut of Microsoft Windows OSs, from Windows 9x through Windows 2003, as well as many Linux distributions. As a technical note, during the installation process, VMware GSX Server and ESX Server temporarily mount the ISO files as the guest's first CD-ROM drive. The procedures to install VMware Tools for Windows and Linux guest VMs are virtually identical across GSX Server and ESX Server.

Using VMware Tools for Windows

To initiate the installation of VMware Tools for a Windows VM, start by selecting VM ► Install VMware Tools from the Server Console menu. You'll first be presented with a preinstallation pop-up box, informing you that the guest VM to receive the tools must be running.

On the Setup Type selection screen, select Custom as your install choice. As a rule of thumb, you really never want an application to automatically make choices for you. Even if you opt for the defaults during a custom install, you'll be aware of the installation points and the services to be loaded in case you need to troubleshoot any problems.

The Custom Setup configuration screen has three main options: Toolbox, VMware Device Drivers, and Shared Folders. If you take a moment to click each option, the Feature Description area will display some details about the option selected. Briefly stated, though, the Toolbox feature improves the usability of a VM by using a group of utilities; the VMware Device Drivers option installs performance-enhancing drivers for the virtual video, mouse, SCSI, and network adapters; and the Shared Folders option facilitates the sharing of files between the guest VM and the host. If this is your first installation, install all the drivers; you'll want to experiment with each. If you're not satisfied with the installation destination of the tools, click the Change button to select a different destination directory.

To install the tools, select Install on the Ready to Install screen. If you have doubt as to any selections previously made, review your choices by clicking the Back button.

The wizard will take a moment to deliver the software packages, and during the delivery process you'll receive several alerts. The alerts state that the VMware Tools drivers haven't passed Windows Logo testing. This is okay; select Continue Anyway on each of the driver alerts. You'll also notice that some of the drivers activate during the install, such as the mouse. You can seamlessly move your mouse between the guest and host windows.

On the Installation Wizard Completed screen, select Finish to complete the install, and you'll be prompted to reboot your guest VM system.

If you don't have VMware tools installed, you'll have to press Ctrl+Alt to release the focus of the mouse from the guest VM and return focus to the host system. The mouse will also underperform in that it will appear to move erratically. Without the video driver, the guest's display will have limited resolution capabilities, and the SCSI drivers simply add better performance characteristics for attached virtual devices. With the tools installed, take a moment to optimize the performance of Windows XP by referring to Chapter 3.

Tip With virtualization application technology storming the IT front, you'll want to find all possible sources of information. VMware's online documentation and community forums are an excellent place to get valuable information about configuring guest VMs. For instance, you can find detailed information on installing VMware Tools for each Windows OS at http://www.vmware.com/support/ws45/doc/new_guest_tools_ws.html.

Using VMware Tools for Linux

The Linux VMware Tools package is built into VMware GSX Server and includes all the same great benefits and ease of installation as Windows Tools. You don't need to download any additional software or use physical media. The easiest way to install the tools is to log into the VM as the root user. If you don't, you'll need to invoke the `su` command with a typical user account. Keep in mind that the example code to follow may not work with your particular distribution; therefore, make adjustments as necessary:

1. To begin the installation, power on the guest VM to receive the packages in text mode; the tools can't be installed if X Windows is running. After the guest VM has completely booted and you've completed the login process, select File ► Install VMware Tools from the VMware Server console menu.
2. You'll now need to mount the virtual CD-ROM and then extract the installer files. Before installing the tools, you should also unmount the CD-ROM, like so:

```
mount /dev/cdrom /mnt/cdrom
cd /tmp
tar xzf /mnt/vmware-linux-tools.tar.gz
umount /mnt/cdrom
```

3. You execute the installer by changing to the package delivery directory and executing a Perl script, like so:

```
cd vmware-tools-distrib
./vmware-install.pl
```

4. Initiate the Linux GUI, startx, and open a terminal session where you can start the VMware Tools background application:

```
vmware-toolbox &
```

If you want the tools to start automatically when the GUI initializes, add the `vmware-toolbox` & command to your system's startup programs. If after adding the tools your guest VM becomes unstable, you can remove the tools by executing `vmware-uninstall-tools.pl` with root privileges.

Configuring VMware GSX Server and ESX Server Virtual Hardware Options

Many similarities and differences exist between VMware GSX Server and ESX Server. This includes differences ranging from GSX Server being a hosted application and ESX Server being a stand-alone product to both virtualization applications utilizing a similar portfolio of virtual hardware options you can install or remove. For performance reasons, you'll find that ESX Server doesn't contain unnecessary frivolities, such as sound and USB support. Like their workstation counterparts, the server products utilize fully configurable BIOSs.

To get the best possible performance out of your guest VMs, don't overallocate guest virtual hardware and undercommit host physical resources. As mentioned in Chapter 4, you want to be frugal with resources but not miserly. For instance, do you really need an audio, USB, or floppy device in a server? Do you necessarily need COM and LPT ports enabled? By identifying your needs and the purpose of the server, you can eliminate overhead and wasted resources by removing unnecessary virtual hardware devices in guest VMs and by disabling components in the BIOS. The net effects of correctly sizing a guest VM are better performance and increased reliability because there's less resource consumption and fewer resources to run awry.

Removing a VM hardware device is straightforward. For GSX Server, while the guest VM is powered down, not suspended, open the MUI, and select Edit Virtual Machine Settings. Highlight the device to be removed, and then select Remove. For ESX Server, connect to the MUI, and click the display name of the VM to be modified. On the Hardware tab, select Remove for the device in question.

For GSX Server, if the VM you're building is lacking hardware, such as an additional virtual NIC or virtual CD-ROM drive, you can add the components by clicking the Add button. ESX Server requires you to click the display name of the guest VM in the MUI and choose Add Device from the Hardware tab. Add Device is located at the bottom of the screen. One reason to add a floppy disk or CD-ROM drive is to use VMware's ability to mount an ISO image.

To avoid excessive redundancy, we'll discuss adding virtual hardware for both GSX and ESX together, and we'll point out significant differences as necessary.

GSX Server hardware, like VMware Workstation, employs the Add Hardware Wizard to help you install virtual hardware devices. If you need assistance with the wizard, please refer to Chapter 4. The ESX Server wizard relies on your ability to know what you're clicking and assumes you have some knowledge of Linux. If you don't know what to do, every hardware device type has a Help button in the lower-left corner of the screen to aid you in the configuration process. Before building VMs for a production environment, you should thoroughly examine each piece of virtual hardware that's made available for guest VMs. Being familiar with the VMware's interface makes administration significantly easier down the road. For each device in GSX Server and ESX Server, we'll highlight its installation procedure.

Hard Disk

Three hard disk choices are presented for GSX Server: Create a New Virtual Disk, Use an Existing Virtual Disk, and Use a Physical Disk. Adding a new hard disk to an existing VM is virtually identical to the process of preparing a drive when creating a guest VM. If you're creating a VM to connect to an existing virtual hard disk, all you have to do is browse to the location of the preexisting virtual disk. Remember that virtual disks will end with a *.vmdk extension.

ESX Server has three choices as well: Blank, Existing, and System LUN/Disk. The Blank and Existing options are similar to GSX Server in configuration, and you can elect to set the Blank Disk Mode option to Persistent, Nonpersistent, Undoable, or Append. Chapter 2 discusses each type if you're unsure of what to select. If you're connecting to a SAN or a raw SCSI disk on an ESX Server, choose System LUN/DISK. Don't forget—we thoroughly cover SANs in Chapter 14. For both GSX Server and ESX Server, remember to make any necessary naming changes for directories and files for the new virtual disk. As always, make sure your naming conventions are consistent and meaningful. If you need to configure the virtual disk to be independent, click the Advanced button during the creation process on the Specify Disk screen.

Physical Disk Creation for GSX Server

Using a physical disk for a GSX Server VM is really the only tricky option during the VM creation process. Generally, using physical disks should be left to advanced users. Conversely, you'll never become an advanced user if you don't try to use physical disks. To that end, don't let the physical disk creation warning scare you away from the performance benefits of using physical disks in your VMs.

On the Physical Disk selection screen, you'll need to choose between using a whole disk or a partition thereon. If you choose to use the entire disk, you'll be asked to provide a virtual disk filename that will store the partition access configuration information. Remember to make any necessary directories to house the file and be sure the filename is meaningful.

If you elect to use a physical partition, you'll be presented with a screen to select which partition on the physical disk to use.

Physical Disk Creation for ESX Server

Using physical disks are a great way to add that extra bump in performance that high-load servers require, such as e-mail or database servers. To add a physical disk to ESX Server, on the Add Hardware page of the guest VM, select Add Device ► Hard Disk ► System LUN/DISK. Select Use Metadata, specify the directory of the *.vmdk file location, and supply the filename. Select SCSI ID from the Virtual SCSI Node list, and then choose Physical Compatibility or Virtual Compatibility. The Physical Compatibility option gives the VM direct access to the disk or LUN, and the Virtual Compatibility option allows you to choose between the different disk modes, such as Persistent, Nonpersistent, Undoable, and Append.

DVD/CD-ROM Drive

When adding a DVD/CD-ROM drive to your guest VM, you have the option of using a physical device or a software image. GSX Server has the ability to autodetect a physical device, or, as with ESX Server, you can specify a device. Additionally, GSX Server can employ legacy emulation; click the Advanced button on the physical drive selection screen. If you use an ISO image, you'll need to specify the location of the ISO file. For GSX Server, you can specify the device node by clicking the Advanced button on the Choose an ISO Image screen.

MOUNTING ISO IMAGES WITH MICROSOFT OPERATING SYSTEMS

Like Linux and guest virtual machines, you can mount ISO images from within Microsoft OSs with the aid of a small utility called Virtual CDrom. It's available for download at http://download.microsoft.com/download/7/b/6/7b6abd84-7841-4978-96f5-bd58df02efa2/winxpvirtualcdcontrolpanel_21.exe. Microsoft doesn't go into any great detail in its documentation except for a small blurb on its MSDN site, <http://msdn.microsoft.com/subscriptions/faq/default.aspx>. Follow these steps:

1. After downloading the compressed file, extract it; then copy the VCdRom.sys file to the %systemroot%\system32\drivers directory.
2. Launch the VCdControlTool executable, and select Driver Control ► Install Driver. Browse to the VCdRom.sys file, and select Open ► Start ► OK.
3. To mount an ISO, select Add Driver ► Mount. Select the ISO to be mounted. If necessary, you have the following options: Suppress UDF, Suppress Joliet, and Persistent Mount.
4. You can now access the ISO like a physical CD drive.

Note You can use network-mapped shares to mount ISO images but not UNC path names. If you'd like to use UNC path names, try downloading (for free) Virtual CloneDrive from <http://www.slysoft.com>.

Floppy Drive

Creating a virtual floppy drive is similar to creating a virtual disk. The wizard will present you with three options: Use a Physical Floppy Drive, Use a Floppy Image, and Create a Blank Floppy Image. There's no real trick to using any of the three options. You just need to decide what's going to suit your needs and then decide if the guest VM will connect to the VM when powering on. ESX Server has two options: System Floppy Drive and Floppy Image. Using floppy images are a great way to keep from having to use sneakernet.

Ethernet Adapter

Adding Ethernet adapters to a VM will also require you to select the network connection type to be used. You have four available options: Bridged, NAT, Host-Only, and Custom. If you're unsure of which type to use, please refer to Chapter 2 where we discuss each network type in detail. ESX Server is making sure you select the correct network connection if you have multiple virtual switches created.

Sound Adapter

When adding a sound adapter to your GSX Server guest VM, the only real issue you'll need to be concerned with is why you think your server needs sound. Short of having audio alarms for intrusion detection systems, there's no real need for sound in a server—it's just one more thing to go wrong and eat up valuable resources. Don't even ask about ESX Server sound support.

Configuring Legacy Devices

Guest VMs for GSX Server and ESX Server can use COM and parallel ports—VMware refers to these as *legacy devices*. ESX Server disables these devices by default for performance reasons and discourages using them for both server products. Unfortunately, newer USB technology isn't supported for ESX Server, but we'll cover how to use it in the "Scripting ESX Server USB Connectivity" section.

Serial Port

Creating a serial port for a GSX VM affords you the ability to not only have access to the host's physical serial port but also to have the ability to output information either to a file or to a named pipe. If you elect to create a virtual port mapping to the host's physical port, you'll need to select which COM port you want to connect to using a drop-down menu. If you decide to send the information of a virtual COM port to a file, you'll need to specify the name of the file and the storage location. The named pipes option asks you to further configure the relationship of the named pipe. The named pipes setting allows you to directly connect two VMs; in addition, it affords you the ability to create a connection between an application on the host VM and the guest VM. It's important to remember that the pipe name is identical on the client and server and takes the form of `\\.\pipe\<pipe name>`.

To enable serial port support on ESX Server (just in case you have UPS requiring COM connectivity), ensure that the host's BIOS has the serial port enabled and the guest VM is in a powered-off state. Edit the guest VM's *.vmx configuration file, and add serial port support, like so:

```
serial0.startConnected = true
serial0.present = true
serial0.file = device
serial0.fileName = "/dev/ttyS0"
```

When the VM boots, the serial port will be automatically connected.

Parallel Port

Adding a parallel port to a GSX Server guest VM is similar to adding a serial port. You'll need to select whether you want to use the host's physical port or use VMware's ability to write to a file. If you have multiple parallel ports, the wizard will allow you to specify which port you want to use. If the new virtual parallel port is to write to a file, you'll need to select the appropriate file output destination.

You may find it necessary to enable parallel port connectivity so that the ESX Server guest VM can use a software security dongle. To enable parallel support on ESX Server, ensure that the host's parallel port is set to PS/2 or Bidirectional in the BIOS. In the VM's *.vmx configuration file, make sure the virtual hardware version is `config.version = 6`. Then, follow these steps:

1. At the CLI of the ESX Server, load the parallel port modules:

```
insmod parport
insmod parport_pc
insmod ppdev
```

2. Modify the VM's *.vmx configuration file to add parallel port support:

```
parallel0.present = true
parallel0.fileName = "/dev/parport0".
parallel0.bidirectional = true.
```

3. Launch the VM and edit its BIOS by selecting F2. Set the parallel port mode to Bidirectional.

Note Only one OS and one VM can use the parallel port at any given time. If you don't want to load ESX Server modules on each host reboot, add the module commands to the `rc.local` file.

Configuring Generic SCSI Devices

To use SCSI peripherals, such as a tape device or scanner, you'll need to add a generic SCSI device to a guest VM. This requires you to select the physical SCSI device from the Device Connection list and then specify a virtual device node to use with the generic SCSI device.

Tip You can find many excellent white papers, notes, and compatibility guides dealing with everything from generic SCSI to configuring blade servers on VMware's Web site: http://www.vmware.com/support/resources/esx_resources.html.

If you're connecting to a tape library, be sure to select every physical tape device, or the library won't appear correctly in the guest VM. You can have the guest VM connect to the physical device at power on. To do this, select the Power On option under Device Status.

Note When editing a VM to add or remove virtual devices, be sure to click the OK button on the Virtual Machine Settings dialog box. If not, your settings will often not take effect.

Configuring a USB Controller

For GSX Server, adding USB support requires you to decide if you want the guest VM to automatically attach to a USB device when it has focus on the host. If you desire this behavior, check the box under Device Status, and that's it.

On ESX Server, USB support is disabled on the console OS because of potential chipset incompatibilities. These incompatibilities are created by the USB controller sharing interrupts with other controller types and the I/O Advanced Programmable Interrupt Controller (IOAPIC) forwarding masked interrupts. Disabling USB support gives the VMkernel control over interrupts. Because it's becoming increasingly important for servers to have access to USB support for application dongles, external hard drives, and other peripherals, you'll learn how to enable USB support in this section.

Note VMware fully documents disabling USB support in ESX Server in its knowledge base under Answer ID 1326. If you're using an older version of ESX Server, 2.1.2 and earlier, you'll definitely want to look at the article. If you're unsure of your version, you can execute the `issue` command.

To enable USB support on ESX Server, you'll need to make sure VMkernel manages the host's PCI devices and that the kernel loads before any USB drivers. You'll need to reassign any controllers dedicated to the console as shared devices. In general, most PCI devices can be assigned to one of three categories: dedicated to VMkernel, dedicated to the Service Console, or shared between both. To set the device category, you'll need to use the `vmkpcidivv` command with the `s` option to specify the sharing of devices. You should execute the `vmkpcidivv` command-line utility as root; it's used to modify memory and device allocation on ESX Server. The following is its usage, and Table 6-1 shows its options:

```
/usr/sbin/vmkpcidivv [-i] [-l name] [-m memsize] [-q] [-v bus:slot:fcn]
```

Table 6-1. *vmkpcidivv Options*

Option	Action Performed
-i	Interactively assists with the “divvy” process
-l	Performs operations on given label
-m	Allocates memory, in megabytes, to given guest OS
-q	Runs a query: <code>vmkmod</code> , <code>vmkdump_part</code> , or <code>vmkdump_dev</code> labels <code>vmfs_part</code>
-v	Assigns device to VMs

To get an idea of how the command works, run `vmkpcidivv -i`. The command will take you through any existing configuration files and lets you make changes. If you make mistakes during the process, you have the opportunity at the end of the configuration update to not commit the changes. You can configure several items with the command:

- The name of the configuration
- Memory allocation to the Service Console
- SCSI controllers and network adapters

After experimenting with the `vmkpcidivv` command, run it again. This time, reassign all controllers dedicated to the Service Console as shared. In general, shared controller interrupts are VMkernel managed and then forwarded to the Service Console on an as-needed basis. You should leave VMkernel-dedicated controllers alone.

The second half of the procedure for enabling USB support for ESX Server requires that you delay the loading of USB drives by editing the `/etc/modules.conf` file. If you're unsure of editing procedures in ESX Server, please refer to Chapter 2, which explains the Vi editor. When editing files, don't forget to save your work.

Note Though not supported by VMware, you can try using USB drives to run guest VMs. Being that USB devices are seen as SCSI devices in Linux, you can try loading the USB device before the VMkernel loads to make the disk available to guest VMs.

Follow these steps:

1. Mark every line in `modules.conf` containing USB controller references, such as Universal Host Controller Interface (`usb-uhci`), Open Host Controller Interface (`usb-ohci`), and Enhanced Host Controller Interface (`usb-ehci`). UHCI is generally used to support the Intel PIIX4 and VIA chipsets, and OHCI generally supports Compaq, iMacs, OPTi, and SiS. For instance, at the beginning of every line containing a reference to `usb-ohci`, `usb-uhci`, or `usb-ehci`, place a pound symbol (`#`). Make a note of each line; you'll need it for step 3. For example, if `modules.conf` has a line containing the alias `usb-controller usb-ohci`, change it to look like the following:

```
#alias usb-controller usb-ohci
```

2. Create the given file in the specified directory, `/etc/rc3.d/S91usb`. If you don't want the USB OHCI module activated on boot, don't create the file. You can load the module using the `modprobe` command at any time.
3. For each line you marked in the `modules.conf` file, you'll need to create a similar line in the `S91usb` file preceded by the `modprobe` command. In our example, the line reads as follows:

```
modprobe usb-ohci
```

Note USB devices aren't supported within ESX Server guest VMs, and VMware briefly explains this in Answer ID 1015 in its knowledge base. You'll be able to use USB devices only from within the console OS. If you need to make USB devices available to guests, you can try using USB-to-IP technology, such as Digi AnywhereUSB. If you're a bit more adventurous, you can look into the USB/IP Project at <http://usbip.naist.jp>.

4. Reboot ESX Server. On reboot, the `/etc/rc3.d/S90vmware` file will run before the `/etc/rc3.d/S91usb` file. This ensures that the VMkernel is ready before the USB drivers are loaded.
5. Execute `cat /proc/interrupts`, and verify that an interrupt is assigned to `usb-ohci`.
6. Run `lsmod|grep -i usb`. You should see two USB modules, `usb-ohci` and `usbcore`.
7. Mount your USB device.

You may be asking yourself what you can do with a console-only USB device, but it really has some value. For instance, you can mount a USB hard drive (or CD-ROM drive) at the console and make it available to guest VMs by using WinSCP3 to copy the files back and forth.

To mount a USB CD-ROM drive, complete the following steps:

1. Run the `mount` command to see what's already in use from the `mtab` file. If you want to see what's in `mtab`, execute `cat /etc/mtab`. Run `fdisk -l` to see the system partitions. Use the results from all commands to avoid creating duplicate names.
2. Create a directory to mount the USB CD-ROM drive:

```
mkdir /mnt/usb/cdrom
```

3. Plug in and power on your USB device:
4. Verify that the correct module is loaded for your device. Not all CD-ROM drives are supported. Run `tail -n50 /var/log/messages`, and pay particular attention to the last few lines of the output. The log messages will tell you if the requisite supporting modules were loaded and will tell you the device ID.

```
Jan 14 16:15:39 VA1ESX2 kernel: scsi4 :➡
SCSI emulation for USB Mass Storage devices
Jan 14 16:15:39 VA1ESX2 kernel: Vendor: ➡
SONY Model: DVD RW DRU-710A Rev: BY01
Jan 14 16:15:39 VA1ESX2 kernel: Type: ➡
CD-ROM ANSI SCSI revision: 02
Jan 14 16:15:39 VA1ESX2 kernel: ➡
Attached scsi CD-ROM➡
sr0 at scsi4, channel 0, id 0, lun 0
Jan 14 16:15:39 VA1ESX2 kernel: ➡
scsi_register_host starting finish
Jan 14 16:15:54 VA1ESX2 kernel: ➡
```

```

sr0: scsi-1 drive
Jan 14 16:15:54 VA1ESX2 kernel: ➡
scsi_register_host done with finish
Jan 14 16:15:54 VA1ESX2 /etc/hotplug/usb.agent: ➡
no drivers for USB product 54c/1f6/1
Jan 14 16:24:58 VA1ESX2 kernel: Device not ready. ➡
Make sure there is a disc in the drive.
Jan 14 16:25:49 VA1ESX2 sshd(pam_unix)[4401]: ➡
session opened for user root by (uid=0)

```

5. If you look through the output and stop at the fourth line, you'll find that the CD-ROM drive is listed as device `sr0` on `scsi4`. Despite that the device is listed as `sr0`, there's no `sr0` device in `/dev`. You'll need to use `scd0` to mount the USB CD-ROM drive. Further down, the output reveals that a driver isn't loaded. Load the `usbserial` module to use the CD device:

```
insmod usbserial
```

6. You can get additional information about your device by executing `cat /proc/scsi/scsi`. You should have output similar to the following:

```

Host: scsi2 Channel: 00 Id: 00 Lun: 00
Vendor: DELL Model: PERCRAID RAID5 Rev: V1.0
Type: Direct-Access ANSI SCSI revision: 02
Host: scsi4 Channel: 00 Id: 00 Lun: 00
Vendor: SONY Model: DVD RW DRU-710A Rev: BY01
Type: CD-ROM ANSI SCSI revision: 02

```

Note If you want to verify that ESX Server correctly found the USB device, you can visit the Web site at <http://www.linux-usb.org/usb.ids> and look up the device's ID from the information found in the messages file. You'll also find excellent support information on the site.

7. To mount CD-ROM device serial 0, execute `mount -r -t iso9660 /dev/srcd0/mnt/usb/cdrom`. Your media should now be accessible from `/mnt/usb/cdrom`.

8. List the contents of the drive:

```
ls /mnt/usb/cdrom
```

9. When you're done using the device, be sure to use the `umount` command to unmount the USB device before unplugging it. If you need help with the `umount` command, use the manual pages, `man umount`.

```
umount -f /dev/srcd0
```

If you want a device to be mounted each time the system is rebooted, place startup commands in the `rc.local` file. Add the mount commands to be executed to the end of the file.

Assuming you make the packages available to ESX Server, you can even burn disks. If you have access to the Red Hat 7.3 installation CDs, you can install the listed RPMs (and any dependencies) for CD-burning purposes. Remember that using USB under ESX Server isn't supported by VMware and should be used only for experimental and temporary/emergency purposes. Here's how to do it:

```
cdda2wav-2.0-11
cdrecord-2.0-11
cdrecord-devel-2.0-11.
dvd+rw-tools-5.17.4.8.6-0
dvdrecord-0.1.4-4
mkisofs-2.0-11
xcdroast-0.98a14-3
```

Note If you need help burning data to a CD-ROM, visit Linux Headquarters to brush up on the basics: <http://www.linuxheadquarters.com/howto/basic/cdrecord.shtml>.

You can also make USB thumb drives and external USB hard drives available to the ESX Server console by performing these steps:

1. Run the `mount` command to see what's already in use from the `mtab` file. If you want to see what's in `mtab`, execute `cat /etc/mtab`. Execute `fdisk -l` to view system partitions. Use the results from all commands to avoid duplicate names.

2. Create a directory to mount the USB thumb drive:

```
mkdir /mnt/usb/thumbdrive
```

3. Plug in your USB drive.

4. Verify that the correct modules are loaded for your device by checking the `messages` file. Pay particular attention to the last few lines. A SanDisk drive, for example, would have the ID (`vend/prod 0x781/0x5150`). The `messages` file will tell you if the requisite supporting modules were loaded:

```
tail -n50 /var/log/messages
```

5. If you see error messages, you'll need to manually load the modules for your device. You can find the USB modules available with a standard ESX Server by executing `find / -name *usb*.o`:

```
/lib/modules/2.4.9-34/kernel/drivers/bluetooth/hci_usb.o
/lib/modules/2.4.9-34/kernel/drivers/media/video/cpia_usb.o
/lib/modules/2.4.9-34/kernel/drivers/net/irda/irda-usb.o
/lib/modules/2.4.9-34/kernel/drivers/usb/ov511/i2c-algo-usb.o
/lib/modules/2.4.9-34/kernel/drivers/usb/dabusb.o
/lib/modules/2.4.9-34/kernel/drivers/usb/hpusbscsi.o
/lib/modules/2.4.9-34/kernel/drivers/usb/serial/usbserial.o
```

```

/lib/modules/2.4.9-34/kernel/drivers/usb/storage/usb-storage.o
/lib/modules/2.4.9-34/kernel/drivers/usb/usb-ohci.o
/lib/modules/2.4.9-34/kernel/drivers/usb/usb-uhci.o
/lib/modules/2.4.9-34/kernel/drivers/usb/usbcore.o
/lib/modules/2.4.9-34/kernel/drivers/usb/usbnet.o
/lib/modules/2.4.9-34/kernel/drivers/usb/usbvideo.o
/lib/modules/2.4.9-vmnix2/kernel/drivers/usb/dabusb.o
/lib/modules/2.4.9-vmnix2/kernel/drivers/usb/serial/usbserial.o
/lib/modules/2.4.9-vmnix2/kernel/drivers/usb/storage/usb-storage.o
/lib/modules/2.4.9-vmnix2/kernel/drivers/usb/usb-ohci.o
/lib/modules/2.4.9-vmnix2/kernel/drivers/usb/usb-uhci.o
/lib/modules/2.4.9-vmnix2/kernel/drivers/usb/usbcore.o

```

6. To load the correct module for a USB thumb drive, enter the following at the command line:

```
modprobe usb-storage
```

7. You can verify the successful attachment of your USB drive to the system by executing `cat /proc/scsi/scsi`. (Also, take a moment to rerun `cat /etc/mtab`). The output should look similar to this:

```

Attached devices:
Host: scsi2 Channel: 00 Id: 00 Lun: 00
Vendor: DELL Model: PERCRAID RAID5 Rev: V1.0
Type: Direct-Access ANSI SCSI revision: 02
Host: scsi3 Channel: 00 Id: 00 Lun: 00
Vendor: SanDisk Model: Cruzer Mini Rev: 0.2
Type: Direct-Access ANSI SCSI revision: 02

```

8. Notice that the thumb drive is on scsi 3. To bring this together a bit more, execute `tail -n50 /var/log/messages`:

```

Jan 14 15:07:38 A1ESX2 sshd(pam_unix)[3938]:
session opened for user root by (uid=0)
Jan 14 15:17:57 A1ESX2 kernel:
Initializing USB Mass Storage driver...
Jan 14 15:17:57 A1ESX2 kernel: usb.c:
registered new driver usb-storage
Jan 14 15:17:57 A1ESX2 kernel: scsi3 :
SCSI emulation for USB Mass Storage devices
Jan 14 15:17:57 A1ESX2 kernel: Vendor:
SanDisk Model: Cruzer Mini Rev: 0.2
Jan 14 15:17:57 A1ESX2 kernel: Type:
Direct-Access ANSI SCSI revision: 02
Jan 14 15:17:57 A1ESX2 kernel: VMWARE SCSI Id:
Supported VPD pages for sdb : 0x1f 0x0
Jan 14 15:17:57 A1ESX2 kernel:
VMWARE SCSI Id: Could not get disk id for sdb
Jan 14 15:17:57 A1ESX2 kernel: :VMWARE: Unique Device

```

```

attached as scsi disk sdb at scsi3, channel 0, id 0, lun 0
Jan 14 15:17:57 A1ESX2 kernel:
Attached scsi removable disk sdb at scsi3, channel 0, id 0, lun 0
Jan 14 15:17:57 A1ESX2 kernel:
scsi_register_host starting finish
Jan 14 15:17:57 A1ESX2 kernel:
SCSI device sdb: 1000944 512-byte hdwr sectors (488 MB)
Jan 14 15:17:57 A1ESX2 kernel:
sdb: Write Protect is off
Jan 14 15:17:57 A1ESX2 kernel:
sdb: sdb1
Jan 14 15:17:57 A1ESX2 kernel:
scsi_register_host done with finish
Jan 14 15:17:57 VA1ESX2 kernel:
USB Mass Storage support registered.

```

9. The USB thumb drive should be the first unused SCSI device on your system; in most cases it will be `/dev/sda1`. Being that ESX Server is already using devices `sda1` through `sda1`, you can eliminate the default settings as the mount point. However, look at the third-to-the-last line from the messages output. The thumb drive is listed as `sdb1`. Mount the thumb drive by executing the following command:

```
mount /dev/sdb1 /mnt/usb/thumbdrive
```

10. List the contents of the drive:

```
ls /mnt/usb/thumbdrive
```

11. When you're done using the device, be sure to use the `umount` command to unmount the USB device before unplugging it. If you need help with the `umount` command, use the manual pages, `man umount`. To have devices mounted each time the system reboots, place mount commands in the `rc.local` file at the end of the file:

```
umount -f /dev/sdb1
```

Scripting ESX Server USB Connectivity

You can further simplify the management of USB devices by using several small Perl scripts. Placing the requisite USB commands in a Perl script will allow you to connect or disconnect devices on demand. You'll have to use a text editor, such as Vi (discussed in Chapter 5), and have a grasp on the Perl scripting basics. We'll continue showing how to work with a USB thumb drive in this example.

You really need to know only a few things to be dangerous with Perl:

- Perl stands for Practical Extraction and Reporting Language and was created by Larry Wall in 1987.
- Scripts should end with a `.pl` extension.
- The first line of the script begins with the *shebang* line, which points to the Perl binary, `#!/usr/bin/perl -w`. You can find the path by executing `which perl` at the CLI.

- All statements end in a semicolon (;).
- Comments begin with pound sign (#).
- Use a backtick (`) to invoke using shell commands.
- You must set permissions on the script to execute, `chmod +x scriptname.pl`.
- Verify that permissions are correctly set to 755, `ls -l scriptname.pl`.
- You can execute scripts by calling the script name at the CLI. If necessary, include the current directory by preceding the script name with a `./`, as in `./scriptname.pl`.

Writing a script to mount a USB thumb drive is just a matter of adding the correct ESX Server commands to the file and setting permissions. If you aren't up to speed on the best practices for error checking and control, then you may want to get a good Apress book on Perl, such as *Beginning Perl*, Second Edition, by James Lee (Apress, 2004). The scripts in this section invoke all requisite Linux modules to connect and disconnect from the USB bus without modifying the `modules.conf` file.

Follow these steps:

1. Write the script using Vi:

```
#!/usr/bin/perl
#Script to mount USB thumb drive.
use warnings;
use strict;
`modprobe usb-ohci`;
`modprobe usbcore`;
`modprobe usb-storage`;
`mount /dev/sdb1 /mnt/thumbdrive`;
`ls -l /mnt/thumbdrive`;
```

2. Change the permissions of the script:

```
chmod +x mountusb.pl
```

3. Verify the permissions:

```
ls -l mountusb.pl
```

4. Execute the script:

```
./mountusb.pl
```

When you're done using your USB device, you can create another script to unmount the device and unload the USB modules. Follow these steps:

1. Write the script:

```
#!/usr/bin/perl
#Script to unmount USB drive.
use warnings;
```



```
use strict;
`umount /dev/sdb1`;
`rmmod -r usb-storage`;
`rmmod -r usbcore`;
`rmmod -r usb-ohci`;
`ls -l /mnt/thumbdrive`;
```

2. Change the permissions of the script:

```
chmod +x umountusb.pl
```

3. Verify the script permissions:

```
ls -l umountusbcd.pl
```

4. Execute the script:

```
./umountusb.pl
```

Building Microsoft Virtual Server VMs

The administration of Microsoft Virtual Server is different from Virtual PC in that it's performed through the Administration Web site; in this sense, Virtual Server is a lot like ESX Server.

Creating a Virtual Server VM is an exercise you're familiar with when it comes to most Microsoft software installs—it's a straight shot from beginning to end. When creating a guest VM, keep in mind all the best-practice information you learned from previous chapters, such as creating a single directory to hold all VM files, naming all files after the VM, and using the appropriate disk mode for the guest.

Follow these steps:

1. To begin the creation, launch the Administration Web site by selecting Start ► Programs ► Microsoft Virtual Server ► Virtual Server Administration Web Site.
2. Under Virtual Machines, select Create. In the Virtual Server Create window, you'll need to supply all the information it takes to make the guest VM.
3. For the VM name, be sure to stick with good naming conventions. If you're using good naming practices, you may want to add a prefix or suffix to the name to note that it's a virtual machine. Oftentimes, in networks with large quantities of physical and virtual servers, it can be difficult to differentiate between the two.
4. Specifying the total amount of memory used in a VM can be tricky. Basically, you want to allocate enough memory for the VM to be able to do 100 percent of its job without overcommitting resources or straining the VM. To that end, give the VM the minimum required by the guest OS. You can always add more memory later to alleviate performance problems. Many times, you'll find that a server can skate by on the minimum, and others will have to be bumped up to industry best practices.
5. Three options are available for a new guest VM in the hard disk category: Create a New Virtual Hard Disk, Use an Existing Virtual Hard Disk, and Attach a Virtual Hard Disk

Later. If you choose to create a new hard disk during the creation of a VM, you'll be stuck with a dynamically expanding disk. This may be fine for a test or lab environment, but for a production environment, dynamic disk performance will prove unsatisfactory. Therefore, if you're creating a production VM, choose Attach a Virtual Hard Disk Later.

6. Network settings for the VM can be set to Not Connected, Bridged to the External Network, or Internal Network. If the VM is to be a network resource, choose External Network. Otherwise, choose Internal Network for a host-only accessible VM.
7. The Virtual Machine Additions area is just a reminder to install VM Additions. We'll discuss how to install these a bit later in the "Virtual Machine Additions" section.
8. After creating the VM, you'll be presented with the Status and Configuration window. You now need to create a virtual hard disk for the VM.
9. From the far left column under Virtual Disks, select Create ► Fixed Sized Virtual Disk. You have the option of creating other disks with the Differencing, Linked, and Floppy options. The Fixed Size Virtual Hard Disk window appears where you'll need to supply the storage directory for the disk and the size. For the storage location, be sure to select the directory you created to store all the VM files and create a directory for the disk. After selecting Create, it will take a moment for the disk to be created, and it will take even longer for large disks.
10. Select the name of the guest VM in the Status area and click Turn On. Install the guest OS as you normally would.

To add virtual hardware or to reconfigure an existing VM, select Virtual Machines v Configure ► *<VM_to_Modify>*. The Configuration portion of the window lists the VM's available hardware choices. We'll highlight the installation process for each option.

General Properties

The General properties page has the setting for the VM's display name and the account settings the VM runs under. If you choose to run the VM under a user account, Virtual Server grants the ability to set the power-on status of a VM when Virtual Server starts, such as automatically turning on the VM. Another option, Action When Virtual Server stops, allows you to set the status of the VM when Virtual Server stops to these options: Save State, Turn Off Virtual Machine, and Shut Down Guest OS. A notes section lets you list helpful information about the machine that helps you manage the VM.

Virtual Machine Additions

Microsoft's Virtual Machine Additions package increases the performance and flexibility of guest VMs running on Virtual Server and are installed after the guest OS is up and running. Installing the tools is a two-part process: first, the tools need to be made available to the guest by selecting Install Virtual Machine Additions; second, the Virtual Machines executable needs to be launched from within the VM. The tools are available to the VM via an ISO. If you have autorun enabled, the installation will start automatically. If you were diligent in optimizing

performance and disabled autorun, you can find the executable (setup.exe) on the system's first virtual CD-ROM drive. The installation takes only a moment and a couple clicks.

You can programmatically install Virtual Machine Additions using an unattended mode with setup.exe and its command-line options. Table 6-2 lists the command-line options you can use during an unattended install when using setup.exe. You should use the command in the format `setup.exe -s -v"/qn [Reboot=ReallySuppress]`". In addition, you can install the software by activating the executable from within Windows Explorer and using the wizard. You can download Virtual Machine Additions from Microsoft's Web site at <http://www.microsoft.com/downloads>.

Table 6-2. *Virtual Machine Additions*

Option	Action Performed
-s	Runs setup program without a GUI
-v	Passes options to Msiexec.exe
/qn	Runs Msiexec.exe without a GUI
Reboot=ReallySuppress	Prevents reboot after installation

If a VM created with Virtual PC is moved to Virtual Server, be sure to reinstall Virtual Machine Additions. Reinstalling the software ensures that the moved guest VM is using the server version of Virtual Machine Additions and avoids any conflicts related to versioning. The enhancements you can expect to experience from installing Virtual Machine Additions include the seamless use of the mouse from the guest window to the host, increased performance, host-to-guest time synchronization, and optimized video drivers.

Memory

The Memory option allows you to increase or decrease the amount of available RAM by using a slide control or by entering the memory size. You can make adjustments in 1MB chunks.

Hard Disks

The Hard Disks Properties option directs Virtual Server to the location of the guest VM's hard drive or adds a hard drive. Within this option, you also have the ability to set virtual disks as undo disks. Being that the Hard Disks Properties dialog box is used only to point to *.vhd files, refer to the "VMware Virtual Disk Drive Geometry" sidebar. If you're creating a new virtual disk, you have the option of creating a dynamically expanding, fixed, differencing, linked, or floppy disk. After creating a disk, you must use the absolute path and the filename.

CD/DVD

In addition to being able to use the host's CD/DVD drives, you can also specify ISO images for a VM to mount. Like virtual hard disks, for ISO images, you need to specify the absolute path to the image. You can add as many (up to four) CD/DVD drives as there are free IDE channels. If you want to add drives to mount ISO images, you can use Virtual CDrom; refer to the "VMware Virtual Disk Drive Geometry" sidebar that covers this tool.

SCSI Adapters

The SCSI Adapter Settings screen adds, removes, and sets the properties of SCSI adapters for virtual hard disks. Short of being able to change the adapter ID, there's not much more to it. If VMs are going to be clustered, you'll need to check the option to share the adapter; it's disabled by default. The virtual SCSI bus is shared to facilitate the quorum needed that provides the software failover support between the clustered VMs. For more information on clustering, refer to Chapter 9.

Network Adapters

The Network Adapters screen provides an interface to add or remove NICs. When creating an additional NIC, you can set it to be connected to an existing external network, internal (host-only) network, or not connected. If you like, you can let Virtual Server generate a MAC for the VM, or you can specify your own.

Scripts

The Scripts Properties dialog box is where you can add scripts that take action based on the current status of the VM in question. By default, scripts are disabled. To enable scripts, select Virtual Server Scripts from the upper-right corner of this screen. You can enable scripts on the host or on VMs attached to the host. For the host, you can supply scripts that process command-line actions when the system starts and stops. This is useful for shutting down VMs and initializing them. The guest VM scripts generally run when a VM changes power states, gets low on disk space, or receives physical hardware errors.

Note Several handy scripts for Virtual Server are available for download from Microsoft's Script Center Script Repository at <http://www.microsoft.com/technet/scriptcenter/scripts/vs/default.msp>. Scripts are available for managing disks, networks, tasks, and VMs. This site is well worth checking out if you don't want to reinvent the wheel!

Floppy Drives

For the virtual floppy drive, you can use the host's floppy drive or an ISO image. If your system doesn't need a floppy drive, disable it for performance reasons. Like virtual hard disks and CD/DVD drives, for ISO images, you need to specify the absolute path of the file.

COM Ports

Any given VM can have up to two active COM ports. If you don't need COM ports, leave them disabled. For the COM port, it can be the host's physical port, a text file, or a named pipe. One valuable use for named pipes with the COM port is for debugging Virtual Server. To debug a VM, follow these steps:

1. Download and install the kernel debugger from Microsoft's Web site at <http://www.microsoft.com/whdc/devtools/debugging/installx86.msp>.
2. Edit the `boot.ini` file of the guest VM, and append `/fastdetect/DEBUG /DEBUGPORT=COM1/BAUDRATE=115200` to the OS boot statement under `[operating systems]`. If you want, you can copy the existing statement and then append the debug line.
3. Configure COM1 of the guest VM to use a named pipe: `\\.\pipe\com1`. Don't check the Wait for Modem option.
4. Connect Microsoft's kernel debugger to the COM port by executing `kd -k com:port=\\.\pipe\com1,pipe,reset=0,reconnect` from the CLI.

LPT Ports

Allowing a guest VM to connect to the host's parallel port is a matter of changing the default option of disabled to LPT1. Once again, if you don't need the support of legacy devices, such as parallel ports, don't enable them on guest VMs.

Managing Server-Class VMs

Managing VMs in a production or test environment is no different than doing the same in traditional physical systems; you'll need to continue to adhere to the same best practices you probably already use, such as backing up, installing antivirus software, monitoring log files, renaming VMs, and performing routine maintenance. In the following sections, we'll discuss backing up, copying, and moving VMs. In addition, you'll further examine VM configuration files, which you started looking at in Chapter 4.

Modifying VM Configurations: Renaming and Moving

Though we thoroughly cover backup strategies for VMs in Chapter 7, we'll briefly introduce you to backing up VMs and their configuration files in this chapter. In addition, you'll look at modifying VM configuration files.

When you first begin experimenting with VMs, you'll soon realize at some point it becomes necessary to rename a guest VM, its configuration file, and its virtual disk file. You'll generally find yourself in this position when you're forced to move a VM from a test environment to a production environment. To ensure the integrity of the VM and a smooth transition, you'll first need to back up your VMs virtual disk and configuration files, rename them as necessary, and then move them to the new host. Whether you're renaming a Microsoft Virtual Server or VMware GSX/ESX Server VM, you can complete the procedure in a few steps.

Renaming a VMware GSX Server VM

Prepping VMware GSX Server to accept a new name for a guest VM is a straightforward process. The VM must be powered down and not in a suspended state before proceeding with the renaming procedure:

1. Remove the VM from the console inventory by selecting the Management tab of the VM to be removed. Right-click the tab, and select Remove from Favorites, which won't delete any files related to the VM to be renamed. Close the GSX Server interface.
2. Locate the VM directory containing the guest's VM configuration file and virtual hard disk file(s). If you used the default installation directory and named the installation directory after your VM, the path to your VM will be similar to C:\Documents and Settings\user\My Documents\My Virtual Machines\<VM_Directory>. Rename the directory and the *.vmx file to exhibit the new name.
3. Though you could simply rename the virtual disk *.vmdk file as well, you should use the vmware-vdiskmanager command. Using this command renames the file but also correctly sets the name of the VM within the virtual disk file's Extent Description area. Here's the command:

```
vmware-vdiskmanager -n old_virtual_disk_name.vmdk ➡
new_virtual_disk_name.vmdk
```

The typical header of a *.vmdk file follows. If you manually change the name of virtual disk files, the disk will retain its old name in the Extent Description area.

```
# Disk DescriptorFile
version=1
CID=fffffffe
parentCID=fffffffe
createType="monolithicFlat"
# Extent description
RW 8388608 FLAT "test-flat.vmdk" 0
# The Disk Data Base
#DDB
ddb.virtualHWVersion = "3"
ddb.geometry.cylinders = "522"
ddb.geometry.heads = "255"
ddb.geometry.sectors = "63"
ddb.adapterType = "buslogic"
```

4. Using a text editor, edit the *.vmx file to reflect the name change. You'll need to modify two lines: the virtual disk configuration line and the display name line. For example, a change may look similar to scsi0:0.fileName = "<New_Name.vmdk>" and displayName = "<New_Name>".
5. Start GSX Server, and open the renamed VM by selecting File ➤ Open Virtual Machine. You'll be asked if you'd like to generate a new UUID. If you need to maintain the MAC address for your VM, select Keep the Existing Identifier; otherwise, select Create New Identifier.

Copying and Moving VMware GSX Server VMs to Other Hosts

Copying and moving GSX Server VMs is virtually identical to doing the same for VMware Workstation. Before beginning the copy or move process, make sure you have a good backup of the guest VM in question. If something runs awry, you'll want to quickly recover to make a second attempt.

After confirming you have a good backup, ensure your user account has sufficient privileges to complete all tasks. Next, you'll need to complete a few preliminary tasks before moving the guest VM:

1. Power down the guest VM to be moved. Make sure it's *not* in a suspended state; moving a suspended VM is a recipe for disaster.
2. Close the VMware GSX Server application.
3. Ensure a transfer medium exists for moving the VM's files. Assuming your VM can fit on a CD or DVD, you can use this type of medium, or you can use a mutually accessible network share. If the guest is going to be placed on a new network host, the easiest way to move the files is over your existing network, going point to point. With the preliminary procedures checked off the to-do list, you can safely move the guest VM.

Note The less a VM is shuttled across a network, the less likely you'll introduce errors into a VM's configuration files and virtual hard disk.

4. Ensure that the new host system has sufficient resources for the added load of the VM to be moved and is not running.
5. Create a directory on the new host for the guest's files. (If you're just moving the location of the guest VM to a different directory on the current host, simply create the new directory in your desired location and locate the VM's files within the new directory.) Remember to create directories with meaningful names for your VMs and then store all related configuration information in the VM's directory. Many administrators name the directories after the VM itself.
6. Make sure the guest VM is powered down. Next, you'll need to find *all* the files on the current host and copy them to the new host in the directory you just created. Files you'll minimally be concerned with are as follows:
 - The configuration file, *.vmx
 - The virtual disk files, *.vmdk
7. Start VMware GSX Server, select File ► Open, browse to the VM's configuration file (*.vmx) in the directory you created, and then select Open. Take a moment to check the VM Settings editor to verify that *every* device is properly configured and is pointing to the correct file location.

Note When you power on a VM moved from its original installation directory, you may be asked if you want to regenerate the UUIDs for the VM. If the VM is a copy, select Yes. If the VM has been moved, select No. Regenerating the UUID changes the MAC address of the guest VM.

8. Check to make sure the VM name, configuration file, and redo log file are all correctly set. You can find this on the Options tab of the VM Settings editor.
9. Select Power ► Power On. If your guest VM fails to properly initialize, make sure you copied every file from the source host, and then double-check that all settings are pointing to the correct locations. If you need to rename your VM, refer to the “Renaming a VMware GSX Server VM” section.

Using VMware Universally Unique Identifiers (UUIDs)

When you move a VM, you'll want to concern yourself with the system's UUID. VMware uses the UUID to generate a MAC address for a VM. The UUID is based on the host's SMBIOS UUID and on the installation of the VMware application and is specifically hooked to the *.vmx file's home directory location the first time the guest VM is booted.

Note SMBIOS presents motherboard management and diagnostic information to a computer system and its software. You can read more about the SMBIOS specification at <http://www.dmtf.org/standards/smbios>.

If you move the location of the guest, VMware will know that the initial install directory has changed because on each boot it compares the value of the UUID.LOCATION entry in the configuration file to the value of UUID.BIOS. If the directory has changed, the UUID is changed, which in turn changes the system's MAC address. Typically, this may not be a problem, but if you're using route maps based on MAC addresses, you may soon find that communications cease. In addition, if you use software that uses SMBIOS (UUID.LOCATION) to identify servers, communications will get interrupted. Lastly, if you're issuing IP addresses based on MAC address, your guest VMs won't be able to communicate if the UUID changes. You can find the UUID settings for any given VM in its configuration file. They're generally nested within the Ethernet settings. The following bit of code is from a VM's configuration file. Notice that each 128-bit value matches for uuid.location and uuid.bios.

```
Ethernet0.addressType = "generated"
uuid.location = "56 4d 7e 42 db dc 5c fb-e9 6f d3 cd ef 07 47 da"
uuid.bios = "56 4d 7e 42 db dc 5c fb-e9 6f d3 cd ef 07 47 da"
```

Foreseeing this issue, VMware configures its applications to ask you if you want to keep the UUID after a virtual machine has been moved. What's the right choice for you? The answer is, it depends. VMware suggests that if the VM is a copy, create a new UUID. Conversely, if the VM has moved to a new location, keep the existing UUID. Make sure no duplicate MAC

addresses appear on your network, because hosts identify each other via MAC address. If duplicates exist, network errors will occur. If you want to permanently change the behavior of the UUID warning, you have the option of always keeping or replacing the UUID each time the VM moves.

Renaming a VMware ESX Server VM

Like GSX Server, the VM to be renamed on ESX Server must be powered down before proceeding with the renaming procedure. Also, make sure you have a backup of the VM. Assuming you're using a default installation of ESX Server and best-practice naming conventions, the configuration files for VMs will be in the `/root/vmware/` directory, and the virtual hard disks will be in the `/vmfs/vmhba:x:x:x:x/` directory.

In case you're wondering what the syntax of the VM's HBA settings is (`vmhba`), we'll slip that information in here. Let's use `vmhba0:0:0:5` as an example. The first zero represents the first SCSI adapter loaded by the VMkernel Module Loader. The second zero represents the adapter target number. The third zero indicates the LUN. The fourth number indicates the partition number; if the number is set to zero, it indicates the use of the entire hard disk.

Best practices dictate that renaming VMs includes the host, configuration file, and virtual hard disk. By naming files and directories after the host, you reduce administrative overhead.

Note If you're not comfortable with the command line, you can rename files using the Manage Files option from within the MUI.

Follow these steps:

1. After changing the host name within the operating system, power down the VM.
2. At the command console or in a terminal session, rename the `*.vmdk` file using the `mv` command. Be sure to change your present working directory to `/vmfs/vmhba:x:x:x:x/`. If you're unsure of the variables in the `vmhba` settings directory, list the contents of the `/vmfs` directory using `ll /vmfs`.

3. Change your current working directory to the location of the VM's configuration file, `cd /root/vmware/<old_name>/`, and rename the configuration file. You can also use the Manage Files option found in the MUI to rename the disk:

```
mv <old_name.vmx> <new_name.vmx>
```

4. Edit the `*.vmx` configuration file using the Vi editor to reflect the name change, vi `<new_name.vmx>`. Modify each line, as in the example that follows, making the old VM's name reflect that of the new:

```
displayName = "<new_name>"
checkpoint.cptConfigName = "<new_name>-6dda118a"
scsi0:0.name = "vmhba0:0:0:5: <new_name>.vmdk"
scsi0:0.fileName = "vmhba0:0:0:5: <new_name>.vmdk"
```

5. In the MUI, remove the VM. You won't have to worry about losing configuration files for disk files because they've been renamed.
6. At the command line, register the renamed VM with ESX Server:


```
vmware-cmd -s register </root/vmware/<new_name>/new_name.vmx>
```
7. Before running the VM, verify the name changes by selecting its Option tab and selecting Click Here under Verbose Options. If necessary, change each verbose option to reflect the new name.

Importing Workstation and GSX Server VMs into ESX Server

Sooner or later you'll need to move a VM from GSX Server to ESX Server. Regardless for the reasons for the move, you'll want to make sure that enough free space is available on the destination SCSI device to receive the virtual disk files (*.vmdk), that paths are correct, and that permissions are set. Follow these steps:

1. Verify that the amount of free space on the new host is sufficient to hold the guest VM's virtual disk to be moved by executing `vdf` at the command line. The virtual disk will be imported into one of your VMFS partitions. Is there room for it?
2. The disks to be imported will need to be available from the Service Console. This means you'll need to copy the files from GSX Server to a non-VMFS partition on the ESX Server, or you can connect to a network share with the files. Being that files are less likely to become corrupt with fewer moves, you're probably better off directly connecting to the GSX server hosting the disks to be imported. To do this, create a share and copy the VM to be moved into the new directory. Set permissions on the share as necessary. On ESX Server, create a directory in `/mnt` to connect to GSX Server (for example, `mkdir /mnt/import`). From ESX Server's console, connect to the GSX Server share using the mount command:

```
mount -t smbfs -o username=administrator,workgroup=<domain>,
password=123456 //RemoteServer/RemoteShare /mnt/import
```

Note If you're running Workstation or GSX on Linux, you can use `ssh` and `scp` to copy files between the two servers.

3. Using the `vmkfstools` command, import the VM's disk. Notice that the filename stays the same but that the disk extensions change. The `vmhba_number` option is the VMFS partition to be imported to on ESX Server:


```
vmkfstools -i </source/filename.vmdk> vmhbaX:X:X:X:<filename.dsk>
```
4. Create a new VM with the same host name and virtual hardware as the original VM. Specify the newly imported disk for the VM's hard drive. Lastly, edit the VMs Verbose Options settings to match any of the original VM options that are vital to your production environment, such as the MAC address.

Moving or Copying VMware ESX Server VMs to Other ESX Hosts

It may become necessary to move an ESX Server VM when equipment goes bad, when leased equipment must be returned, or when upgrades are made. Some administrators have multiple copies of the same VM on several servers for redundancy purposes. In the event of hardware failure or an OS crash, downtime is significantly reduced because the solution merely needs to be turned on. Moving or copying an ESX Server VM from one ESX Server requires you to copy the VM's virtual disk, copy its configuration file, and register it with the new host.

Note VMware recommends that virtual disks not be directly moved between VMFS network shares. The virtual machine to be copied should first be exported from its VMFS volume using `vmkfstools` to the local file system and then moved. VMFS is a proprietary file system, and disk corruption can occur with a direct network move.

Follow these steps:

1. Power down the VM to be copied or moved.
2. Export the VM's virtual disk using `vmkfstools` with the `-e` option:

```
vmkfstools -e </Disk_Destination.vmdk>➡
vmhbaX:X:X:X:<VM_Source_Disk_Name.dsk>
```

For example:

```
vmkfstools -e /export/vm1.vmdk vmhba0:0:0:5:vm1.vmdk
```

3. List the contents of the export directory to verify the export. The `vmkfstools` command will create files 2GB in size in the destination partition:

```
ll /export
```

Tip You can use ESX Server disks exported with `vmkfstools` in VMware GSX Server and Workstation.

4. VMware ESX Server enables ssh by default. You can use the Secure Copy (`scp`) command to send files between the servers securely because the data will be encrypted. In our example, you'll use two options: `-rp`. The `-p` option preserves modification and access times as well as the modes of the files to be copied. The `-r` option recursively copies the entire directory. When you execute the command, you'll have to supply the password of the user specified in the `scp` command:

```
scp -rp /<VM_Source_Directory> UserID@Remote_Server:➡
/<VM_Destination_Directory>
```

A typical production example of the command looks like the following:

```
scp -r /vmtest root@10.100.2.11:/vmtest
```

5. Now that the virtual disk has been copied to the new destination, it's time to copy over the VM's configuration file. The example here assumes you used a default install of ESX Server. Be sure to change your directories as necessary:

```
scp /root/vmware/vmtest/vmtest.vmx root@10.100.2.11:➔
/root/vmware/vmtest/vmtest.vmx
```

6. All the necessary files for the VM should now be on the new host server. Import the VM's virtual hard disk into ESX Server using the `vmkfstools` command:

```
vmkfstools -i </vmtest.vmdk> vmhbaX:X:X:vmtest.dsk
```

7. Register the VM's configuration file with the MUI using the `vmware-cmd` command. After registering the VM, verify the VM's configuration before running it. Be sure to look at the Verbose Options area on the Options tab:

```
vmware-cmd -s register /root/vmware/vmtest/vmtest.vmx
```

Renaming a Microsoft Virtual Server VM

Renaming a VM for Microsoft Virtual Server follows approximately the same process as Virtual PC. The VM must be powered down and not in a suspended state, the virtual hard disk file and configuration file will need to be renamed, and edits to the configuration file must be made. Follow these steps:

1. Launch the Virtual Server Administration Web site, and select the VM to be renamed from under Status. Select Turn Off.
2. Because you may not be using the default directory structure to store virtual disks for guest VMs, select Edit Configuration under Status. Note the location of the *.vmc file and the location of the *.vhd file in the Virtual Hard Disk Properties section.
3. After collecting the VM's configuration information, select Master Status from the sidebar, select the VM in question, and select Remove from the pop-up menu.
4. Locate the VM directory containing the guest's VM configuration file and virtual hard disk file from the information collected in step 2. Rename the directory containing the VM's files, the *.vhd file, and the *.vmc file to exhibit the new name of the VM.
5. Using a text editor, edit the *.vmc file to reflect the name change. You'll need to modify three lines: the virtual disk configuration lines and the display name line:

```
<absolute type="string">C:\VirtualServerVMs\vmstest\vmstest.vhd</absolute>
<relative type="string">.\vmstest.vhd</relative>
<name type="string">vmstest</name>
```

6. Launch Virtual Server, and select Add from under Virtual Machines. Specify the absolute path of the renamed VM. Include the name of the configuration file and the *.vmc extension.

7. Under Configuration, select Hard Disks. Supply the absolute path to the renamed virtual disk file. Include the name of the configuration file and the *.vhd extension.
8. If the VM is a copy of an existing VM, be sure to delete the Ethernet MAC address value from `<ethernet_card_address type="bytes">MAC_ADDRESS</ethernet_card_address>` to avoid having duplicates on your network. On initialization, Virtual Server will create a new unique MAC.

Copying and Moving Virtual Server VMs to Other Hosts

Microsoft recommends two methods for backing up its VMs. The first method is to treat the VM like a typical physical server or workstation to be backed up. For instance, if you're using QiNetix or Backup Exec in your backup environment, load the backup agent on the VM and include it in your existing backup process. The advantage to using the first approach is that you won't have to create any new administrative processes to accommodate new guest VMs. The second method for backing up Microsoft Virtual Server VMs is to back up virtual hard disk and configuration files of powered-down VMs. Backup software will treat the VMs as if they were just a group of normal files. During the backup process, you'll want to specifically back up the virtual hard disk image file (*.vhd), the configuration file (*.vmc), and the saved state file (*.vsv) if the VM is suspended. The advantage of the second approach is that the entire state of a VM can be restored by using a few files.

Caution Be careful when backing up VMs in a suspended mode because the System State is saved to memory in two locations: partly in the *.vhd file and partly in the *.vmc file. Restoring a VM in suspend mode can and generally does create an unstable VM. If possible, avoid this approach.

When it comes time to move a guest VM between hosts, power down the VM to be moved and locate two files: the virtual hard disk file (*.vhd) and the VM configuration file (*.vmc). Before initiating the move, make sure you have a good backup of the VM to be moved. If the move process fails, you'll want to be able to rapidly revert to your existing guest VM configuration. After creating and confirming you have a good backup, ensure that your user account has the necessary privileges to complete all tasks. Then follow these steps:

1. Power down the guest VM to be moved. Make sure that it's *not* paused; moving a paused VM is a recipe for disaster.
2. Close the Microsoft server application.
3. Ensure that a transfer medium exists for moving the guest's files. Assuming your VM can fit on a CD or DVD, you can use this type of medium, or you can use a mutually accessible network share: copy the files to the network share and then move them to their final destination. If the guest is going to be placed on a new network host, the easiest way to move the files is over your existing network, going point to point.

4. Make sure the new host system has sufficient resources for the added load of the VM to be moved. Next, make sure Microsoft Virtual Server is correctly licensed and configured on the new destination host machine and not powered on.
5. Create a directory on the destination host to receive the guest VM files. (If you're just moving the location of the guest VM to a different directory on the current host, create the new directory in the desired location and locate the VM's files within the new directory.) When creating directories for VMs, use meaningful names to store all configuration files and information: name the directories after the VM for easier administration.
6. Ensure that the guest VM is powered down. Now, find *all* the files for the guest VM to be moved on the current host and copy them to the new host in the directory you just created. Files you'll minimally be concerned with are *.vmc, which is the configuration file, and *.vhd, which is the virtual disk file.
7. Launch Virtual Server, and select Add from under Virtual Machines. Specify the absolute path of the renamed VM. Include the name of the configuration file and the *.vmc extension.
8. Under Configuration, select Hard Disks. Supply the absolute path to the renamed virtual disk file. Include the name of the configuration file and the *.vhd extension.
9. If the VM was moved, it will need to retain its MAC address; you don't need to regenerate its MAC address, as in the copy process. If the VM is a copy of an existing VM, be sure to delete the Ethernet MAC address value from `<ethernet_card_address type="bytes">MAC_ADDRESS</ethernet_card_address>` to avoid having duplicates on your network. On initialization, Virtual Server will create a new unique MAC. If the guest VM being moved is from Virtual PC to Server, be sure to reinstall Virtual Machine Additions.

VMWARE VIRTUAL DISK DRIVE GEOMETRY

VMware lists the drive geometry for virtual hard disks near the top of *.vmdk files. You shouldn't open or edit a virtual disk file because of the possibility of introducing errors and rendering the disk unusable. If warnings pique your interest and make you want to do things you're not supposed to, then experimenting with virtual disk drive geometry will be exciting for you. A typical *.vmdk file will have several entries regarding the virtual disk, including cylinders, heads, sectors, and the adapter disk type. For example:

```
# Extent description
RW 268435456 SPARSE "ide.vmdk"
# The Disk Data Base
#DDB
ddb.virtualHWVersion- = "3"
ddb.geometry.cylinders = "16383"
ddb.geometry.heads = "16"
ddb.geometry.sectors = "63"
ddb.adapterType = "ide"
```

Most of the information regarding the disk is clear, save for the RW in the extent description. The number appears to be a description of the virtual hard disk's size. For instance, 268435456 is a decimal representation of the octal number 200000000. If you drop the last seven zeros, you'll be left with the number 200; 200 octal is equal to decimal 128—the size of the VM's virtual hard disk in gigabytes. To clarify the previous example, the following RW extent descriptions are calculated for you to further demonstrate the relationships between the IDE RW decimal value, octal value, and drive size:

IDE RW Decimal Value	Octal Value	Drive Size in GBs
2097152	10000000	1
4194304	20000000	2
6291456	30000000	3
8388608	40000000	4
20971520	100000000	10
41943040	200000000	20
62914560	300000000	30
83886080	400000000	40

Manually changing the drive geometry will allow you to reduce or increase the size of a disk without using third-party software, such as Norton Ghost or MaxBlast. To figure out what to set the RW, cylinders, heads, and sectors to, create a new drive the size you desire and view its drive geometry. Change the settings on the drive to be resized to match; using a good hex editor, such as WinHex, will be invaluable in modification process because dynamic disk files can be gigabytes in size—Notepad isn't going to cut it. Next, remove the virtual disk from the guest VM, remove the VM from the MUI, and then close the virtualization application. Reopen VMware, and add the VM back and its disk by creating a new disk and specifying the existing disk as the type.

Being that you'll experience varying degrees of failure (less so with SCSI virtual hard disks), don't expect to resize a production drive using this technique and have it work for long. For production VMs, be sure to use reputable third-party utilities to move bootable systems from disk to disk. More information is available about drive geometry on VMware's Web site at http://www.vmware.com/support/esx21/doc/esx21admin_migrate_other_vms.html#1049351; the site covers how to modify drive geometry using the `vmkfstools` command with the `-g` option.

Working with VMware GSX Server and ESX Server *.vmx Configuration Files

In general, manually changing a VMware configuration file isn't necessary and isn't recommended. If you need to make changes, be careful of typos and accidental deletions. VMware recommends making changes through the Virtual Machine Console. Poor editing of a configuration file can render a VM worthless. Every virtual machine has a *.vmx configuration file that contains the settings for the VM. You can locate the file with the aid of the Virtual Machine Editor. The Options tab displays the location of the VM's configuration file. You'll now look at some more options found in a configuration file that we didn't cover in Chapter 4.

You can control virtual device behavior by changing the device's value. Values range from simple TRUE/FALSE options to directory path statements. For instance, the configuration file will list devices as not being present if they have a status of FALSE. Look at the following snippet from an ESX Server VM's configuration file regarding system disks. The first two `scsi` statements tell ESX Server that the VM has the LSI SCSI adapter enabled for the hard disk listed in the fourth line. If you want your VM to discontinue using the disk, set the value in the third line to FALSE. The remaining statements control the properties for the virtual CD-ROM and floppy disk drive. If you want a better level of performance for the VM, set the value in the fifth line to FALSE; the CD-ROM should be enabled only when it's needed. The *.filename values for the floppy and CD-ROM drive are the mount points for each device.

```
scsi0.present = "TRUE"
scsi0.virtualDev = "vmxlsilogic"
scsi0:0.present = "TRUE"
scsi0:0.name = "vmhba0:0:0:5:vm1print1.vmdk"
ide0:0.present = "TRUE"
ide0:0.fileName = "/dev/cdrom"
ide0:0.deviceType = "atapi-cdrom"
floppy0.startConnected = "FALSE"
floppy0.fileName = "/dev/fd0"
```

In the following configuration example, the first two lines are fairly self-explanatory; you can edit the first line to change the VM's name in the MUI. The second line is the amount of memory allocated to the VM in megabytes. It's important the `guestOS` option correctly reflects that of the VM's OS. VMware makes optimizations based on the value setting. Other values you'll see include `linux` and `netware`. In the fourth line, `RemoteDisplay.depth` controls the color depth of a VM. Increasing the color depth to 24 bits can negatively impact performance. Conversely, if you can get away with 8 bits, use 8 as the value. The fifth line controls the performance of a VM when it has the keyboard and mouse focus, and the sixth line controls performance when a different VM has focus. You can set `priority.grabbed` to `normal` or `high`, and you can set `priority.ungrabbed` to `normal` or `low`. If you want guest VMs to respond better in the console, you can set `priority.grabbed` to `high`; however, this will negatively impact the performance of the other VMs. In a lab environment, you may want to set `priority.ungrabbed` to `low` to boost the performance of the VM that has focus to decrease latency. The `tools.syncTime` statement configures a VM to either synchronize with the host's clock or not. If you want your VMs to use NTP, it's safe to set this option to FALSE.

```
displayName = "vm1print1"
memsize = "384"
guestOS = "winNetStandard"
RemoteDisplay.depth = "16"
priority.grabbed = "normal"
priority.ungrabbed = "normal"
tools.syncTime = "FALSE"
```


After creating a snapshot of a VM, several lines are added to its *.vmx configuration file. Most of the lines deal with how the snapshot will be treated for the VM's power cycles. However, one particular statement of interest is `gui.restricted`. The `gui.restricted` line enables the restricted user interface. This option hides the toolbar, hides all functions on the Power menu and Snapshot menu, prohibits changes to virtual networking settings, and denies access to the VM Control Panel.

```
gui.restricted = "FALSE"
```

The `undopoint.action` statement directs VMware to do one of several things with a snapshot, such as keeping changes or reverting to its original state. Table 6-3 lists its options.

Table 6-3. *undopoint.action* Statement Values

Value	Action
<code>autoRevert</code>	On power down, reverts to original state of snapshot
<code>autoCommit</code>	Automatically updates snapshot on power down
<code>prompt</code>	On power down, prompts user what to do with snapshot

Several additional `undopoint` statements are created when a snapshot is taken. Looking at the first line, `resotreFromCheckpoint`, this statement instructs the application to keep its current state when set to `FALSE` or to revert to a snapshot when it's set to `TRUE`. The second line, `checkpointedOnline`, will have a value of `TRUE` if the snapshot was taken while a guest was running. The third line, `protected`, will have a setting of `TRUE` if the snapshot is designated as locked in the console. A locked snapshot prevents it from being updated.

```
undopoint.restoreFromCheckpoint = "FALSE" Revert snap when true
undopoint.checkpointedOnline = "TRUE" Snapshot taken while OS running.
undopoint.protected = "FALSE" True for locked snapshot
```

The location of the undo log is denoted in the VM's disk redo statement. You can place the redo log in the same directory as the virtual disk, `.\`, or you can direct it to any network or local resource:

```
scsi0:0.redo = ".\12gbscsi.vmdk.REDO_a00356"
```

Note Using redo logs, you can share virtual disk base images. This is handy for setting up many servers that are similar in nature, such as Web servers. An excellent article on this is available at http://www.vmware.com/support/reference/common/disk_sharing.html. The article details how to create the base image and the corresponding redo log files.

The last thing we want to touch on in this section are the Ethernet settings. In the following *.vmx code, a single Ethernet adapter is configured for the VM and is noted as being present by TRUE in the first line. The second, third, and fourth lines describe the VM's connection type, the adapter's name, and the network in which to connect. The fifth line informs you that a unique MAC (line 8) has been generated for the VM rather than being statically assigned (static). The ethernet0.generatedAddressOffset line calculates the UUID for the VM that generates the MAC address. After a MAC address is generated and a duplicate address is found to exist, the offset value is added. VMware will continue to check for duplicate addresses and continue to add the offset until a unique MAC address is derived. To generate the next available MAC address, the offset number (represented in decimal) is added to the last byte of the MAC address (represented in hex). In the eighth line, the da hexadecimal is equal to 218 decimal. Add the offset from the ninth line to 218 to arrive at the next value for the last byte of the second MAC address, 238 decimal or EE hexadecimal.

```
Ethernet0.present = "TRUE"
Ethernet0.connectionType = "monitor_dev"
Ethernet0.devName = "vmnic0"
Ethernet0.networkName = "Network0"
Ethernet0.addressType = "generated"
uuid.location = "56 4d 7e 42 db dc 5c fb-e9 6f d3 cd ef 07 47 da"
uuid.bios = "56 4d 7e 42 db dc 5c fb-e9 6f d3 cd ef 07 47 da"
ethernet0.generatedAddress = "00:0c:29:07:47:da"
ethernet0.generatedAddressOffset = "20"
```

Working with Virtual Server *.vmc Configuration Files

Virtual Server's configuration file is similar to Virtual PC's. The file is stored in a text-editable XML format. It's used to store a guest VM's configuration information and can be manually edited. Before editing the file, be sure to make a backup of the file in case you enter errors during the editing process.

One necessary change you'll want to make to the configuration file is the VM's MAC address to avoid duplicates on your network. Unlike VMware's use of the UUID to help manage MAC addresses, Virtual Server will passively "allow" duplicates to occur if you're making a copy of a VM to run concurrently with the parent image. If you're just moving the VM, it's not necessary to change the MAC address.

To change the MAC address, find the section of the configuration file containing the text string `<ethernet_card_address type="bytes">0003FFD6C41F</ethernet_card_address>`. Remove the 48-bit address from the string and save the file. In our example, the MAC address is 0003FFD6C41F. On the VM's next boot, Virtual PC will see that the MAC address has been removed, and it will generate a new MAC address for the VM. You can also change the MAC address of a VM by using the Administration Web site and editing the network adapter properties.

Other entries in the *.vmc configuration file you may want to experiment with are the settings to change the name and location of the *.vhd configuration file (ide_controller), screen size and resolution (video), RAM allocation (memory), and mouse behavior (mouse). Because the file is a text representation of the Virtual Machine Options and Settings dialog box, you can programmatically add or remove devices on dozens of machines with little scripting effort.

Note If you want to add virtual devices to a guest VM and are unsure of the settings in the *.vmc file, create a VM and remove all the devices. Next, add the device in question, and look at the changes made to the *.vmc file.

Performing Command-Line Management

In Chapter 4, you looked at two CLI executables, `vmware.exe` and `Virtual PC.exe`. We also covered basic keyboard shortcuts. In the following sections, you'll look at a few other commands for VMware: `vmware-vdiskmanager` and `vmkfstools`. You'll also learn about Virtual Server's `msiexec.exe`.

VMware

The `vmware-vdiskmanager` utility for Windows hosts allows you to manage, modify, and create virtual disks from the command line. You can also use the command in scripts for automating virtual disk management tasks, such as breaking large virtual disk files into chunks to be exported to DVD. Additionally, you can use the command to increase the size of a virtual disk beyond what it was initially created at without adding an existing disk or using third-party disk imaging products. If you find a need to revert a VM's preallocated disk to a dynamic disk, then `vmware-vdiskmanager` is the command for you; you may find yourself in this situation when you need to add VMs to a server and no disk space exists. Examples of how to use the command follow:

- To increase the size of an existing virtual disk, use `vmware-vdiskmanager -x 20GB virtual_disk_name.vmdk`.
- To convert a preallocated disk to a dynamic disk, use `vmware-vdiskmanager -r virtual_disk_source.vmdk -t 0 virtual_disk_destination.vmdk`.
- To create a virtual IDE disk, use `vmware-vdiskmanager -c -t 0 -s 80GB -a ide virtual_disk_name.vmdk`.
- To rename a virtual hard disk, you must first remove it from the VM to which it's attached; use `vmware-vdiskmanager -n old_virtual_disk_name.vmdk new_virtual_disk_name.vmdk`.

Note VMware discusses shrinking and defragmenting virtual disks and other handy options of `vmware-vdiskmanager` on its Web site at http://www.vmware.com/support/gsx3/doc/disks_vdiskmanager_eg_gsx.html#1088627.

The `vmkfstools` command for ESX Server allows you to manage, modify, and create virtual disks from the command line. Because both `vmkfstools` and `vmware-vdiskmanager` allow you to increase the size of a virtual disk, there's no benefit in allocating more space to a virtual disk than necessary. The hazard in overcommitting disk space is that you can't reduce the size of the disk in the future. Our treatment of the `vmkfstools` in this section is by no means exhaustive; VMware discusses this command at length on its Web site in each virtualization application's administration manual. You can find ESX Server's manual at http://www.vmware.com/support/esx21/doc/esx21admin_vmkfstools.html#999999. The syntax for the command is `vmkfstools <options> <path>`.

- To create a new virtual disk, you need to specify the size and name of the disk with `vmkfstools -c <size_in_megabytes>m vmhbx:x:x:x:<virtual_disk_name>.disk`.
- To import a virtual disk into a VMFS partition, use `vmkfstools -i <virtual_disk_source>.disk vmhbx:x:x:x:<virtual_disk_destination>.disk`.
- To export a virtual disk from VMFS partitions for use on GSX Server or Workstation, use `vmkfstools -e <destination_disk_name>.vmdk vmhbx:x:x:x:<source_disk_name>.disk`.
- To list VMFS virtual disks, use `vmkfstools -l vmhbx:x:x:x`.

Before making changes to a VMFS virtual disk, such as increasing a virtual disk, the VMFS volume needs to be writable. To do this, power down all VMs accessing the volume and execute `vmkfstools --config writable`. To grow a disk, you need to use the `-X` option. After increasing the disk, the file system may need to be modified to see the new larger disk:

```
vmkfstools -X <number_of_megabytes_to_grow_to>m
```

The option to display virtual disk geometry with `vmkfstools` is `-g`. This option becomes handy when an import from GSX Server to ESX Server fails because of the difference in how ESX Server and GSX Server handle drive geometry. Before importing a disk, you should view the current geometry of the disk in GSX Server before importing it to a VMFS partition:

```
vmkfstools -g <GSX_virtual_disk>.vmdk
```

After the import, run the command again on the newly imported disk:

```
vmkfstools -g <ESX_virtual_disk>.disk
```

If the drive geometry doesn't match, you can correct the problem by specifying the correct information in the configuration file by adding a drive geometry entry:

```
scsi <adapter-id>:<target-id>.biosGeometry = "<cylinders>/<heads>/<sectors>"
```

Microsoft

You can install Microsoft Virtual Server using the Microsoft Installer, `msiexec.exe`. This is valuable for deploying Virtual Server by using group policies or performing an unattended installation. The syntax for the command has several parameters and options:

```
msiexec.exe {/i|/a} "msifile" [ALLUSERS=value] [PIDKEY=value]
[SERVICESSTARTUPMANUAL=value] [WEBSITEDEFAULTPORT=value]
[{INSTALLDIR=value|TARGETDIR=value}] [ALLUSERS=value]
[NOSUMMARY=value] [/qb | /qn | /qr | /qf] [/l logfile]
```

Table 6-4 details the common configuration parameters.

Table 6-4. *Virtual PC Microsoft Installer Options*

Option	Action Performed
/i	Installs Virtual Server.
/x	Uninstalls Virtual Server.
/a	Installs Virtual Server on the network share.
/q	Sets installation user interface: /qn for none, /qb for basic, /qr for reduced, /qf for full.
/l <File>	Provides the installation log file path.
PIDKEY=<KeyWithoutDashes>	This is the Virtual Server product key without dashes.
SERVICESSTARTUPMANUAL=<Number>	Automatically starts using 0. Does annual start using 1.
YOUBSITEDEFAULTPORT=<Number>	Specifies the Administration Web site port. This default is 1024.
INSTALLDIR=<Path>	This is a custom installation directory.
TARGETDIR=<NetworkShare>	This is the location of the installation package when used with /a.
NOSUMMARY=1	Suppresses summary page.

You can perform an unattended installation for a user called *test* by executing the example:

```
msiexec.exe /i "Virtual Server 2005 Install.msi"
ALLUSERS="test" PIDKEY=<KeyCode> /qn
```

If you want to install Virtual Server for all users, drop the ALLUSERS option:

```
msiexec.exe /i "Virtual Server 2005 Install.msi" PIDKEY=<KeyCode> /qn
```

If you want to deploy Virtual Server using group policies, you'll need to invoke the administration option, /a, and specify a directory from the group policy that will read the installation files, TARGETDIR. You can create a log file for each install by using the -l option and pointing it to a network share:

```
msiexec.exe /a "Virtual Server 2005 Install.msi"
PIDKEY=<KeyCode> TARGETDIR=<NetworkShare> /qn /l <NetworkShare>
```

Using the Windows System Preparation Tool

The Windows System Preparation Tool (Sysprep) allows you to prepare OS images for automated deployment scenarios, such as rolling out successive VM servers on a single host or creating VMs for quick setup in a lab environment. Microsoft suggests that if you use the Sysprep tool that you use group group policies to reset security settings. In this section, you'll learn how to use Sysprepped VMs:

1. Locate a copy of the Sysprep tool; you can download it from Microsoft's Web site at <http://www.microsoft.com/downloads>. Additionally, you can use a search engine to locate it. Make sure you have three files: Sysprep.exe, Sysprep.inf, and Setupcl.exe. Sysprep.exe prepares the VM for imaging, and Sysprep.inf is a file containing any configuration settings the VM image requires for its first boot. Setupcl.exe is a setup wizard.
2. Create a VM, and install the operating system. As the system administrator, install any hotfixes, patches, and service packs.
3. Create a secondary account, such as SysprepAdmin, and place it in the Administrators group. Log off and then back in with the new admin account.
4. Install all applications and any necessary updates. Make any necessary customizations, such as tweaking the desktop, configuring Internet Explorer, installing printers, and so on.
5. Off the root of a VM's hard disk, create a directory labeled c:\sysprep. Copy Sysprep.exe and Setupcl.exe into it. You'll copy Sysprep.inf into the directory after you create it in step 10. Under the Sysprep directory, create a directory titled drivers. You'll need to use the drivers directory if you're using virtual devices that the OS doesn't recognize, such as the LSI controller. If you're using it, create a directory under drivers titled DiskController and place the LSI driver in it. The driver directories should have only the device driver and related *.inf and *.cat files. Avoid placing any *.exe files in the directories. You'll later use these directories in the Sysprep.inf file.
6. Log off and back on as the system administrator. Add the secondary administrator account profile from step 3 to the Default User profile in the Control Panel. Grant the Everyone group access to the profile. Now, delete the secondary administrator account. Copying the profile ensures all VM users have the profile configured from step 4. Removing the secondary administrator account removes any security risk the account has associated with it.
7. Take a moment to optimize the VM. Remove all unnecessary overhead, such as frivolous programs, temporary files, accessories, and icons. You may even want to run the Disk Cleanup Wizard and defragment the hard drive.
8. Before running Sysprep.exe, make sure the VM to be imaged isn't joined to the domain. At the CLI, run Sysprep.exe from within its directory from step 5. On the System Prep Tool screen, select Use Mini Setup and Detect Non-Plug and Play Hardware from under Options. Select Reseal to start the process. You can pass several additional command-line options to the executable. To get help with the command, see Table 6-5, or execute sysprep.exe /?.

Table 6-5. *Sysprep Command-Line Help Options*

Option	Action
/forcshutdown	Forces a shutdown of VM that fails under normal command usage.
/noreboot	Doesn't shut down after running.
/nosidgen	Doesn't generate new SID on reboot. This isn't useful for making multiple copies of VMs.
/pnp	Reenumerate all devices in VM. This is helpful if you're using legacy devices.
/reboot	Reboots the VM rather than shuts down, and starts mini-setup.

9. After completing the Sysprep process, shut down the VM and make an image of it. Like physical computers, you can create a bootable ISO of the VM, or you can make a copy of the VM's directory. The copy option reduces working time. Be sure to name the directories of the new VM after the VM itself. Also, make any necessary name changes to the *.vmdk and *.vmx files.
10. You can use Setup Manager, Setupmgr.exe, to create the Sysprep.inf file. The executable should come with the Sysprep package downloaded from Microsoft. Launch the executable, and answer each question in the wizard. If you leave something out, you'll have to supply the information during the initial boot. When using Setup Manager, be sure to select the Sysprep setup option, the correct OS of the VM, and an unattended install. You'll basically be supplying all the information that's normally supplied during a typical installation of a Microsoft OS.
11. Start the new VM. On boot, the mini-setup wizard executes and uses the settings from the Sysprep.inf file to configure the VM.
12. Rename and join the VM to your domain or workgroup as necessary.

Monitoring VM Performance

When looking at how to maintain high levels of performance for your VMs, you'll need to stick to the tried-and-true methods of performing a baseline analysis and then follow up with periodic maintenance reporting to find system and network trends. For Windows systems, you'll be working with System Monitor and dealing with guest VM counters. Microsoft includes special counters for Virtual Server as does VMware for GSX Server. You'll also look at ESX Server's built-in performance monitoring Web site and the esxtop command. When creating a baseline for the host, you'll look at four main categories: CPU, RAM, disk, and network bottlenecks.

Monitoring ESX Server Performance

ESX Server installs performance-monitoring capabilities by default. vmkusage presents historical data on Web-based graphs, and esxtop displays real-time processor statistics for ESX Server; you'll look at vmkusage first and then esxtop.

To harness the power of vmkusage, you'll need to activate it by executing vmkusagectl install at the command line. By default, vmkusagectl will run once a minute through cron. It

gathers information from /proc and stores gathered information in a database. The information is viewable by opening a Web browser and pointing it to `https://<ESX_SERVER_IP>/vmkusage`. Each guest VM is listed along with its CPU, memory, network, and storage statistics. Graphs for each VM resources are listed below the general statistics of the VM. Sometimes vmkusage doesn't properly graph collected information. If this is the case for you, blow away the database and reactivate vmkusagectl:

```
vmkusage -regroove
vmkusagectl uninstall
vmkusagectl cleandb
vmkusagectl install
```

Note You can use two commands with the performance tools in ESX Server, vmkusage and vmkusagectl. They both perform identical functions save for a few options that may work only with vmkusage, as in -regroove. If an option doesn't work with one command, try the other; in addition, you may have to try options with and without a preceding dash. You can get help with each by supplying either vmkusage -help or vmkusagectl support.

Table 6-6 lists common options to use with vmkusagectl.

Table 6-6. Popular vmkusagectl Command Options

Option	Action Performed
graph	Creates guest VM graphs.
hyper	Polls every 10 seconds and updates every minute. Runs regroove and uninstall first.
monthly	Creates monthly graphs using larger databases. Runs regroove first.
nograph	Doesn't create guest VM graphs.
regroove	Deletes databases and starts over.
verbose	Loquaciousness is rewarded with copious verbiage.

Use the following to get the default capabilities of vmkusagectl and graph:

```
vmkusagectl install graph
```

Use the following to poll more frequently to gather more graph data:

```
vmkusagectl install hyper graph
```

Use the following to generate monthly graphs:

```
vmkusagectl install monthly graph
```


Another command you can toss in your bag of tricks for monitoring ESX Server is the `esxtop` command. It displays utilization statistics of each physical process and installed memory. In addition, `esxtop` monitors disk and network bandwidth for each physical disk and physical network adapter. The `esxtop` command has many options available; you can get help by using the manual pages at the command line, `man esxtop`.

Note The first output line displayed with `esxtop` displays several statistics, including uptime, worlds, and load average. The term *worlds* refers to VMkernel running processes. Three world types exist: System, Service Console, and Virtual Machine. Load average is important because you can quickly determine the CPU load. A load average of 1 means that the server's physical CPU is completely utilized. A load average of 2 means the physical server needs twice as many processors. A load average of .5 means the CPU is half utilized. Be sure to read the `man` pages on this to get a feel for the historical display.

While `esxtop` is running, you can pass several commands to it interactively to toggle information on and off. Table 6-7 lists the commands you'll need to gather baseline information regarding the CPU, memory, disks, and networking information. To get an idea how the interactive commands work, run `esxtop` and toggle all options off and toggle each on individually.

Table 6-7. *esxtop Interactive Commands*

Interactive Command	Action Performed
c	Toggles display of ESX Server CPU information
f	Toggles additional fields
m	Toggles display of ESX Server memory information
p	Toggles display of ESX Server swap information
d	Toggles display of ESX Server disk information
n	Toggles display of ESX Server network interface information

You can also pass a handful of command-line options to `esxtop` upon execution. The command options are particularly important because they help you automate data gathering by limiting the data collection window and dumping the results to a text file. Table 6-8 lists the delay (d), iterations (n), and batch (b) options.

Table 6-8. *esxtop Command Options*

Option	Action Performed
d	Updates delay in seconds (5 is the minimum)
n	Performs n updates and then exits
b	Sends output to other programs, scripts, or a text file

If you were creating a baseline and wanted to use `esxtop` to run for five minutes at thirty-second intervals and finally dump the data to a file for further processing, you'd need to execute `esxtop d 30 n 10 b > esxdatadump.txt`. Being that you're redirecting output to a file, you won't see screen updates.

Now that you've captured your baseline data, you'll need to create a baseline worksheet for historical trending purposes—there's nothing like management wallpaper to build your case for additional networking equipment. In the worksheet, you'll want to minimally record CPU, RAM, disk, and network averages. Over time, you should be able to graph server usage trends and determine the growth characteristics of the network. Without historical data, information from `esxtop` is meaningless.

You can identify a host CPU bottleneck by examining load average. A load average of 1 means the server's physical CPU is completely utilized. A load average of 2 means the physical server needs twice as many processors or fewer guest VMs. A load average of .5 means the CPU is half utilized. Ideally, you'll want to utilize about 60–80 percent of the physical CPU (PCPU). Loading up the processor beyond this can cause excessive paging during high utilization periods, which significantly impacts performance. If you have high utilization levels, look at each VM's logical CPU (LCPU) to determine the resource hogs. You can find the virtual machine ID in the MUI's Display Name field. Use this number to identify a VM listed in the VCPUID field from the `esxtop` output. Three columns of data are available for each VM: %USED, %READY, and %EUSED. The %USED field is the percentage of the physical CPU being used by the VM. The %READY field is the percentage of time a VM is ready to run but can't get scheduled. VMware suggests that this number shouldn't exceed 5 percent. The %EUSED field represents how much a VM is saturating the physical CPUs. A high CPU saturation level is a good indicator of a resource hog—you may need to move this VM to a less-utilized server or add processors.

To determine if a bottleneck exists with memory, first look at the total amount of free system memory; if this number is relatively low and a fair amount of swap memory is used, you should install more RAM. Determining memory bottlenecks for guest VMs requires that you compare the maximum memory used (the %MEM column) with swapped memory utilized (the SWPD column). To see the swapped memory field for a VM, press the F key while `esxtop` is running and select R. You can expect to see some memory swapping, but excessive or a consistent increase in usage over time can indicate that the system needs more physical memory.

Identifying network bottlenecks is a matter of knowing the physical constraints of your hardware and network. With the performance thresholds known, you can compare these to the NIC information in `esxtop`. As a baseline approach, the sum of MbTx/s and MbRx/s shouldn't approach anywhere near the full duplex rating for the server's NIC. If you find that the average throughput is consistently high, you may need to add faster network adapters or even add a server to handle the load.

Identifying hard disk bottlenecks requires you to know the performance characteristics of the host's controller card and have first verified that a memory bottleneck doesn't exist. Excessive memory swapping can create a disk bottleneck. Like a network adapter, the value for MBr/s and MBw/s shouldn't consistently be pegged at the hard disk controller's rating, and the r/s and w/s values shouldn't be pegged as specified by the manufacturer. If either entry is consistently reaching near the hardware's threshold, you can try to alleviate the bottleneck by first defragmenting hard drives and then turning to a hardware solution, such as using faster controllers with more disk interfaces.

Monitoring VMware GSX Server Performance

VMware GSX Server has performance counters that can be managed through the host's Performance console application. Unfortunately, the counters are available only for Microsoft hosts. You can, however, monitor the performance of Linux VMs. The performance counters included with GSX Server allow you to monitor memory usage, virtual hard disk access, and VM network traffic. To track the performance of a VM, it must be powered on. The VMware performance counters track the status of the virtual machine proper and *not* that of the guest OS.

Creating a baseline report for GSX Server should include processor, memory, disk, and network recordings. You should generate the baseline report when the host is first configured, and you should regenerate the report at regular intervals. Having historical data for the baseline measurement will help you identify trends and troubleshoot resource-related problems. Your report should include the minimum, maximum, and average values for all the counters.

Identifying CPU bottlenecks requires that you add a few extra counters to your baseline report. The counters to add from the host are %Processor Time, Interrupts/Sec, and Processor Queue Length. If Processor Time is consistently greater than 90 percent, you guessed it—you have bottleneck. This value should be around 60–80 percent to handle usage spikes. Interrupts/Sec will vary widely based on the CPU in your system. Based on your hardware, you'll have to dig around on the Internet to find a number for an alert value; for example, you can consider 3000–4000 to be a bottleneck for Pentium processors. On average, Processor Queue Length should be less than two: the queue is where data piles up when the CPU is overloaded. If you find that the host has a bottleneck, you can add more processors or remove VM guests.

You can identify disk bottlenecks by looking at Virtual Disk Reads/Sec, Virtual Disk Writes/Sec, Host Physical Disk Reads/Sec, Host Physical Disk Write/Sec, Host Physical Disk Queue Length, and Host % Disk Time. Because data is being essentially passed from the guest VM to the host, the Disk Reads/Sec and Disk Writes/Sec counter should share approximately the same values. Identifying hard disk bottlenecks requires you to know the performance characteristics of the host's controller card and have first verified that a memory bottleneck doesn't exist. Excessive memory swapping can create a disk bottleneck. The value for these counters shouldn't be redlined near the hard disk controller's rating as specified by the manufacturer. The Physical Disk Queue Length indicates disk requests waiting to be serviced. In general, physical systems should have a value of less than 2. If any of the counters consistently reach the hardware's threshold, you can try to alleviate the bottleneck by first defragmenting hard drives. You may need to provide a hardware solution by upgrading to faster disk controllers with more disk interfaces.

Memory bottlenecks should be nonexistent with the price of memory being so inexpensive; however, that doesn't mean you shouldn't monitor memory. The counters that should be minimally monitored are Percent Guest Physical Memory Touched, Host Memory Available MBytes, Host Memory Pages/Sec, and Host Paging File % Usage. Percent Guest Physical Memory captures the percentage of simulated physical memory recently used by the guest. If a VM touches its maximum memory setting, it may be memory starved. To verify this, check the duration the VM uses as the maximum amount of memory; if the touch times are high relative to running time, you'll want to allocate more RAM to the VM. The Host Memory Available MBytes counter reveals the amount of physical memory for running processes. This number should average no less than 5–6MB. Host Paging File % indicates page file usage. If the percentage is consistently greater than 90 percent, it's time to add more memory.

Identifying network bottlenecks is a matter of knowing the physical constraints of your hardware and network. With performance thresholds known, you can compare these to the counters that can help identify network bottlenecks. The counters of interest here include Network Bytes Sent/Sec, Network Bytes Received/Sec, and Host Network Bytes Total/Sec. The aggregate sum should be well within the published throughput constraints of your hardware. If the total is consistently high, you may want to add faster network adapters or add more of them; you can also look for improperly configured speed/duplex hardware settings.

Monitoring Virtual Server Performance

Monitoring Virtual Server guest VM performance is similar to that of a physical server, and the process for creating a baseline is the same. Most of the counters included with Virtual Server generate a cumulative total of the resources being monitored since the host or VM was powered on. Cumulative totals are good for looking at trends of disk usage over time or at total network throughput over time. Being able to target bottlenecks requires you to use the host's counters or use the limited output of Task Manager.

Note Rather than using the counters made available with the install of Virtual Server to monitor hard disk space usage, you can write a script that queries your domain servers and reports the data through a Web page. To get you started, read the “Monitor Disk Space on Multiple SQL Servers” article at <http://www.databasejournal.com/features/mssql/print.php/3296731>.

The most useful Virtual Server counter for baseline and trend analysis is the CpuUtilization counter. It provides the percentage of CPU resources allocated per VM. This counter shouldn't reach and remain at 100 percent utilization. If it does, then either the host resources are overcommitted or a process is misbehaving. If this problem occurs from overcommitted resources, you can solve the problem by adding more processors or by offloading guest VMs to other underutilized servers. If it's a runaway process, Task Manager can help you identify the culprit.

You can identify disk bottlenecks by looking at Physical Disk Reads/Sec, Physical Disk Write/Sec, Physical Disk Queue Length, and Host % Disk Time. Identifying hard disk bottlenecks requires you to know the performance characteristics of the host's controller card and have first verified that a memory bottleneck doesn't exist. Excessive memory swapping can create a disk bottleneck. The value for these counters shouldn't be redlined near the hard disk controller's rating as specified by the manufacturer. The Physical Disk Queue Length indicates disk requests waiting to be serviced. In general, physical systems should have a value of less than 2. If any of the counters consistently reach the hardware's threshold, you can try to alleviate the bottleneck by first decreasing hard drives. You may need to provide a hardware solution by upgrading to faster disk controllers with more disk interfaces.

Memory bottlenecks should be nonexistent with the price of memory being inexpensive; the counters that should be minimally monitored are Memory Available MBytes, Memory Pages/Sec, and Host Paging File % Usage. The Memory Available MBytes counter reveals the amount of physical memory for running processes. This number should average no less than

5–6MB. Host Paging File % indicates page file usage. If the percentage is consistently greater than 90 percent, it's time to add more memory.

Identifying network bottlenecks requires you to know the physical constraints of your hardware and network. With performance thresholds known, you can compare these to the counters that can help identify network bottlenecks. The counter of interest here is Network Bytes Total/Sec. The total should be well within the published throughput constraints of your hardware. If the total is consistently high, you may want to add faster and/or additional network adapters or look for improperly configured speed/duplex hardware settings.

Performing Fault Monitoring and Fault Tolerance

We'll now introduce you to a few sample fault-monitoring and fault-tolerance strategies for VMware to prepare you for the forthcoming enterprise management chapter. In Chapter 14, we'll cover additional fault-tolerance issues in greater detail. Two approaches exist for the fault monitoring and fault tolerance of your VMs; the first approach uses free scripts available in VMware's Scripting API (available at http://www.vmware.com/pdf/Scripting_API_215.pdf), and the second method uses VirtualCenter. VirtualCenter provides a robust built-in monitoring system that performs certain actions given defined circumstances.

VirtualCenter uses a central console so you can access and manage all your VMs from one location. In addition to extensive logging and performance graphing, new servers can be quickly deployed with the Instant Provisioning Deployment Wizard. It's a template-based system capitalizing on VMware's standardized virtual hardware. The Virtual Infrastructure SDK integrates with third-party software to capitalize on existing applications. If you want to learn more about VirtualCenter, please visit VMware's Web site.

The Scripting API has scripts to start all VMs, list all VMs, create and commit a redo log, and enumerate a VM's IP. Two scripts from the API are of particular interest: `status.pl` and `hb_check.pl`. The `status.pl` script returns the power status of a VM, and `hb_check.pl` monitors the heartbeat of a VM at a given interval in seconds. If the heartbeat isn't detected, the script will stop the given VM and start a new VM based on the supplied configuration file.

You can use the first script, `status.pl`, to verify the power state of a VM, and then the output can perform a particular action. For instance, the script comes in handy during backups when a VM is required to be shut down. If the script finds the VM in a powered-on state, a second script can be called to gracefully power off the VM. Once the backup is finished, the VM can be restarted either by passing commands from the backup application or by scheduling actions with the `at` command.

The `hb_check.pl` script is especially exciting because you can use it with little modification:

1. Download and save the script to a directory on your server.
2. Edit the VmPerl scripting directory statement to match that of the running VMs.
3. Execute the script at the CLI. If you don't specify an interval, it will default to 30 seconds:

```
./hbcheck.pl <VM_Config_To_Check> <VMConfig_To_Start> <Interval_in_Seconds>
```

If the script doesn't detect a heartbeat within the interval specified, it will attempt to stop the VM and then start a new VM in its place. If you don't want to specify a different configuration file, specify the same configuration file option for `Config_to_Check` and `Config_To_Start` to restart the failed VM.

Summary

In this chapter, you learned how to properly configure and install server VMs. We also covered techniques for monitoring production VM resource utilization and scripting basics for fault monitoring and fault tolerance. In the next chapter, you'll learn how to recover a failed virtual server and to properly plan and test VM deployments. Additionally, we'll cover backup alternatives, including using network backup agents, online snapshots, and offline "flat-file" backups.



Backing Up and Recovering Virtual Machines

For virtual machines running in a production environment, backup and recovery is just as serious as for all the other servers on the network. When running servers inside VMs, you'll be faced with several new challenges, as well as advantages, when planning and implementing a backup strategy.

In this chapter, you'll explore the process of planning for and administering backup and recovery operations on VMs and VM hosts. Along the way, you'll see the different approaches you can take to secure VM data, which include the following:

- Traditional agent-based backups
- Non-agent-based backups
- Flat-file backups

Also, many organizations have embraced the idea of maintaining a warm standby VM server that can be brought online if a primary server fails. This approach, for many organizations, may mean that data is unavailable for a few minutes following the loss of a server. If you can't afford to cluster all your systems, or if some of your applications don't support clustering, then you may find this approach to be a perfect fit. Since the focus of this chapter is purely on VM backup and recovery, we'll walk you through the process of maintaining a standby VM server in Chapter 14.

Optimizing your backup strategy often means much more than simply installing software and letting it do its magic. Oftentimes, custom scripting is required to get the backup results you desire. Because of the importance of getting your virtual infrastructure to work around your needs (instead of the other way around), we'll also show you several scripting ideas to both enhance and automate the backup, recovery, and availability of your VMs. Let's start with the most common production backup method today—running agent-based backups.

Performing Traditional Agent-Based Backups

As VMs perform write operations, some data winds up in the host system's physical memory before it's passed to a virtual disk file on the hard disk. This architecture is common in both the VMware and Microsoft products, and as a result, neither vendor supports online backups of virtual disk files. Instead, both strongly recommend you install backup agent software on the VMs in order to back them up, as opposed to running a single backup agent on the host system.

Installing backup agents on your VMs is the only surefire way to guarantee the reliability of backing up their virtual disk files. Keep in mind, however, that there's more to a VM than just its disk files. Each VM has configuration files too, and these files should be backed up by a backup agent running on the VM's host system.

Note All the major backup vendors, including EMC (Legato), Veritas, CommVault, and Computer Associates, have performed testing at some level with their backup agents and virtual machine software. You should find that the backup agent will behave as if it's running on a physical system, with no awareness of the VM software that's hosting the virtual machine.

If you don't have an enterprise-class backup software suite but still need to run VMs 24/7, then you could use the backup tool included with each OS to back up your running VMs. For example, you could use Windows Backup to back up Windows VMs and could use the `dump` command to back up Linux VMs. Each of these scenarios is described later in the “Performing Non-Agent-Based Backups” section, but first let's take a look at agent-based backups.

Running Backup Agents on VMs

Running backup agents on VMs is the preferred method for backing up virtual disks, according to both Microsoft and EMC. A *backup agent* is software that runs as part of a backup software application, allowing you to back up a system either over the LAN or through a storage network to backup media.

When backing up VMs, several considerations will directly affect backup and recovery performance. When planning your backup infrastructure to include VMs, you have three primary architectural choices:

- Backing up over the LAN
- Backing up to the host
- Backing up directly to storage (local or SAN-attached storage)

Let's first look at the traditional LAN backup configuration.

Backing Up over the LAN

Most organizations today employ LAN-based backups. With this technique, backup data is sent over a network from the system being backed up to a system acting as a media server. The media server provides a physical interface between the systems being backed up and the storage hardware, such as a tape library. Figure 7-1 shows this configuration.

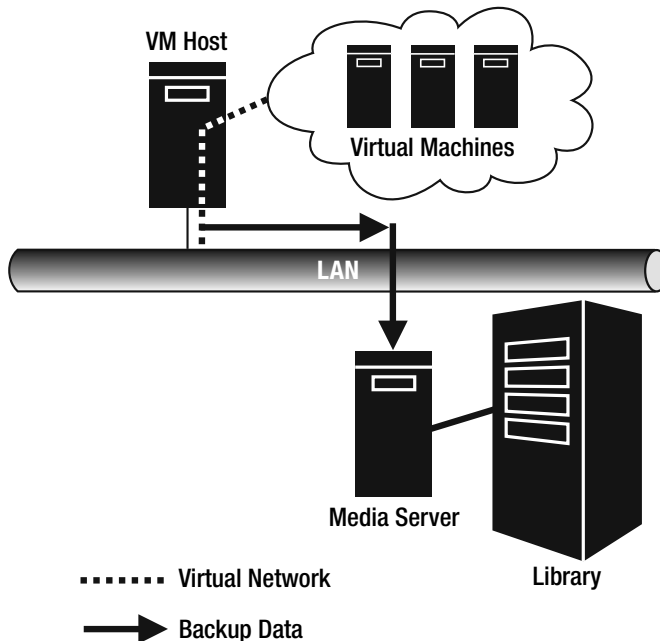


Figure 7-1. *Backing up a VM over the LAN*

With a LAN-based backup architecture, you could either network the VMs directly to the media server using bridged networking or have the VMs connect using a host-only network and route through the host to the media server. If the VMs are routing through a host system, remember that the media server will also need a static route present (or have the LAN router's route table updated) so that the media server can route back to the virtual machine. If routing isn't one of your strengths, then we recommend going with bridged networking to connect your VMs to the media server since this is the easiest scenario to configure.

Caution You can't use NAT to connect VMs on a private network to a media server. While the VMs could communicate to the media server through NAT, the media server wouldn't be able to directly communicate with a VM on its private IP address.

For those of you who want to have their VMs on a private host-only network, then let's consider a sample configuration. Assume the VM being backed up in Figure 7-1 has an IP address of 192.168.2.11/24. Now consider that the VM's host system has IP addresses of 192.168.2.1/24 (virtual interface) and 172.16.1.20/24 (production LAN interface). Finally, let's assume the media server has an IP address of 172.16.1.19/24. For the backups to work between the VM and media server, both systems will need to know how to route to each other. With the VM, this should be easy since you could just assign it the gateway address of 192.168.2.1. However, the media server may already have a default gateway on the production LAN with no knowledge of your private host-only network. If this is the case, you can assign a static route to the media server so that it knows how to reach the VM's host-only network. For the VM to reach the private network, its data will need to go through the production LAN interface of the VM's host system. So, at this point, you have all the information needed for creating a static route on the media server, which includes the following:

- **Destination network:** 192.168.2.0
- **Subnet mask:** 255.255.255.0
- **Gateway address:** 172.16.1.20

If you need a quick refresher on classless interdomain routing (CIDR) notation, remember that it's a means to represent a subnet mask as a single decimal value. So, the /24 that follows each IP address represents the number of consecutive binary 1s in the subnet mask. Since there are 8 bits in an IP address octet and since there are twenty-four 1s, you have three sets of eight consecutive 1s. If you convert eight consecutive 1s (11111111) from binary to decimal, you get 255.

Note If you're still craving more information on binary numbering, routing, and TCP/IP subnetting, point your Web browser to <http://www.learntosubnet.com>.

With this information now in hand, you can configure static routes on the media server. You do this using the `route add` command on either Windows or Linux systems. Here are sample commands to run if the media server is running on either a Windows box or a Linux box:

- **Windows:** `route add -p 192.168.2.0 mask 255.255.255.0 172.16.1.20`
- **Linux:** `route add -net 192.168.2.0 netmask 255.255.255.0 gw 172.16.1.20`

With the potential routing issues now behind you, you should have connectivity between your VM and the media server. If both systems allow ICMP requests, you should be able to run `ping` commands to test for the network connectivity between the two systems (for example, run `ping mediaserver` from the VM).

Tip If your backup software's architecture includes a central management server that initiates and manages all backup and recovery operations, ensure that network connectivity also exists between each VM and the management server.

At this point, you should be ready to install backup agents from your backup software vendor inside the VMs and to perform backups. With this architecture, it's likely that the LAN may become a bottleneck for backup data streams. For example, a 100Mb/sec LAN provides for 12.5MB/sec ($100\text{Mb} \div 8$) of bandwidth, which translates to 45GB per hour ($12.5\text{MB} \times 60\text{ seconds} \times 60\text{ minutes}$). Although this may seem like a pretty large number, it can be significantly reduced if multiple backups are running simultaneously and having to share the bandwidth of the media server's network interface. One alternative to solve a network bottleneck is to upgrade the network to 1Gb/sec or higher bandwidth. Another alternative is to configure the VM host system as a media server. We'll cover that option next.

Backing Up to the Host

When VMs are configured to back up directly to the host, no backup data has to traverse the production LAN. Figure 7-2 illustrates this configuration.

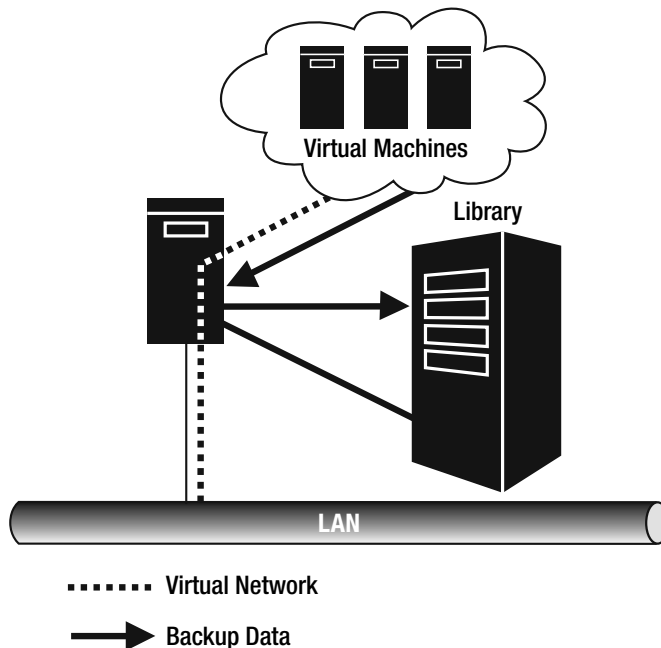


Figure 7-2. Backing up a VM through the VM host

In this configuration, the VM host is configured as a media server in your existing backup infrastructure and connects to physical backup devices such as a tape library either through a SCSI connection or through a storage network. Figure 7-2 shows a library SCSI attached to the VM host.

This configuration offers fast backups when compared to pushing backup data over a production LAN, and it may also allow other LAN-based backups to run faster, with VM backups no longer running over the LAN. The only disadvantage to this approach is that you'll need to connect the VM host to a backup storage device, which may involve purchasing additional storage resources or reallocating existing resources.

With this type of backup architecture, you should first configure the host system as a media server in your existing backup network. We prefer to configure the media server first because the installation of your backup software may require a reboot. Also, you may need knowledge of the media server installation while installing the backup agent software on the VMs. Once you have the media server configured, you can then install the backup agent software on each virtual machine.

Backing Up Directly to Storage

Neither Virtual PC 2004 nor Virtual Server 2005 supports connecting VMs to storage devices such as tape drives or libraries either through a SCSI bus attached to the host or through a SAN. Hard disks are the only device located on a SAN that Virtual Server supports. All other storage devices can't be accessed by Virtual Server 2005 VMs. Basically, this means that if you're hosting VMs using either of the Microsoft virtualization applications, configuring your VMs to perform network-based backups—either internally to the VM host or over the LAN—is your best backup option.

VMware Workstation, GSX Server, and ESX Server all support connecting VMs directly to SCSI or Fibre Channel storage devices, such as tape drives and libraries. You can find more information on using the Generic SCSI Device hardware resource to create this connection in Chapter 6.

With support for other storage devices, you can configure a VM to be its own media server and thus back up directly to a local SCSI or Fibre Channel storage device. Figure 7-3 shows this configuration.

In backing up directly to a storage device such as a tape drive, you don't have any network bandwidth contention, whether on a logical host-only network or over a physical LAN. This approach offers the best possible performance but requires the additional storage resources to allocate to the VMs you want to back up. If your backup platform supports dynamic drive and library sharing in a SAN, then you may find backing up VMs directly to storage on a SAN to be the perfect solution.

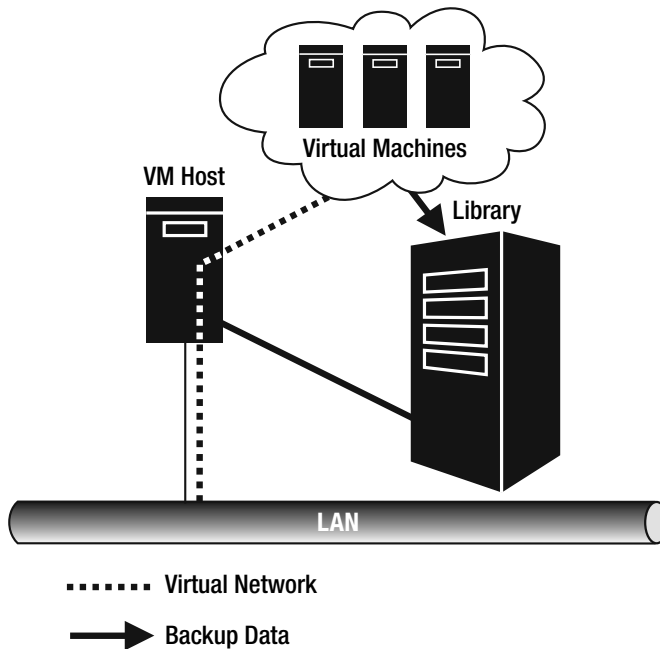


Figure 7-3. *Backing up a VM directly to a storage device*

If you're now sold on using your backup software to back up a VM directly to a storage device, here's the general procedure to make it all work on VMware GSX Server:

1. Add the storage device to the VM as a generic SCSI device.
2. Start the VM, and make sure it can see the storage device.
3. Install your backup application's media server software, and configure the storage device for use with your backup software.
4. Install your backup application's backup agent software on the VM.
5. Back up the virtual machine.

Although installing and running backup agents on VMs protects their virtual disk files, you still need to protect each VM's configuration files. You do this by backing up the host system.

Running Backup Agents on the Host

Running backups on the host system serves two general purposes:

- Secures each VM's configuration files
- Secures the host system's data, drivers, and configuration to prepare for a system failure or disaster

Caution Running an open file agent such as St. Bernard Software's Open File Manager on the host system won't guarantee reliable backup of open virtual disk files and thus should be used with extreme caution. You should perform frequent test restores to verify the validity of a virtual disk file backed up via an open file backup agent.

Caution Windows Server 2003's Volume Shadow Copy Service doesn't support online snapshots of virtual disk files, so this service shouldn't be used as a method of backing up VMware, Virtual PC, or Virtual Server virtual disk files.

In terms of setup, installing a backup agent on a VM host is no different from installing an agent on any other server on your network. With agents securing the data on the VM virtual disk files, remember that you don't even need to have your backup agent attempt to back them up. This means that if your backup agent supports filtering, you can filter out all virtual disk files from the backup, thus preventing the backup agent from even attempting to back up a virtual disk file.

If you don't plan to install backup agents on your VMs, another alternative is to power down the VMs prior to backing up the host. When a VM is powered down, its virtual disk files are no longer locked by the VM application and thus can be backed up safely. Later in the "Performing Flat-File Backups" section, you'll look at ways to automate powering down running VMs so that they can be backed up without needing a backup agent installed on them.

Now that you've seen how to protect VMs using enterprise backup applications, let's look at the tools operating systems have available for stand-alone local backups.

Performing Non-Agent-Based Backups

If you're running VMs in a small or home office, purchasing enterprise-class backup software may not be an option. Instead, you may find that the backup tools included with each VM's operating system are well-suited for supporting your backup needs. In this section, you'll examine tools for backing up the two most popular virtualized operating systems: Windows and Linux.

These tools support the three most popular backup types:

- **Full (normal):** Backup of all files on a disk
- **Incremental:** Backup of all files that have changed since the time of the last backup
- **Differential:** Backup of all files that have changed since the time of the last full backup

Full backups are the most popular backup method for VMs, as they secure every file on a virtual hard disk. If you have a small tolerance for data loss, or if you don't have the media to support daily full backups, you may need to mix incremental or differential backups with the full backups. One popular approach is to run a full backup every Saturday and run incremental backups Monday to Friday. This means that if a virtual disk failed, you'd lose no more than one day's worth of data. If you need even better protection, you could run a weekly full backup on Saturday, differential backups daily (Monday to Friday), and incremental backups every four hours. This means that the most data ever lost would be four hours. The drawback to this approach, however, is that it would involve running backup jobs (the incrementals) during business hours, which can slow down server performance during the time of the backup. In this approach, you'd need to restore that last full backup, the last differential backup, and every incremental backup that occurred after the time of the last differential. Since a differential backup copies all files that have changed since the time of the last full backup, any earlier (prior to the time of the last differential) differential or incremental backups wouldn't be needed for recovery.

With backup strategy, there really is no single answer. You have to decide on your organization's tolerance for data loss and use that as a baseline to determine the frequency and types of backups to run on your VMs.

With a baseline for backup strategy under your belt, you'll now learn how to use Windows Backup to secure data.

Caution If you're using a local tool to back up your VMs or host system, use extreme caution when storing backup data in a single file on the host system. If the host system's hard disks are corrupt or become damaged, you'll lose not only your original data but your backup as well! A best practice when backing up to a file is to store the backup file on another server by specifying either a UNC path or a network file system (NFS) mount point as the location for the backup file.

Using Windows Backup

Windows Backup (formerly called NT Backup), shown in Figure 7-4, is a tool that has been included with the Windows operating systems as far back as Windows NT. You can run this tool as either a GUI application or a command-line utility; it supports backing up data to removable media devices such as tapes or to a single file on a hard disk. This flexibility gives you plenty of options when backing up VMs.

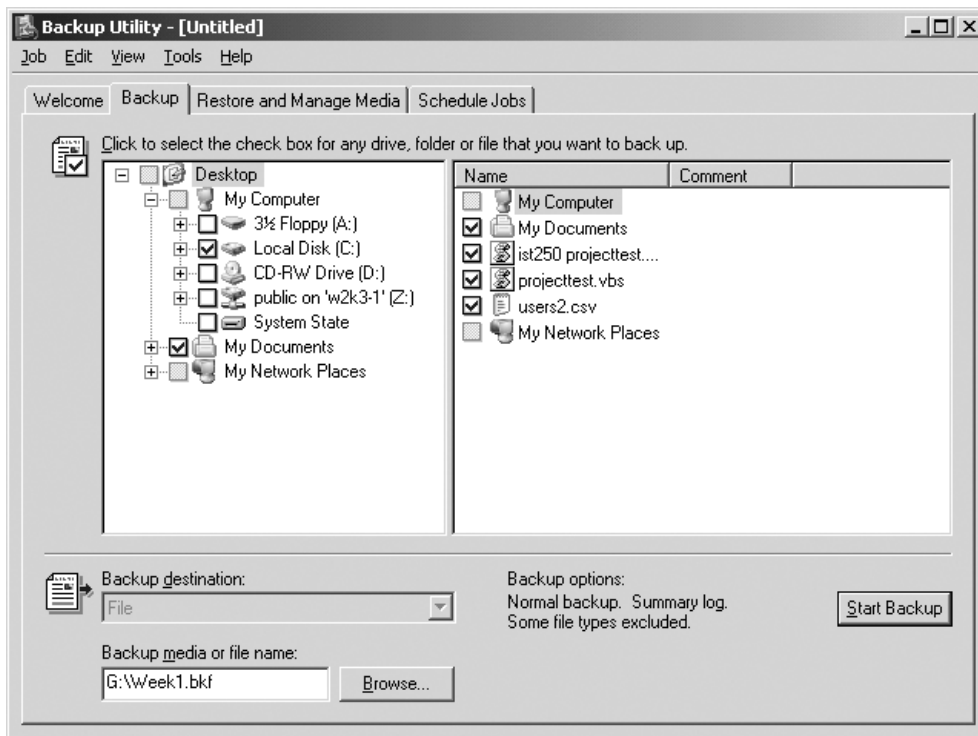


Figure 7-4. *Windows Backup*

Note The procedures outlined in this section are for a Windows Server 2003 virtual machine. The procedures for other Windows operating systems will be similar but may not be exactly the same.

To perform a backup with Windows Backup, follow these steps:

1. To open Windows Backup, click Start ► All Programs ► Accessories ► System Tools ► Backup.
2. If the tool opens in wizard mode, you'll need to answer a series of questions to start the backup. If the tool doesn't open in wizard mode, proceed to step 3.
3. Once the tool opens, click the Backup tab.
4. Next, select the disks (or specific data) to back up. You do this by checking the box that corresponds to the data to be backed up. You'll see that the GUI behaves similarly to Windows Explorer.

5. Now browse to or enter a destination for the backup data in the Backup Media or File Name field.
6. When you're satisfied with your selections, click the Start Backup button.
7. To begin the backup, verify that all your selections are accurate in the Backup Job Information dialog box, and then click the Start Backup button. If you want the backup job to run at a later time or on a recurring basis, you can click the Schedule button at this time to schedule when the job should run.

Tip To back up all the data on a virtual machine, in wizard mode you'd select the Back Up Everything on This Computer option. If the tool starts in standard mode, click the Backup tab and then select the My Computer check box. At a minimum, you should always back up all pertinent data drives, the boot drive, and the system drive, as well as the system state.

With the default options selected, Windows Backup will always perform a full backup. If you want to perform either an incremental or a differential backup, in step 7 you'd need to click the Advanced button in the Backup Job Information dialog box and then select the appropriate backup type in the Backup Type drop-down menu.

You can use the same tool to restore data by following these steps:

1. Open Windows Backup by clicking Start ► All Programs ► Accessories ► System Tools ► Backup.
2. If the tool opens in wizard mode, you'll need to answer a series of questions to start the restore. If the tool doesn't open in wizard mode, proceed to step 3.
3. Once the tool opens, click the Restore and Manage Media tab.
4. Next, expand the appropriate media item (such as a file or tape), and select the data to restore (see Figure 7-5).
5. Now click the Start Restore button.
6. In the Confirm Restore dialog box, click OK. The restore job should run and complete.

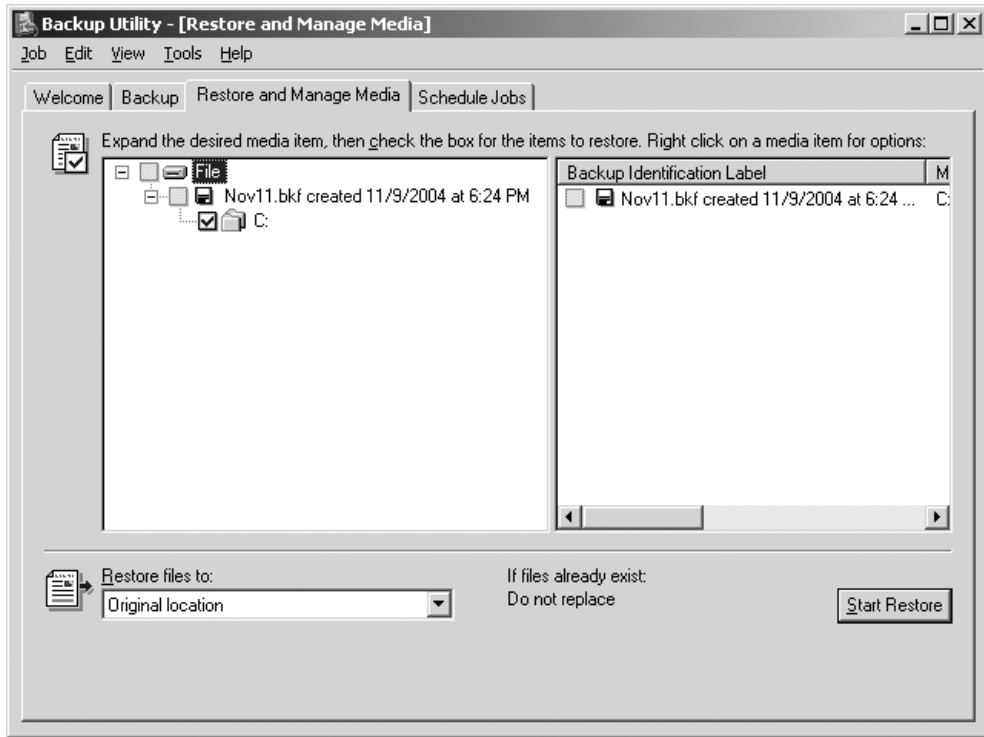


Figure 7-5. *Selecting data to restore using Windows Backup*

At this point, you have the general procedures to use Windows Backup to back up and recover data stored on VMs. Keep in mind that you can still use two of the architectures mentioned in the previous section as methods for saving backup data. For example, you could store backup data in a file saved on a mapped network drive located on another physical server on the network. This will allow you to get your VM's backup data off the VM host. Since one of the primary purposes of backing up data is to protect against system failure, it's never a good idea to store a VM's backup data on its host system.

One other alternative with Windows Backup is to add a generic SCSI storage device to your VM (VMware only) and back up the VM directly to the storage device. In this scenario, you'll need to carefully manage backup media to ensure that the host system doesn't try to use the same tapes being used by a VM. Otherwise, one system may overwrite backup data generated by another system.

Now that you've seen the possibilities with Windows Backup, next you'll look at using `dump` and `tar` to back up Linux volumes. If you're interested in running Windows Backup from the command line, you can refer to the "Performing Flat-File Backups" section later in this chapter.

Backing Up Linux File Systems

The `dump` command has been around practically forever, dating back to early Unix operating systems (beginning with AT&T Unix version 6). This command allows you to perform full and incremental backups to a local storage device or volume or to a remote mount point. `tar` is

another popular tool that can package files and folders for transferring over a network and for performing backups. In the following sections, we'll cover how to back up your Linux VMs using either `dump` or `tar`. Also, you'll see how to use the cron daemon to create an automated backup schedule for your Linux VMs. Let's get started by looking at `dump`.

Backing Up with `dump`

Listing 7-1 is a shell script that uses `dump` to perform a full backup of a VM named W2KFS1.

Listing 7-1. *Backing Up with `dump`*

```
#!/bin/sh
#
# Filename: LinuxVmDumpBackup.sh
# Purpose: Uses dump to back up a VM to Tape Drive 0. This script performs
# a full backup of a VM's hard disks.
#=====
# ==== Assign variables ====
DriveToBackup="/dev/hda2"      # Name of drive to back up
BackupDestination="/dev/nrft0" # tape drive 0, tape not rewound after bu

# ==== Run the backup=====
/sbin/dump 0uf $BackupDestination $DriveToBackup
```

In the script, the `DriveToBackup` variable identifies the drives to be backed up, and the `BackupDestination` variable defines the target backup device. This script will perform full backups of a virtual machine. You can refine this script to perform incremental backups following a full backup by changing the `dump` level specified in the `dump` command. A level of 0 indicates a full backup. The values 1 to 9 in the command sequence indicate incremental levels. For example, the following are the script modifications necessary to run a full backup on Sunday and incremental backups from Monday to Friday:

- **Sunday full:** `/sbin/dump 0uf $BackupDestination $DriveToBackup`
- **Monday incremental:** `/sbin/dump 1uf $BackupDestination $DriveToBackup`
- **Tuesday incremental:** `/sbin/dump 2uf $BackupDestination $DriveToBackup`
- **Wednesday incremental:** `/sbin/dump 3uf $BackupDestination $DriveToBackup`
- **Thursday incremental:** `/sbin/dump 4uf $BackupDestination $DriveToBackup`
- **Friday incremental:** `/sbin/dump 5uf $BackupDestination $DriveToBackup`

With this approach, you'd have six versions of the script with the only difference in each version being a single line in the script. After the Friday backup completes, you'd want to insert a new tape in the drive to be used by the next backup cycle, starting on Sunday.

To restore data backed up with `dump`, you need to use the `restore` command. To perform a complete restore of all data on a tape, use the following syntax:

```
restore rf <backup device>
```

Following the earlier example, run this command:

```
restore rf /dev/nrft0
```

Another popular method of using `restore` is to run the command in interactive mode. To run a restore in interactive mode, you use this syntax:

```
restore if <backup device>
```

In the previous backup example, you'd run `restore if /dev/nrft0` to start the interactive restore.

Using this approach, the command will show you an index of data backed up on a tape and allow you to select the files and folders you want to restore. When looking to restore just a few files or folders, this method is popular. Let's look at an example of an interactive-mode restore. Assume you need to restore a file named `contactinfo.doc` from the `/data/marketing` folder. Prior to running the restore operation, first navigate to the directory in which you want the files restored to; in this example, you'd run `cd /data/marketing`. Next, insert the tape that contains the data to be restored into the tape drive. Finally, run the following commands to recover the file:

```
#restore if /dev/nrft0
restore >ls
.:
data/          misc/
restore >cd data
restore >ls
./data:
marketing/      /sales          /service        /support
restore >cd marketing
restore >ls
./marketing:
brochure.doc    contactinfo.doc  suggestionlist.doc
restore >add contactinfo.doc
restore >ls
./marketing:
brochure.doc    *contactinfo.doc  suggestionlist.doc
restore >extract
```

Notice that after you add the `contactinfo.doc` file to the recovery list using the `add` command, you ran the `ls` command in the `marketing` folder, and the selected file (`contactinfo.doc`) was preceded with an asterisk (*). This indicates the file has been selected for recovery. Once you run the `extract` command, all files selected for recovery will be restored at that time.

While `dump` has been a popular backup tool, `tar` has significantly grown in popularity, primarily because of its reliability, ease of use, and flexibility. Next, we'll cover how to use `tar` to protect your VMs.

Backing Up with tar

`tar` is another popular tool for backing up data. With `tar`, you can back up a VM's contents into a single file. Listing 7-2 shows a script that uses `tar` to back up a VM's root directory and associated subdirectories to an NFS mount point.

Listing 7-2. *Backing Up with tar*

```
#!/bin/sh
#
# Filename: LinuxVmTarBackup.sh
# Purpose: Uses tar to back up a VM to an NFS mount point. This script performs
# a full backup of the root directory and filters out /tmp, /proc, and /mnt.
#=====
# ==== Assign variables ====
VMName="FS1"      # Name of VM to back up
Today=$(date +%m-%d-%y) #Loads current date as mm-dd-yyyy into variable
DirToBackup="/"   #Specifies to back up root.
NfsPath="mediasrv1:/data/Backup" #Specifies NfsPath to back up to
#Name of backup file. Includes date stamp.
BackupDestination="/mnt/backup/"$VMName-"$Today".tar"
#Name and location of backup log file
BackupLog="/mnt/backup/"$VMName"-Backup_Log_"$Today".log"
#Directories to exclude from backup
ExcludeList="--exclude /proc --exclude /mnt --exclude /tmp"
#==== Mount backup folder ====
mount -t nfs $NfsPath /mnt/backup

# ==== Run the backup=====
tar -cpvf $BackupDestination $DirToBackup $ExcludeList >&$BackupLog

#==== Unmount backup folder ====
umount /mnt/backup
```

In this script, you back up the root (/) folder while excluding the /tmp, /proc, and /mnt folders. The NFS mount point is mounted to /mnt/backup. This folder must be created before the script runs for the first time. The output backup file that's created will include information on the VM's name as well as the date that the backup was performed. This is useful if you intend to back up several VMs to the same location. In the sample `LinuxVmTarBackup.sh` script, if the backup was executed on December 20, 2004, the output file would be named `FS1-12-20-2004.tar`. The backup's associated log file would be named `FS1-Backup_Log_12-20-2004.log`.

To restore a tar backup file, you use the extract (x) option in place of the c option. For example, to extract a tar backup archive back to the root (/) directory, you run the following command:

```
tar xpvf /mnt/backup/FS1-12-20-2004.tar --overwrite
```

This command extracts the `FS1-12-20-2004.tar` archive to the root folder. The `--overwrite` option causes the tar command to overwrite like files during the restore operation.

Now that you've looked at both tar-based and dump-based backups and restores, we'll cover how to automate the backup process with cron.

Note Another excellent free Linux backup tool is the Advanced Maryland Backup Disk Archiver (AMANDA), which automates network-based backups of Linux hosts to a central media server. This is similar to the approach for backing up using backup agents. You can download a free copy of AMANDA and read more about this excellent product at <http://www.amanda.org>. In Chapter 14, we'll show you an example of using AMANDA to configure backups to run across several systems.

Automating Backups with cron

By using the cron daemon, you can automate the execution of scripted backups. This gives you the ability to set up your backup schedule and let it take care of itself. Jobs executed by cron are stored in the `/etc/crontab` file. By default, the file contains the following lines:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

In this file, the purpose of the first four lines is to declare variables that establish cron's working environment. The run-parts section is where you can schedule tasks to run. By default, hourly, daily, weekly, and monthly tasks are already configured. Any scripts stored in the `/etc/cron.hourly` folder will run on the first minute of each hour, and scripts in the `/etc/cron.daily` folder run each day at 4:02 a.m., by default. Using these folders gives you an easy means to have several scripts execute simultaneously with a single line of code in the crontab file. To understand how the time and date settings are configured, let's examine the crontab file format illustrated in Figure 7-6.

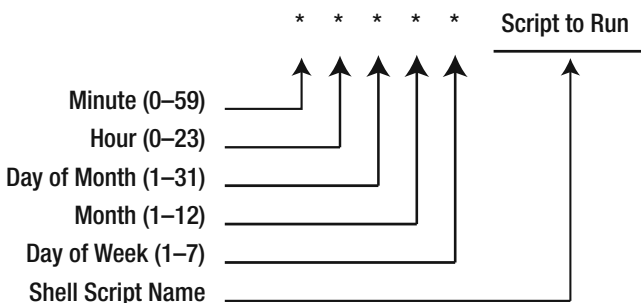


Figure 7-6. *crontab* file text format

The first five positions (separated by spaces) in a line of the crontab file specify the day and time to run the script, and the sixth position indicates the script to be executed. The * wildcard can represent all instances of a value. So, if you wanted a backup script job to run at 1 a.m. every day, you'd use the values 0 01 * * *. With the day of the week parameter, the value 1 represents the first day of the work week, which is Monday. Sunday equals 7.

Now let's consider the required line in the crontab file that's needed to execute the LinuxVmTarBackup.sh script. To perform this task every Friday night at 11 p.m., you'd add this line:

```
0 11 * * 5 /root/scripts/LinuxVmTarBackup.sh
```

Think you've got it now? Take a look at the last default line in the crontab file shown earlier. When do the scripts located in the /etc/cron.monthly folder run? If you want to test yourself, don't read the next sentence until you have an answer! Okay, if you came up with the first day of each month at 4:42 a.m., you're correct!

As you can see, cron can be invaluable in automating the backups of your VMs.

Tip cron is a powerful tool, and in this section we merely scratched the surface of its capabilities. For more information on cron, point your Web browser to <http://www.redhat.com/docs/manuals/enterprise> to download the Red Hat Enterprise Linux System Administration Guide.

Performing Flat-File Backups

If you don't plan to install backup agents on your VMs and don't plan to run local independent backups on each VM, then performing flat-file backups is your best alternative. The technique of flat-file backups originated as a way to back up databases without having to use a backup agent. When a flat-file backup is performed on a database, the database first shuts down (is taken to an offline state), and then its related files are backed up. Shutting down the database prior to backup guarantees that you'll be able to back up all the data contained within the database's related files. When a database is online, the database application locks the database's related files so that normal backups aren't possible. The only reliable way to back up an online database is by using a backup agent that's supported by the database application.

So, if you relate a VM's disk and configuration files to a database's data and log files, you have the same problem with backups as you would with databases. You can't just run a backup of the host system and expect to reliably back up running VMs. Not only does each VM application lock virtual machine files while VMs are running but also the VMs use physical memory as a buffer for writes to virtual disk files. This means that not all the data that a VM thinks is inside a virtual disk is actually in the virtual disk file on the host. So, backing up a virtual disk file while a VM is running could mean that you're backing up a corrupted disk, which will be useless when you attempt a restore.

Now that you've seen why you shouldn't try to back up a running VM directly from the host, let's look at the procedure for running a flat-file backup. To perform this backup, you must follow three general steps:

1. Shut down the VM.
2. Back up the VM's related files.
3. Start up the VM.

Normally, all VM files reside in a single folder, so backing up the offline VM should include simply making a copy of a VM's folder.

Caution If a VM is configured to map one of its hard disks to a physical disk on the host system, you must be sure to include the disk in the VM's backup definition. For example, if a VM's configuration and a single virtual disk file are stored in the `E:\VMs\NT4Srv` folder but the VM also maps a disk to the host's F drive, you must back up the F drive on the host as well as the `E:\VMs\NT4Srv` folder when the backup job runs.

If you're running VMs in production, it's likely you'll want the VM to be offline for as little time as possible. With this in mind, the easiest way to automate a VM shutdown, backup, and startup is by using a script. In the following sections, we'll cover how to script flat-file backups of the following VM configurations:

- VMware Workstation on Windows
- VMware Workstation on Linux
- VMware GSX Server on Windows
- VMware GSX Server on Linux
- Microsoft Virtual PC 2004
- Microsoft Virtual Server 2005

Each section details scripts for automating the backup process, so you can just turn to the section that's right for you, depending on the virtual machine applications in your environment. We'll start by looking at scripting flat-file backup jobs for VMware Workstation.

Tip If you want to schedule Windows batch scripts to run on a recurring basis, you can do so using the Scheduled Tasks system tool. This tool allows you to select a schedule pattern in which to run a job and also gives you the ability to supply user credentials under which to execute the job. This prevents you from having to code the user credentials inside the script.

Running VMware Workstation Flat-File Backups

Workstation VMs lock their configuration file while a VM is open, so to fully back up a VMware VM, you must not only power down the VM but also close the specific VM you plan to back up. Lucky for you, VMware Workstation provides an easy method for automating this. To configure a VM to automatically close once it's powered down, perform the following steps:

1. In the Workstation UI, click the VM you plan to back up. Then click the VM menu, and select Settings.
2. In the Virtual Machine Settings dialog box, click the Options tab.
3. Under the Options tab, click the Power setting, and then check the Close After Powering Off or Suspending box (see Figure 7-7).
4. Click OK to close the Virtual Machine Settings dialog box.

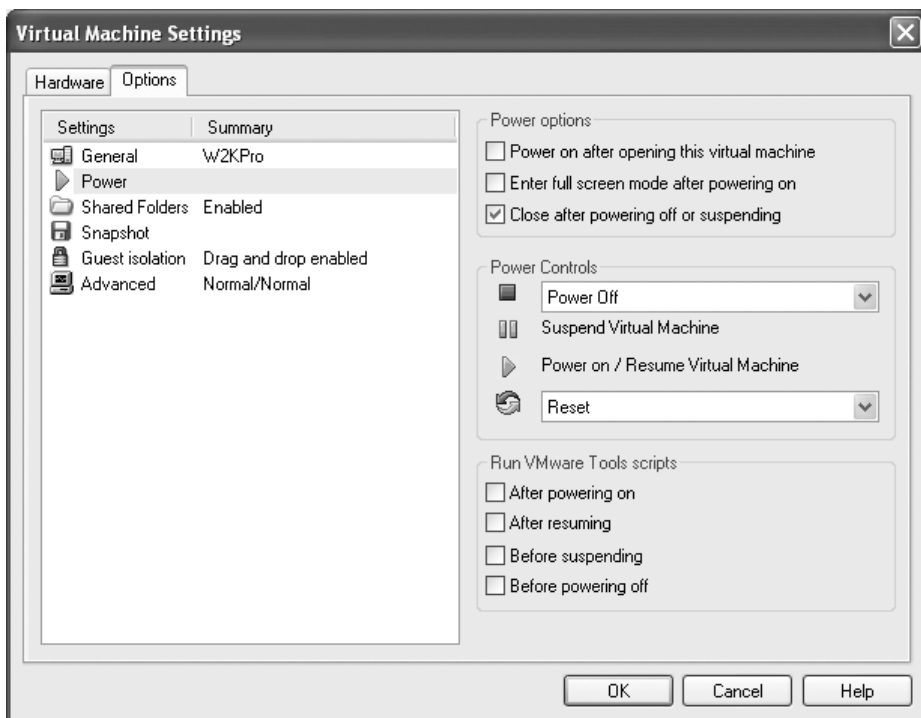


Figure 7-7. *Selecting for the VM to automatically close when powered down*

In the next sections, we'll begin by looking at backing up Workstation VMs running on Windows operating systems and finish with a look at backing up Workstation running on Linux.

Running VMware Workstation on Windows Flat-File Backups

One of the easiest methods to script backup jobs on Windows operating systems is using tried-and-true batch scripting. We've selected this method for the examples in this book because of its ease of use and overall adoption in the field. Since many of the administrators we encounter run both Windows and Linux operating systems inside VMs, we'll provide examples of scripting flat-file backups for both operating systems. In each example, backups are executed using the `ntbackup.exe` command-line utility. Both of the batch files covered use a core set of variables to establish a source and destination for the backup. For your own use, you need to change the variables to suit your specific environment. Here's a quick overview of each of the variables that will be used:

- **VMName:** Host name of VM to be backed up
- **VMFolder:** Location of VM's files on the host system
- **VMXPath:** Full path to VM's configuration file
- **Today:** Current date formatted as mm-dd-yyyy (used to date stamp backup files)
- **BackupFolder:** Backup data destination folder (create this folder prior to running backup)
- **BackupFile:** Filename of backup job (stamped with current date)
- **JobName:** Friendly name for job (stored in backup file)
- **Description:** Description of job (stored in backup file)

With the variables now under your belt, let's look at backing up a Windows VM running on a Windows host.

Backing Up a Windows VM

Listing 7-3 shows a batch file that can be used to automatically shut down a Windows virtual machine, back up its files, and then start it up.

Listing 7-3. *Backing Up a Windows VM on a Windows Host*

```
:: Filename: WinVMWSwinBackup.bat
::
:: Purpose: Automates the backup of a virtual machine by shutting down the VM,
:: using ntbackup.exe to back up the VM's files, and then automatically starting
:: back up the VM. This script backs up a VM named W2KFS, which is located in
:: the E:\VMs\W2KAS folder. You would need to alter the VM name and paths for
:: this script to run in your environment.
::=====
@echo off
:: Set Variables
set VMName=W2KFS
set VMFolder= E:\VMs\W2KAS
```

```

set VMXPath= E:\VMs\W2KAS\W2KFS.vmx
set Today=%Date:~4,2%-~Date:~7,2%-~Date:~10,4%
set BackupFolder=G:\Backup\%VMName%
set BackupFile=%BackupFolder%\%Today%.bkf
set JobName=%Today% Full Backup
set Description=Starting %Today%

:: Stop VM Using shutdown.exe (requires UNC path to VM)
:: On W2K and earlier hosts, shutdown.exe requires installation of the resource kit
shutdown.exe /m \\%VMName% /s /t 0

:: Wait for VM to shut down (uses ping -n <seconds> to the loopback address
:: to create a delay)180 seconds = 3 minute delay.
ping -n 180 127.0.0.1 > nul

:: Back up VM
IF NOT EXIST %BackupFolder% md %BackupFolder%
ntbackup backup "%VMFolder%" /J "%JobName%" /D "%Description%" /F "%BackupFile%"

:: Start VM (path to VMware.exe file must be specified)
"D:\Program Files\VMware\VMware Workstation\VMware.exe" -x "%VMXPath%"

```

In this example, a VM is backed up to a local drive on the system. If you wanted to back up the VM to a UNC path, you'd need to change the BackupFolder variable. For example, to direct backup data to the \\Max\Backups share, you edit the BackupFolder variable line so that it reads `set BackupFolder=\\Max\Backups\%VMName%`.

Notice that the ping command is used in the script for the purpose of creating a delay. The `-n` parameter for the ping command specifies the number of echo requests to send. Each echo request will take approximately one second, so using this command gives you a simple means to insert a delay into the script. An alternative to using ping is to use the Sleep.exe resource kit tool.

After the VM shuts down and the batch script waits the allotted time, the ntbackup.exe command executes to back up the VM. Control isn't returned to the batch file until ntbackup.exe completes the backup job, so there's no need to insert a delay into the batch file following the ntbackup.exe command. The last line of the batch file uses the VMware.exe command to start the VM. You'll see almost identical logic in the next script, which will perform a flat-file backup of a Linux VM.

Backing Up a Linux VM

The toughest aspect of scripting a Linux VM flat-file backup on a Windows host is having to gracefully shut down the VM prior to backup. To do this, you must send a remote shutdown command to the Linux VM. To do this securely, we decided to use one of the many available secure shell (ssh) tools available for Windows operating systems.

We selected the PenguinNet shareware software to perform the remote ssh connection and shutdown command. You can download this tool from <http://www.download.com>. With PenguinNet, you can create a profile that includes all the ssh connection settings and commands to run during the ssh connection. This allows you to fully script a remote shutdown in

a batch file without having to include any confidential information in the batch file. For example, the username, password, system name to connect to, and commands to run are all embedded in the profile.

In PenguinNet, you can create connection profiles using the connection profiles toolbar in the PenguinNet GUI. Figure 7-8 shows the connection profile used in our WinVMWSLinuxBackup.bat file. When you click the Send Commands button in the Connection Profiles dialog box, you can enter commands to automatically run once the session is established. Figure 7-9 shows these settings.

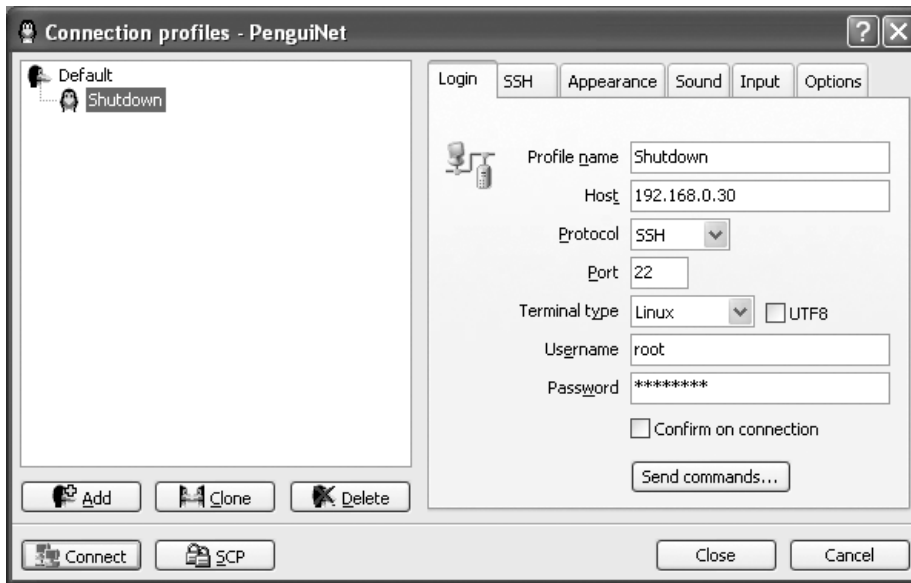


Figure 7-8. PenguinNet shutdown profile settings

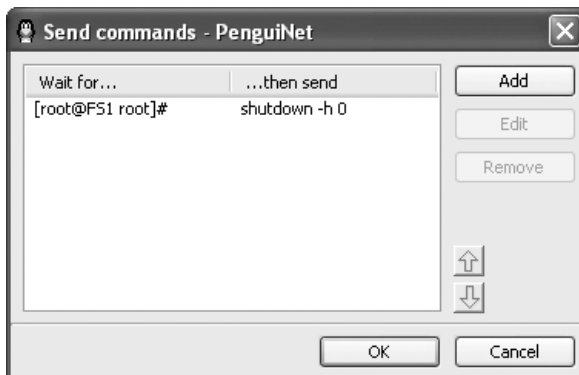


Figure 7-9. PenguinNet shutdown profile Send Commands settings

Once you've created the profile, you must turn the profile into a shortcut so the batch file can use it. You can create a shortcut to the profile by simply dragging the profile from the PenguinNet GUI and dropping it into any folder in Windows Explorer. Once this is done, you can use the command referenced by the profile shortcut in the batch file. Listing 7-4 shows a batch file that remotely shuts down a Linux VM, backs it up, and then restarts the VM.

Listing 7-4. *Backing Up a Linux VM on a Windows Host*

```
:: Filename: WinVMWSLinuxBackup.bat
::
:: Purpose: Automates the backup of a Linux virtual machine by shutting down
:: the VM, using ntbackup.exe to back up the VM's files, and then automatically
:: starting back up the VM. This script backs up a VM named LinuxFS1, which is
:: located in the E:\VMs\LinuxFS1 folder. You would need to alter the VM name
:: and paths for this script to run in your environment.
::=====
@echo off
:: Set Variables
set VMName=LinuxFS1
set VMFolder=E:\VMs\LinuxFS1
set VMXPath=E:\VMs\LinuxFS1\LinuxFS1.vmx
set Today=%Date:~4,2%-~7,2%-~10,4%
set BackupFolder=G:\Backup\%VMName%
set BackupFile=%BackupFolder%\%Today%.bkf
set JobName=%Today% Full Backup
set Description=Starting %Today%

:: Stop VM Using PenguinNet.exe to issue shutdown command
"D:\Program Files\PenguinNet\PenguinNet.exe" profnum://4278190084

::Wait for VM to shut down (uses ping -n <seconds> to the loopback address
:: to create a delay) 180 seconds = 3 minute delay.
ping -n 180 127.0.0.1 > nul

:: Back up VM
IF NOT EXIST %BackupFolder% md %BackupFolder%
ntbackup backup "%VMFolder%" /J "%JobName%" /D "%Description%" /F "%BackupFile%"

:: Start VM (path to VMware.exe file must be specified)
"D:\Program Files\VMware\VMware Workstation\VMware.exe" -x "%VMXPath%"
```

As you can see, this batch script is almost identical to the Windows VM backup script. The primary difference with this script is the methodology used to remotely shut down the Linux VM. Since the Windows shutdown.exe command couldn't be directly run against the Linux VM, we launched PenguinNet to run a remote shutdown command via an ssh connection. Next, we'll cover backing up VMs running on a Linux host system.

Running VMware Workstation on Linux Flat-File Backups

In this section, we show how to use shell scripts to automate the backups of VMs running on a Linux host running VMware Workstation. As with the previous section, this script will shut down the VM, back it up, and then restart the VM.

Here's a quick overview of each of the variables that will be used:

- **VMName:** Host name of VM to be backed up
- **Today:** Current date formatted as mm-dd-yyyy (used to date stamp backup files)
- **DirToBackup:** Location of virtual machine's files on the host system
- **VMXPath:** Full path to VM's configuration file
- **NfsPath:** Backup data destination folder (create this folder prior to running backup)
- **BackupDestination:** Path and filename of backup job (stamped with current date)
- **BackupLog:** Name and location of tar backup log file

Listing 7-5 shows the shell script that will shut down a Windows VM, back it up to an NFS mount point using tar, and then restart the VM. The shutdown command is issued via a remote ssh connection from the host to the VM. Since Windows doesn't natively support ssh, you'll need to install it on the Windows VM. Since ssh is natively supported by Linux, this script can also be used to back up a Linux VM running on a Linux host.

Tip You can download the free OpenSSH ssh server application for Windows at <http://sshhwindows.sourceforge.net>. OpenSSH is also packaged for Windows Services for Unix and can be downloaded at <http://www.interopsystems.com>.

Listing 7-5. Backing Up a Windows VM on a Linux Host

```
#!/bin/sh
#
# Filename: LinuxVmWSBackup.sh
# Purpose: Uses ssh and shutdown.exe to remotely shut down a Windows VM. Then uses
# tar to back up the VM. When the backup completes, the VM is restarted.
#=====
# ==== Assign variables ====
VMName="W2KFS1"      # Name of VM to back up
Today=$(date +%m-%d-%y) #Loads current date as mm-dd-yyyy into variable
DirToBackup="/data/vms/W2KFS1" #Specifies to back up W2KFS1 folder
VMXPath="/data/vms/W2KFS1/W2KFS1.vmx"
NfsPath="mediasrv1:/data/Backup" #Specifies NfsPath to back up to
#Name of backup file. Includes date stamp.
```

```

BackupDestination="/mnt/backup/"$VmName-"$Today".tar"
#Name and location of backup log file
BackupLog="/mnt/backup/"$VmName"-Backup_Log_"$Today".log"
#==== Shutdown VM ====
ssh $VMName shutdown -h 0

#==== Mount backup folder ====
mount -t nfs $NfsPath /mnt/backup

# ==== Run the backup=====
tar -cpvf $BackupDestination $DirToBackup >&$BackupLog

#==== Unmount backup folder ====
umount /mnt/backup

#==== Restart VM ====
/usr/bin/vmware.exe -x $VMXPath

```

You may have noticed that this script is similar to the one used in the “Backing Up with tar” section of this chapter. The primary differences with the script are the lines added to shut down and start the VM. To stop the VM, an ssh connection is established to issue the shutdown command. To restart the VM, the `vmware.exe` command-line tool is used. The `-x` parameter starts the VM specified in the `$VMXPath` variable.

Now that we’ve covered VMware workstation backups, we’ll cover VMware GSX Server.

Running VMware GSX Server Flat-File Backups

VMware GSX Server products, being built for production, offer much more to aid in backing up your VMs. To begin with, GSX Server allows you to configure VMs to gracefully shut down when they’re stopped. All that’s required is to install the latest version of VMware Tools in the VM and to configure the VM to shut down the guest when powering down. To configure the VM power-off settings, you need to perform these steps prior to running any scripted backups:

1. In the VMware Virtual Machine Console UI, click the VM you plan to back up. Then click the VM menu, and select Settings.
2. In the Virtual Machine Settings dialog box, click the Options tab.
3. Under the Options tab, click the Shutdown drop-down menu, and select the Shut Down Guest power-off option. Then check the Close After Powering Off or Suspending box. Figure 7-10 shows both of these settings.
4. Click OK to close the Virtual Machine Settings dialog box.

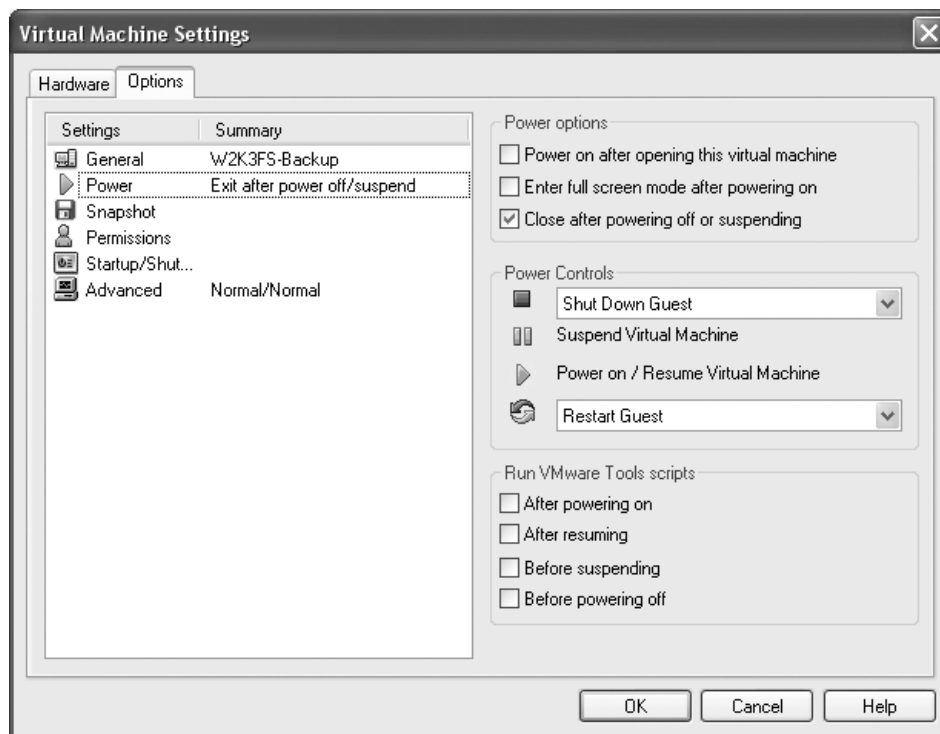


Figure 7-10. *Selecting GSX VM power-down options*

Notice that the previous steps also had you configure the VM to close when powered down. This is required to unlock all VM files so that they can be backed up. Once the VM options are set, all that's left is to put together a script to automate its backup. In the next two sections, you'll see how to script backups on both Windows and Linux hosts.

Running VMware GSX on Windows Flat-File Backups

Since GSX Server can gracefully shut down either a Windows VM or a Linux VM with a single command, there's no difference between scripting a backup for either. Prior to looking at the backup script, let's first examine the purpose of each variable that's used:

- **VMName:** Host name of VM to be backed up
- **VMFolder:** Location of virtual machine's files on the host system
- **VMXPath:** Full path to VM's configuration file
- **Today:** Current date formatted as mm-dd-yyyy (used to date stamp backup files)
- **BackupFolder:** Backup data destination folder (create this folder prior to running backup)
- **BackupFile:** Filename of backup job (stamped with current date)
- **JobName:** Friendly name for job (stored in backup file)
- **Description:** Description of job (stored in backup file)

The script uses the VMware-cmd.exe tool, which is located by default in the Program Files\VMware\VMware VmPerl Scripting API folder on the host system. The syntax of VMware-cmd used in this section's script is as follows:

```
VMware-cmd <path to vmx file> <start | stop>
```

As you can see, the syntax is relatively simple. Following the command, you need to specify the path to the VM's configuration file and then use the start or stop parameter. Now that you've examined the unique aspects of this script, refer to the script itself (see Listing 7-6).

Listing 7-6. Backing Up a VM on a Windows GSX Server Host

```
:: Filename: WinVMGSXBackup.bat
::
:: Purpose: Automates the backup of a virtual machine by shutting down the VM,
:: using ntbackup.exe to back up the VM's files, and then automatically starting
:: back up the VM. This script backs up a VM named W2KFS, which is located
:: in the E:\VMs\W2KAS folder. You would need to alter the VM name and paths
:: for this script to run in your environment.
::=====
@echo off
:: Set Variables
set VMName=W2KFS
set VMFolder= E:\VMs\W2KAS
set VMXPath= E:\VMs\W2KAS\W2KFS.vmx
set Today=%Date:~4,2%-%Date:~7,2%-%Date:~10,4%
set BackupFolder=G:\Backup\%VMName%
set BackupFile=%BackupFolder%\%Today%.bkf
set JobName=%Today% Full Backup
set Description=Starting %Today%

:: Stop VM Using VMware-cmd
"D:\Program Files\VMware\VMware VmPerl Scripting API\VMware-cmd" %VMXPath% stop

::Wait for VM to shut down (uses ping -n <seconds> to the loopback address
:: to create a delay) 180 seconds = 3 minute delay.
ping -n 180 127.0.0.1 > nul

:: Back up VM
IF NOT EXIST %BackupFolder% md %BackupFolder%
ntbackup backup "%VMFolder%" /J "%JobName%" /D "%Description%" /F "%BackupFile%"

:: Start VM Using VMware-cmd
"D:\Program Files\VMware\VMware VmPerl Scripting API\VMware-cmd" %VMXPath% start
```

Compared to VMware Workstation, scripted flat-file backups are much simpler with GSX Server. The same holds true for GSX Server running on a Linux host, which you'll look at next.

Running VMware GSX on Linux Flat-File Backups

Scripting GSX Server flat-file backups on Linux hosts is almost identical in theory to Windows hosts. In fact, both use the `vmware-cmd.exe` tool to facilitate the backup. The only major difference is in the tool used to back up the data itself (tar with Linux and `ntbackup` with Windows). Before getting to the script itself, we'll briefly define its variables:

- **VMName:** Host name of VM to be backed up
- **Today:** Current date formatted as mm-dd-yyyy (used to date stamp backup files)
- **DirToBackup:** Location of virtual machine's files on the host system
- **VMXPath:** Full path to VM's configuration file
- **NfsPath:** Backup data destination folder (create this folder prior to running backup)
- **BackupDestination:** Path and filename of backup job (stamped with current date)
- **BackupLog:** Name and location of tar backup log file

This section uses the `vmware-cmd` tool to stop the VM prior to backing up and restarting the VM once the backup completes. Its syntax is identical to the Windows version of the tool. So, for example, to stop a VM named Mail1, you'd run `vmware-cmd /data/VMs/W2KMail1/mail1.vmx stop`. Aside from this command, you should see that the format and syntax of this script, shown in Listing 7-7, resembles that of the tar backup script presented earlier in the chapter in Listing 7-5.

Listing 7-7. Backing Up a VM on a Linux GSX Server Host

```
#!/bin/sh
#
# Filename: LinuxVmGSXBackup.sh
# Purpose: Uses vmware-cmd to remotely shut down a Windows VM. Then uses tar
# to back up the VM. When the backup completes, the VM is restarted.
#=====
# ==== Assign variables ====
VMName="W2KFS1"      # Name of VM to back up
Today=$(date +%m-%d-%y) #Loads current date as mm-dd-yyyy into variable
DirToBackup="/data/vms/W2KFS1" #Specifies to back up W2KFS1 folder
VMXPath="/data/vms/W2KFS1/W2KFS1.vmx"
NfsPath="mediasrv1:/data/Backup" #Specifies NfsPath to back up to
#Name of backup file. Includes date stamp.
BackupDestination="/mnt/backup/"$VMName-"$Today".tar"
#Name and location of backup log file
BackupLog="/mnt/backup/"$VMName"-Backup_Log_"$Today".log"
#==== Shutdown VM ====
vmware-cmd $VMXPath stop

#==== Mount backup folder ====
mount -t nfs $NfsPath /mnt/backup

# ==== Run the backup=====
tar -cpvf $BackupDestination $DirToBackup >&$BackupLog
```

```
#==== Unmount backup folder ====
umount /mnt/backup
```

```
#==== Startup VM ====
vmware-cmd $VMXPath start
```

Now that you've tackled flat-file backups of the most popular VMware products, next you'll explore scripted flat-file backups for the Microsoft VM applications.

Running Virtual PC 2004 Flat-File Backups

Although Virtual PC supports the command-line startup of VMs, it doesn't support command-line VM shutdowns. Because of this, performing flat-file backups of Virtual PC VMs requires a little creativity. Since Virtual PC officially supports only Microsoft OS VMs, in this section we'll cover a batch script that will shut down a Windows VM, perform a backup, and then restart the VM.

In the batch script we put together to back up Virtual PC VMs, we use `shutdown.exe` to remotely shut down the VM and then use `taskkill.exe` to close the Virtual PC application. This is needed because Virtual PC will hold a lock on its VM's files while the application is still running (regardless of the state of its VMs). After shutting down the VM and closing Virtual PC, the script then uses the `ntbackup.exe` command-line tool to back up the VM. When the backup completes, the `Virtual PC.exe` command-line tool restarts the VM. A few variables are used in the script:

- **VMName:** Host name of VM to be backed up
- **VMFolder:** Location of VM configuration and virtual disk files
- **Today:** Current date formatted as mm-dd-yyyy (used to date stamp backup files)
- **BackupFolder:** Target location for backup
- **BackupFile:** Name of backup file (includes path loaded into BackupFolder variable)
- **JobName:** Friendly name of backup job, referenced by NTBackup.exe
- **Description:** Describes backup, referenced by NTBackup.exe

Now let's look at the actual script (see Listing 7-8).

Listing 7-8. *Backing Up a VM on a Virtual PC Host*

```
:: Filename: VPCWinBackup.bat
::
:: Purpose: Automates the backup of a Virtual PC 2004 virtual machine by shutting
:: down the VM, closing the Virtual PC application, using ntbackup.exe to back up
:: the VM's files, and then automatically starting back up the VM. This script
:: backs up a VM named W2KWeb, which is located in the E:\VMs\W2KWeb folder.
:: You would need to alter the VM name and paths for this script to run in your
:: environment.
::=====
```

```

@echo off
:: Set Variables
set VMName=W2KWeb
set VMFolder= E:\VMs\W2KWeb
set Today=%Date:~4,2%-~7,2%-~10,4%
set BackupFolder=\\Dempsey\Backups\%VMName%
set BackupFile=%BackupFolder%\%Today%.bkf
set JobName=%Today% Full Backup
set Description=Starting %Today%

:: Stop VM Using shutdown.exe (requires UNC path to VM)
:: On W2K and earlier hosts Shutdown.exe requires installation of resource kit
shutdown.exe /m \\%VMName% /s /t 0

::Wait for VM to shut down (uses ping -n <seconds> to the loopback address
:: to create a delay) 180 seconds = 3 minute delay.
ping -n 180 127.0.0.1 > nul

:: Close Virtual PC Application and any child processes
taskkill /F /IM "Virtual PC.exe" /T

:: Back up VM
IF NOT EXIST %BackupFolder% md %BackupFolder%
ntbackup backup "%VMFolder%" /J "%JobName%" /D "%Description%" /F "%BackupFile%"

:: Start VM
"Virtual PC.exe" -singlepc -pc %VMName% -launch

```

Remember that tools such as shutdown and taskkill require administrative rights to execute, so this batch script should run under an administrator account. You can easily accomplish this by configuring the script to run periodically using the Scheduled Tasks system tool and configuring the option to run the script under a specific user account.

Running Virtual Server 2005 Flat-File Backups

Flat-file backups with Virtual Server 2005 are slightly more challenging compared to the VMware GSX product line. This is because Virtual Server 2005 doesn't gracefully shut down Linux VMs when powered down. However, it supports a graceful shutdown of Windows VMs. Even without graceful shutdown of Linux VMs, Virtual Server supports automatically saving the state of any running VM at the time the Virtual Server service stops. Furthermore, this product supports automatically resuming a saved VM when the service restarts.

If the bulb is starting to flicker a little bit, it's for a good reason. The built-in automatic save and restart capabilities of Virtual Server make it easy to use a batch script to back up all VMs on a single host system. If your preference is to back up each VM independently, then the methodology for backing up the VM will change slightly. Virtual Server 2005 doesn't have native support for command shell administration; instead, it supports using COM for scripted administration. What this essentially means is that backing up a single VM will require a script written in a scripting language such as Visual Basic Script (VBScript) or Perl. To cover both the "single VM" and "all VMs" backup scenarios, the next two sections will show you scripts that will do each task. To back up all VMs, we'll show you how to use a simple batch script, and to back up a single VM, we'll show you some VBScript code that will get the job done.

Backing Up All Virtual Server VMs

The key to backing up all VMs on a Virtual Server host system using a single script is to prepare the VMs to automatically save their state if the Virtual Server service stops. Doing so will allow you to automatically suspend all running VMs when Virtual Server stops, which will place each VM's related files in a state so that it can be successfully copied. To prepare the VMs for backup, you need to set their default behavior for when Virtual Server starts or stops. To do this, follow these steps:

1. On the Virtual Server host system, click Start ► All Programs ► Microsoft Virtual Server ► Virtual Server Administration Website. If any VMs are running, shut them down before proceeding.
2. When the Web site opens, mouse over the Configure item under the Virtual Machines menu, and select the first virtual machine listed.
3. You should now see the status of the VM you selected. Scroll down the Web browser window until you see the Configuration section of the page. In this section, click the General Properties link.
4. Now check the Run the Virtual Machine Under the Following User Account box. Enter a username and password of an account with administrative rights under which to start the virtual machine.
5. Now in the Action When Virtual Server Starts menu, select Always Automatically Turn on Virtual Machine. In the Action When Virtual Server Stops menu, select Save State. Figure 7-11 shows the settings described in steps 4–5.
6. Scroll to the bottom of the General Properties page, and click the OK button.
7. Repeat steps 2–6 for each additional VM on the server.

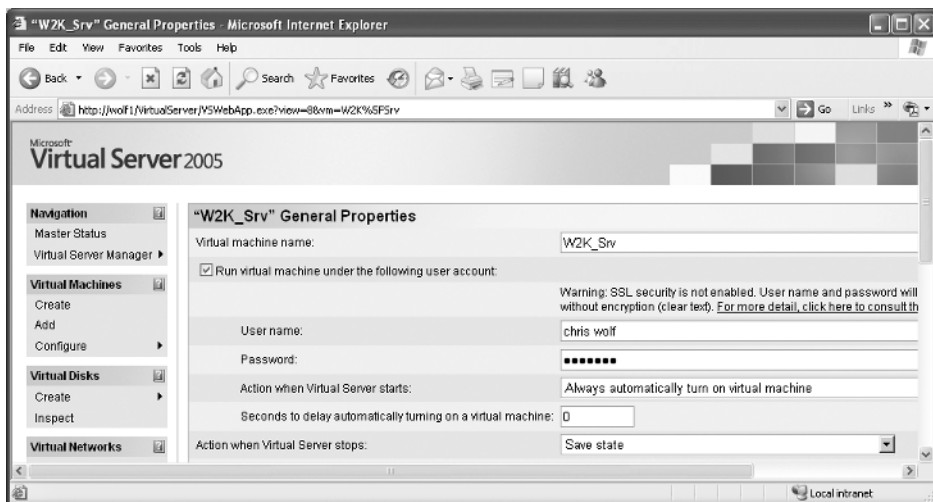


Figure 7-11. Setting Virtual Server VM start and stop actions

Once you've set the start and stop actions for each VM, you're ready to run a script to back them up. The script used in this section to back up all VMs on a Virtual Server host uses the following variables:

- **VMRootFolder:** Location of parent drive or folder containing all VMs
- **Today:** Current date formatted as mm-dd-yyyy (used to date stamp backup files)
- **BackupFolder:** Backup data destination folder (create this folder prior to running backup), which can be a local or UNC path
- **BackupFile:** Filename of backup job (stamped with current date)
- **JobName:** Friendly name for job (stored in backup file)
- **Description:** Description of job (stored in backup file)

With an understanding of the variables used, refer to the script (see Listing 7-9).

Listing 7-9. *Backing Up All VMs on a Virtual Server Host*

```
:: Filename: VS2005AllVMBBackup.bat
::
:: Purpose: Automates the backup of all VMs on a VS 2005 host. VMs
:: must be configured to Save State when the VS service stops and to automatically
:: turn on when the VS service starts. After the service stops, ntbackup.exe is
:: used to back up the root folder that contains all of the hosts VMs.
:: When the backup completes, the Virtual Server service is restarted, which in
:: turn resumes all saved VMs.
::=====
@echo off
:: Set Variables
set VMRootFolder= E:\VMs
set Today=%Date:~4,2%-%Date:~7,2%-%Date:~10,4%
set BackupFolder=\\Zeus\Backups\VMs
set BackupFile=%BackupFolder%\%Today%.bkf
set JobName=%Today% Full Backup
set Description=Starting %Today%

:: Stop Virtual Server Service - Saves state of all VMs
net stop "Virtual Server"

:: Back up VMs
ntbackup backup "%VMRootFolder%" /J "%JobName%" /D "%Description%" /F "%BackupFile%"

:: Start Virtual Server Service - Resumes all saved VMs
net start "Virtual Server"
```

As you can see, this is one of the shortest scripts in the entire chapter, yet it's one of the most powerful. By taking advantage of the built-in save state and VM startup features of Virtual Server, you're able to back up all VMs on the server with just a few commands. If you're just interested in backing up one VM, you'll see how to do that next.

Backing Up a Single Virtual Server VM

To back up a single VM running on Virtual Server 2005, you must first save the VM's state. This will quiesce the VM while the backup runs. Once the backup completes, the VM can be restarted. Suspending the VM prior to backup will place the VM's configuration and virtual disk files in a fixed state so that they then can be reliably copied.

Unfortunately, Virtual Server 2005 doesn't allow you to directly administer VMs from the command line, so for this specific task you'll use VBScript. This script uses the following variables, which you'll need to modify for it to successfully back up one of your VMs:

- **strVMName:** Identifies the name assigned to virtual machine to back up (visible in Virtual Server's Web UI)
- **strVMFolder:** Provides location of VM's files
- **strToday:** Defines current date formatted as mm-dd-yyyy
- **strBackupFolder:** Identifies location where backup data is stored
- **strBackupFile:** Stores name and full path to backup (.bkf) file
- **strJobName:** Defines friendly name for job (stored in backup file)
- **strDescription:** Holds description of job (stored in backup file)
- **strNTBackupCommand:** Stores complete backup command syntax that's passed to the `ObjShell` command shell object
- **objVirtualServer:** Provides connection to Virtual Server application
- **objVirtualMachine:** Provides connection to specific virtual machine
- **objShell:** Runs `ntbackup.exe` shell command from within script
- **errBackup:** Holds return code for `ntbackup.exe` command

With the variables out of the way, let's look at the script (see Listing 7-10).

Note The methods used in this script are supported only on Windows Server 2003 or higher host systems.

Listing 7-10. *Backing Up a Single VM on a Virtual Server Host*

```

' Filename: VS2005VMBBackup.vbs
'
' Purpose: Automates the backup of a single VMs on a host system
'          by first suspending the VM, then running the backup, and finally
'          resuming the VM.
' =====
Option Explicit
'Declare variables
Dim strVMName, strVMFolder, strToday, strBackupFolder, strBackupFile
Dim strJobName, strDescription, strNTBackupCommand
Dim objVirtualServer, objVirtualMachine, objShell, errBackup

'Load current date (formatted as mm-dd-yyyy) into variable strToday
strToday = Month(Date) & "-" & Day(Date) & "-" & Year(Date)

'Define Script Variables
strVMName = "W2K_Srv"
strVMFolder = """"E:\VMs\W2K_Srv""""
strBackupFolder = "G:\Backup\VMs"
strBackupFile = Chr(34) & strBackupFolder & "\" & strVMName & "_" & strToday _
                & ".bkf" & Chr(34) 'Chr(34) is used to include a quote in the string
strJobName = Chr(34) & strToday & " Full Backup" & Chr(34)
strDescription = Chr(34) & "Starting " & strToday & Chr(34)
strNTBackupCommand = "ntbackup backup " & strVMFolder & " /J " & strJobName & _
                    " /D " & strDescription & " /F " & strBackupFile

'Instantiate Virtual Server Object
Set objVirtualServer = CreateObject("VirtualServer.Application")

'Instantiate Virtual Machine Object
Set objVirtualMachine = objVirtualServer.FindVirtualMachine(strVMName)

'Save VM State
objVirtualMachine.Save()

'Instantiate command shell object
Set objShell = WScript.CreateObject("Wscript.shell")

'Use ntbackup.exe to back up saved VM
errBackup = objShell.Run(strNTBackupCommand,1,True)

'Restart VM
objVirtualMachine.Startup()

```


Since the script saves the state of a running VM before backing it up, and since the save state feature runs transparently to the guest OS, this script can back up any Virtual Server VM, regardless of its installed OS.

At this point, we've covered all the primary flat-file backup scenarios for both Windows and Linux VMs. Next, we'll cover a method for performing backups in real-time: online snapshots.

Taking Online Snapshots

Online snapshots are a feature supported in the VMware line of virtualization applications. With online snapshots, you can create a point-in-time copy of a running virtual machine. With an online snapshot saved, you can revert to the snapshot later. This gives you the ability to create an image of a VM, make any changes you want, and with the click of a button roll the VM back to the time of the last snapshot.

A disadvantage of this feature is that it's GUI-driven. If you want to do a snapshot, you need to do it from the VMware management UI. Because of this, you shouldn't look at snapshots as a way to replace the other backup methodologies already mentioned in this chapter but rather as a feature that can aid in testing and training. In our opinion, whenever backups are left up to humans to manually perform (instead of being automated), it's just a matter of time before a backup is skipped.

Although not necessarily a good choice strictly for backups, here are some reasons to use the snapshot feature:

- **Testing:** Create a baseline VM, and then create a snapshot. After testing is complete, revert the VM to the previously saved snapshot.
- **Demonstrations:** Take a snapshot before a software demonstration. Once the demonstration completes, you can power down the VM and revert to the saved snapshot.
- **Training:** You can walk students through a task and afterward revert to the beginning so the students can practice.

Because of snapshots' limited scope, VMware supports only a single saved snapshot per VM, so you'll need to decide carefully when you plan to perform a snapshot.

Taking Snapshots

Taking a snapshot of a VM is a relatively simple process. Here are the steps to follow:

1. In the VMware management UI, boot up the VM (if it isn't currently started).
2. With the VM running, click the Snapshot menu, and select Save Snapshot.
3. Wait for the snapshot to complete.

The snapshot information will be saved in the same location as the VM's virtual disk files. The snapshot files will have `_REDO` as part of their extension, making them easy to identify.

Recovering a Snapshot

Snapshot recovery is just about as easy as taking the initial snapshot. To recover a snapshot, perform these steps while the VM is running:

1. Click the Snapshot menu, and select Revert to Snapshot.
2. If prompted to confirm that you want to revert to the saved snapshot, click Yes.

In a few moments, the system will be restored to the time of the last snapshot.

Although there's no flexibility in the number of snapshots you can take (one), you can set a few options in regard to how snapshots behave. Let's quickly look at them.

Setting Snapshot Options

VMware allows you to set the following options for snapshot behavior:

- Disable all snapshots (not available if a current snapshot is saved).
- Lock the current snapshot, preventing any future snapshots accidentally taken from overwriting it.
- Select how powering down the VM affects the snapshot. VMware will allow you to select from the following power-down behavior options:
 - **Just Power Off:** Do nothing to last snapshot (default setting).
 - **Revert to the Snapshot:** Roll back VM's hard disks to their state as of the last snapshot.
 - **Update the Snapshot:** Apply the current VM saved data to the snapshot (rolls snapshot forward to current point in time).
 - **Ask Me:** Causes a dialog box to be displayed when the VM is powered down, allowing user to decide how to handle the snapshot.

To configure these options, follow these steps:

1. With the VM open in the VMware UI, click the VM menu, and select Settings.
2. In the Virtual Machines Settings dialog box, click the Options tab.
3. Now click the Snapshot setting (see Figure 7-12). Make any desired changes to the snapshot, and then click OK.

Tip If you want to exclude a particular virtual disk from a snapshot, configure the disk as an independent disk. You can access this setting through the virtual disk's advanced properties in the VMware UI.

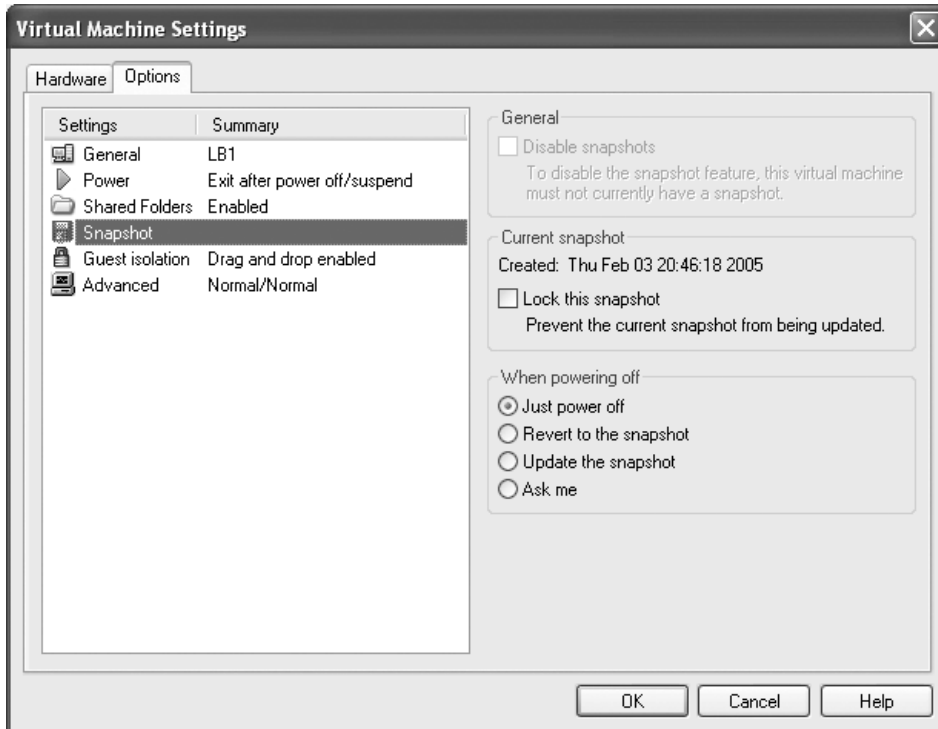


Figure 7-12. *Configuring VM snapshot options*

Again, while snapshots are useful for training, testing, and demonstration purposes, they shouldn't be looked at as a substitute for VM backups. Now that you've examined all methods for securing VM data, next we'll cover the general procedures for recovering VMs.

Performing a Full System Recovery

The entire point of backing up data is to protect your organization's data from loss stemming from corruption, system or disk failure, or a natural disaster. Although understanding how to reliably back up virtual machine data is critical, you shouldn't overlook the process for restoring the data.

To prepare for recovery, a best practice is to ask yourself, can I rebuild my production systems from scratch and have them configured exactly the same as they are now? Many administrators answer "no" to this question, so if that's your answer, you're not alone. To fully rebuild your systems following a disaster, you need to know exactly how the systems were configured. This configuration information includes the following:

- Host name
- TCP/IP configuration
- Disk drive configuration, disk configuration, and partitions (including SCSI IDs)
- Installed applications
- Installed hardware

With this information in hand, it's much easier to successfully recover lost data or systems. The method you choose to back up your VMs will ultimately dictate the techniques you'll use for recovery. In the next two sections, we'll present the methodologies for recovering VMs backed up locally while online, as well as recovering VMs that were backed up directly from the host via flat-file backups.

Restoring Online VM Backups

The process for restoring data from an online VM backup is nearly identical to the data recovery process for physical systems. If you needed to restore a few files or folders, you'd use either your enterprise backup product or a utility such as `dump` or Windows Backup to recover the data. The processes involved in recovering files using `dump` and Windows Backup were outlined earlier in this chapter.

When you're faced with completely restoring a VM from an online backup, the recovery process is slightly more difficult. One of the major advantages of virtualization applications is that they emulate nearly all VM system hardware (see Chapter 1). This means that if a VM's host system crashes, you can recover the VM's backup data to a new VM running on a different host system. With emulated hardware, it's likely that the VM's OS won't even notice the change.

Caution VMware has long noted that its hardware emulation doesn't support moving a Linux VM from a host with an Intel CPU to a host with an AMD CPU. Doing so will cause an irrecoverable kernel panic at startup. This problem doesn't occur when moving VMs with other installed operating systems between hosts with different CPUs.

Let's further discuss the scenario of a VM's host crashing. This is equivalent to one of your servers or workstations crashing. When this happens, the general recovery process involves the following steps:

1. Reconfigure the system's hardware as close to the failed system's configuration as possible. This will include installed devices and hard disks. The number of disks as well as the size of each disk should be equal to or larger than the number of disks on the original system.
2. Power on the rebuilt system (VM), and reinstall the operating system. During installation, use the same host name and TCP/IP settings assigned to the original system.
3. Repartition and format the system's hard disks to their original specifications.
4. If backup agent software was installed previously, reinstall the backup agents.
5. Use your local or enterprise backup software to perform a complete full system restore from the VM's backup data. Note that all the examples used in this chapter involved exclusively performing full backups. Since this isn't practical in many environments, your recovery may involve restoring a full backup and several incremental (or a differential) backups following the recovery of the full backup.
6. When the recovery finishes, restart the system.

Although we used the term VM throughout the previous steps, keep in mind that the same steps apply to host system recovery. You'd start by reinstalling any necessary hardware, and then you'd reinstall the OS, re-create the host system's disk partitions, and then perform the full system restore. Following the full system restore of the host, you'd finally need to restore the data for each VM running on the host.

Restoring Flat-File VM Backups

Flat-file VM backups are much easier to restore than online backups, since the flat-file backup includes the VM's configuration files as well as its virtual disk files. Since virtual disk files are always dynamically changing, you should have to restore just the most recent flat-file backup of a VM in order to fully recover it. Following the restore of the VM files, you then power on the VM and validate its operation. You may notice that some devices, such as the CD-ROM or floppy drive, may not be present. This is likely because of the VM's devices being mapped to static drive letters on the host system that no longer exists. For example, if a VM was configured to use the D drive on the host as the CD-ROM drive and, when the host is recovered, the CD-ROM is assigned the letter G, the VM may not recognize the CD-ROM drive. To correct this, just update the VM's settings.

Summary

As you can see, with sound reliable backups, VM recovery is truly the easiest part of the data protection process. When protecting VMs, one practice to avoid is to never test your backups. Microsoft once stated that a backup administrator is only "...two failed recoveries away from a pink slip." When data is needed and you can't recover it, you may be the one facing the ultimate responsibility for the data loss. (Blaming the hardware vendor or power company doesn't always work!)

To aid in recovery, many organizations stage recovery servers that already have a base OS installed. You can apply the same philosophy to VMs. Since VMs exist as merely collections of files, it's easy to save a virtual machine base OS installation to a DVD. With this approach, you'd simply need to copy the base VM back onto a host server and then rename the VM to that of the original. This can save you significant time when you need to recover data while under the gun. Another approach with staged-recovery VMs is to create a base VM and run the Windows sysprep tool on it. After sysprep runs, just shut down the VM and back it up to another location. The next time the VM starts, the Windows mini-setup will run, and you'll then be prompted to provide a name and TCP/IP information for the system.

When taking backup and recovery seriously, it's important to periodically practice restoring your backup data. Just because a backup program told you that a backup was successfully completed doesn't mean the backup data will restore successfully. Many of us have had to learn this lesson the hard way. You shouldn't have to suffer the same fate as those before you!

This chapter marks the end of the general VM management section of the book. In the remaining chapters, you'll be exposed to the many other aspects of virtualization in IT today. If you're still thirsty for information on VMs, then turn to Chapter 11 or Chapter 14 for additional examples and advanced virtual machine configuration topics.



Using Virtual File Systems

In this chapter, we'll focus on how to use virtual file systems. This chapter will introduce you to two of the most widely used virtual file systems, Distributed File System (DFS) and Andrew File System (AFS). We'll outline the considerations for deploying and managing these technologies in Windows and Linux environments. Because virtual file systems rely on other services, we'll also cover other technologies, including Samba and Kerberos.

Introducing DFS

DFS creates a virtual view of shared directories on a network. It works by creating a namespace with references to network shares and tracks file access. When a user requests a particular resource, DFS matches the request with the correct server and directs the client to the resource. To the end user, the process is seamless; folders look as if they're stored on a single hard drive. For administrators, a single virtual filespace means that servers can be swapped in and out of service without impacting applications or the end user, because the virtual filespace remains constant regardless of what's on the backend. As the DFS administrator, you have the ability to decide what shared folders will appear in the DFS namespace, design the directory namespace hierarchy, and define the names of the shared folders appearing in the namespace. Conveniently enough, security is enforced by existing file system and shared folder permissions.

With DFS implemented on your network, you can make files from across a WAN more accessible to users, experience a reduction in duplicative file storage, and experience an increase in file availability with replication. Without DFS, you'll have to browse the resources on each server in a network to find what's needed; if you have many servers, this can be a cumbersome task and a documentation nightmare. Before jumping into the configuration process, we'll introduce some DFS nomenclature and discuss DFS a bit further.

Note Running DFS across a WAN can negatively impact overall performance by adding unexpected loads. Make sure sufficient bandwidth is available to support your implementation; newly implemented DFS filespace become quickly popular with end users!

Before reading the next few paragraphs outlining the structure and terms used with DFS, be sure to associate the DFS terms with something you already know, such as DNS. Knowing the hierarchical structure of DNS can aid in your understanding of DFS. For instance, the term

root in DNS describes the beginning of the namespace, just as *root* in DFS describes the beginning of the file namespace. Moreover, understanding that DNS is a logical structure rather than a physical one will help you grasp the logical structure of DFS.

A DFS *namespace* refers to the virtual view of network resource shares on disparate servers minimally consisting of a root, links, and targets. The namespace begins at the root and maps to additional root targets; root targets are physical DFS host servers. The DFS *root* is the beginning of the DFS namespace and generally refers to the whole namespace. Stand-alone roots begin their UNC paths with the server name, and domain-based roots begin their UNC paths with the domain name.

A root can map to additional root targets, which in turn map to shared folders on a different server. We use the term DFS *path* to describe a UNC path that begins at a DFS root, and *root target* refers to the actual server hosting the DFS namespace. Domain-based DFS roots may have multiple root targets, and stand-alone DFS roots can have just one root target. A stand-alone DFS namespace consists of a single root target having its information stored in the host's registry. A domain-based DFS namespace has its information stored in Active Directory. Domain-based DFS roots may have multiple root targets offering fault tolerance and load sharing; on the other hand, stand-alone DFS namespaces aren't natively fault tolerant unless you use clustering.

The last couple of terms you'll want to be familiar with are *link* and *referral*. The term *link* is used in three ways. A DFS *link* is a component that lies in the DFS path and maps to link targets. A *link target* is the mapped destination of a DFS link; it can be a UNC path or even a shared folder. The term *link referral* describes the list of link targets for a given link. Referrals are the target lists a client receives when accessing a DFS root or a DFS link, and *root referral* describes the root target list for a DFS root. The transaction between the DFS service and the client receiving the referral lists is a transparent process, as is the name resolution process used in DNS. Using `dfsutil.exe /pktinfo`, you can view the contents of the referral cache. Unfortunately, the DFS tools aren't installed by default; therefore, you'll need to install the Windows 2003 Support Tools to use the DFS Support Tools. The Support Tools are located on the Windows 2003 CD-ROM in the `\support\tools` folder. Using the `suptools.msi` setup file, follow the installation wizard. In addition to all the other cool tools, such as the Dependency Walker, Network Connectivity Tester, Port Query, and Process Viewer, after completing the install, you'll also have access to the following:

- The DFS `dfscmd.exe` command-line tool used for scripting DFS tasks related to roots, links, and targets
- The `dfsgui.msc` DFS snap-in that can be installed on Windows XP SP1 machines for remote administration and can be used for configuring namespaces, targets, referrals, and replication
- The `dfsutil.exe` command that's the advanced DFS utility to help you in many DFS administrative tasks, such as determining the namespace size, exporting/importing namespaces, checking the site name/IP address of a computer, adding/removing root targets, and updating site information for root servers

Note Though we're discussing virtual file systems in this chapter, we'll show how to configure AFS and DFS on virtual machines.

Implementing Windows DFS

Before implementing DFS, you need to be aware of the services it requires. Table 8-1 outlines DFS dependencies. In addition, the DFS service requires several services to be running on DFS root servers and domain controllers; for instance, root servers need the Server, Workstation, and Security Accounts Manager services.

Table 8-1. *DFS Dependencies*

Dependency	Purpose
Active Directory replication	Replicates DFS objects to all domain controllers
SMB protocol	Vital for communications with root servers
RPC service/locator	Needed for DFS tools to communicate with servers
DFS service	Required on root and domain controller servers for DFS to function properly

Now that you're aware of the DFS dependencies, you should also be aware of the network protocols and ports required to support DFS; being that you know Active Directory and SMB are required, you have a good idea of the ports you'll need to deal with at the perimeter of your network. They are as follows:

- Lightweight Directory Access Protocol (LDAP) (UDP/TCP 389)
- NetBIOS (UDP 138, TCP 139)
- RPC (TCP 135)
- SMB (UDP/TCP 445)

Implementing DFS touches all aspects of your production environment. To successfully deploy DFS beyond the lab environment, you must completely understand Microsoft inter-networking, including Active Directory and dynamic DNS. Moreover, you must know how to wield the power of your firewall and routing technologies. When deploying DFS in your infrastructure, Microsoft suggests you use this checklist:

- The PDC emulator is working.
- DNS and Active Directory replication are working.
- Firewalls don't block DFS RPC ports.
- The DFS service is running on all domain controllers and root servers.
- Active Directory sites and site costs are configured, and the Bridge All Sites Links option is enabled.
- Client computers aren't running legacy operating systems and are domain members.
- The HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/Lsa/restrictanonymous registry entry isn't set to 2 for Windows 2000 Servers.

We’ve tackled DFS nomenclature, DFS prerequisites, and some gotchas; you now have enough information to be dangerous in the lab, so we’ll show how to perform a generic DFS configuration. Once we get the basics out of the way, we’ll cover root replicas, replica DFS links, load balancing, and fault tolerance.

Installing DFS is easy; Windows 2000 and Windows 2003 install it by default. Before clicking through the basic setup, take a moment to think through the ramifications of end users being able to see the root names and link names that will be used in your DFS environment. To that end, you should be aware of these limitations:

- Roots can’t be nested within each other.
- Every DFS root must have its own shared folder.
- Server clusters shouldn’t share the same DFS root name as nonclustered roots.
- Domain-based DFS roots must have the same UNC share name and DFS root name.
- Domain-based DFS roots and domain controller shares shouldn’t use the same name.

The next thing you need to consider is whether you want to install a domain-based root or stand-alone root; Table 8-2 lists the considerations.

Table 8-2. *Domain-Based and Stand-Alone DFS Root Considerations*

Domain-Based DFS Root	Stand-Alone DFS Root
Requires Active Directory	Doesn’t use Active Directory
Uses File Replication Services (FRS)	Doesn’t support FRS
Can have multiple root-level targets	Limited to one root-level target
Achieves fault tolerance through FRS	Achieves fault tolerance via server clustering

When using domain-based roots, you should limit DFS links to fewer than 5,000; on stand-alone DFS roots, you should limit DFS links to 50,000. The disparity in the sheer number of links can be explained by the overhead of Active Directory generating significantly larger amounts of network traffic. Before deploying DFS, a solid assessment of your WAN should be your first priority.

Note When using Windows 2000 and Windows 2003 to create a DFS environment, you must use a Windows 2003 operating system to administer DFS.

You’ll begin the configuration process by creating a domain-based DFS root. Select Start ► All Programs ► Administrative Tools ► Distributed File System. On the Action menu, select New Root. The New Root Wizard will launch and prompt you through the creation process. As depicted in Figure 8-1, your first decision is to choose a domain or stand-alone root.

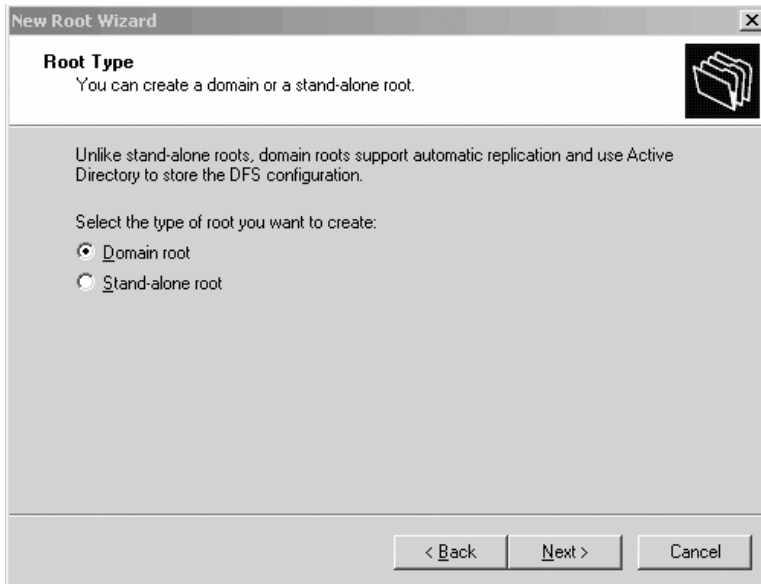


Figure 8-1. DFS root type selection

On the Host Domain selection screen, you'll need to provide the DFS host domain, and on the Host Server selection screen, you'll need to supply the FQDN of the DFS root host. Next, as shown in Figure 8-2, you'll need to supply a unique root name for the DFS root. Being that the Comments field isn't viewable by end users, make sure the root name adequately describes the root to help in end-user navigation.

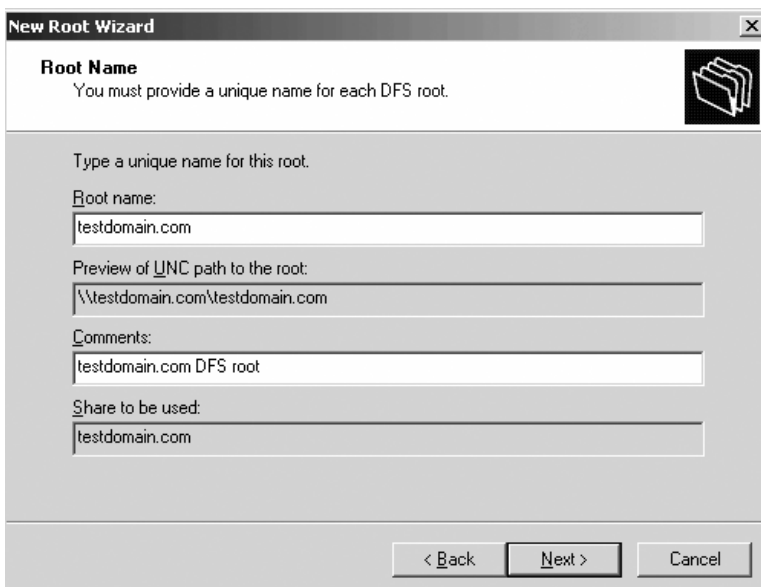


Figure 8-2. DFS root name entry

On the Root Share selection screen, create a directory to be shared; the DFS wizard will take care of automatically setting up the folder and sharing it. Conversely, you can select a predefined share if you want. The wizard will complete the install by summarizing your selections and present them to you, as shown in Figure 8-3. Review your selections, as it will be difficult to “undo” a DFS deployment in the future. The entire creation process ends in an uneventful finale where you’re presented with the DFS interface and the configured file space.



Figure 8-3. *New Root Wizard completion summary*

In the event of server failure, you can make the namespace defined by the domain DFS root available if you created additional root targets (replicas) beforehand; once again, the only way to ensure availability is to minimally have two domain controllers and two root targets. If you don’t have two servers, the namespace will be unavailable when the server flatlines.

You can add root targets by selecting Action ► New Root Target. The New Root Target Wizard will walk you through the process. Removing a root target is just as easy; highlight the target, and select Action ► Remove Root. Yes, it really is that simple to create or remove a replica of your root. Windows 2003 DFS is a significant improvement over Windows 2000.

When creating a DFS link, the directory specified must first be shared. If the share for the directory doesn’t exist, you can’t build a DFS link to it. When creating shares, be sure to use good passwords, share permissions, and access control. To create a link, select Action ► New Link. As shown in Figure 8-4, enter the name for the link and the path, its UCN path, and the referral cache timeout. The referral cache stores the link information on the client based on the number specified in seconds. Setting the cache too high will keep clients from seeing newly created links within a reasonable amount of time. Setting the time too low will create unnecessary network traffic.

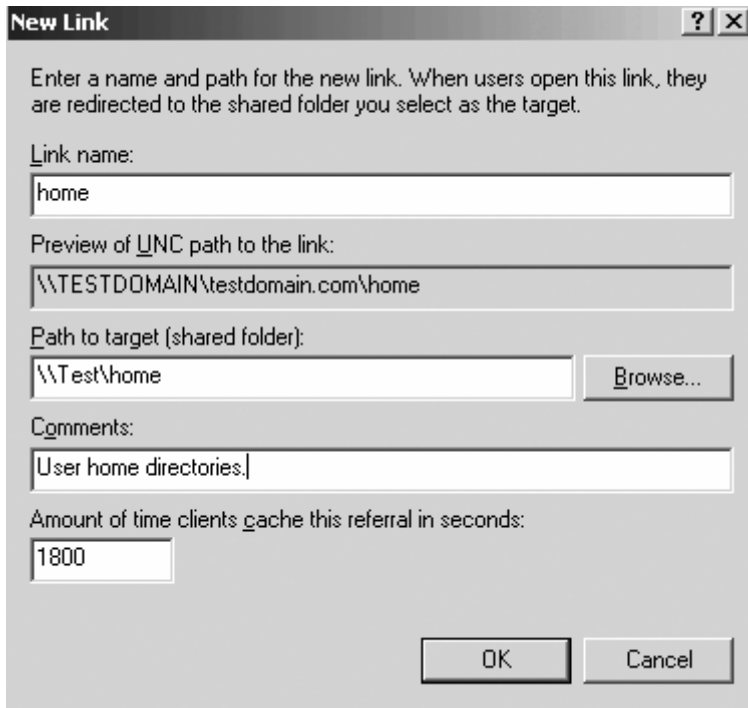


Figure 8-4. *Creating a DFS link*

Connecting clients to a DFS namespace requires Windows 9x operating systems to install the DFS client; later versions of Windows—such as Windows NT, Windows 2000, and Windows XP—natively support connectivity to the DFS namespace. To connect, simply open My Network Places, find the namespace root server, and locate the root target. You can create a shortcut on your desktop to the DFS namespace just as you'd make any other shortcut. Assuming you want to create a shortcut to the root, right-click the root target and select Copy. Next, right-click your desktop, and select Paste Shortcut.

Tip You can make your domain's DFS filespace more accessible by publishing it in Active Directory. To do this, select your namespace root and then select Action ► Properties from the menu. Click the Publish tab, and check the Publish This Root in Active Directory option.

Clearly, there isn't much to configuring DFS and being able to take advantage of a single filespace in a network. However, don't be duped into believing it's good enough to create a file-space without some type of backup or redundant system. Take a moment to configure FRS for your domain-based DFS initiative. When using multiple targets, you'll have to decide on a replication method for synchronizing data between the targets; you can replicate data doing it manually, using third-party tools, or using FRS. It's best to manually replicate static or read-only data. The drawback to manual replication is that changes are updated infrequently.

If you're implementing a stand-alone DFS deployment, you have no other option than to manually replicate data by performing a simple copy or using a tool such as Robocopy. On the other hand, FRS is used for domain-based DFS namespaces; it works by detecting changes made to files or folders and replicating the entire file to the servers in the replica set.

You can set up FRS in a continuous mode (changes are replicated within five seconds of a file being closed) or at scheduled intervals. One drawback to continuous replication is that the circuits in a WAN can be completely utilized with replication data. Moreover, using a scheduled interval to update data can leave replica sets unsynchronized for extended periods of time, which causes multiple versions of a particular file to exist. When it comes time to synchronize, FRS will use a conflict resolution process for changed files by permitting the most recently updated version to be replicated in the replica set; this holds true for directory name changes as well. If two directories are created in a replica set with identical names, FRS will rename the most recently created folder.

You can maintain replication integrity by using the update sequence number (USN) journal record to change files on replica members. When it comes time for you to decide on a strategy for making critical data available, you'll want to read "Choosing an Availability Strategy for Business-Critical Data" in the Windows Server 2003 Deployment Kit, which is available at <http://www.microsoft.com/resources/documentation>.

Note NuView StorageX, documented in this book's appendix, provides a distributed directory service that focuses on performance. StorageX logically aggregates DFS storage and makes policies available for automating administrative tasks, such as replication, client failover, capacity optimization, and reporting.

We've yapped enough about the benefits and drawbacks of FRS, so now we'll cover how to configure it for DFS. First open DFS, and select the domain DFS root or link to be configured for replication. On the Action menu, click Properties and select the Replication tab. To initiate replication, on the Action menu, select Configure Replication and complete the Configure Replication Wizard.

Caution FRS replica sets shouldn't be configured on a volume managed by remote storage because of severe performance impacts and data corruption.

If you want, you can also exclude files and folders by selecting either File Filter or Subfolder Filter on the Replication tab and then selecting Edit. Type the filename of a file or folder name you want to exclude from replication. Next, click Add to append the file or folder name to the exclusion list.

This brings our discussion of DFS to a close; we've decided to concentrate on the Linux integration aspects of DFS in this chapter, primarily because little of it has been documented to date. You can find plenty of documentation for setting up DFS in Windows environments at the Windows 2003 Server home page (<http://www.microsoft.com/windowsserver2003>).

Implementing Linux DFS

Integrating Linux with Microsoft DFS and Active Directory requires you to install and configure Samba on your server. Samba is an open-source implementation of Microsoft's SMB and Common Internet File System (CIFS) protocols. Though SMB is primarily a network file-sharing protocol, it also aids in several other important network tasks, such as network browsing, printing, resource authenticating, and extended file attribute handling. CIFS is a dialect of SMB that was implemented with Windows NT 4. *Dialect* refers to the message packets defining a particular version of SMB. If you break out a network analyzer, such as Network Instruments' Observer, you'll find that SMB lurks around the top two layers of the OSI model and relies on NetBIOS over TCP/IP (NBT) for transport at the network layer. In this section, we'll first cover how to set up Samba to act as a Kerberized domain controller serving DFS shares, and then we'll finish the section by showing how to join the Samba server to an existing domain.

Before installing Samba, check to see if it's installed on your Linux system:

```
rpm -qa | grep samba
```

If you see a list of packages, you may want to upgrade your system to the latest stable version of Samba. The prerequisites for Samba vary, but you can look them up for your Linux distribution at <http://www.samba.org>. For RPM distributions, in addition to having the development environment installed, you'll need the following:

- krb5-workstation
- krb5-libs
- krb5-devel
- pam_krb5
- openldap
- openldap-clients
- openldap-devel

You can verify the installation of RPM prerequisites by querying the RPM database:

```
rpm -qa | grep openldap  
rpm -qa | grep krb
```

Time synchronization is important for all computer networks. Windows and Linux OSs require systems to have accurate time for Kerberos ticketing, NT LAN Manager version 2, and event logging. The default configuration of Kerberos requires hosts to be time synchronized within five minutes of each other. Active Directory uses Kerberos for authentication, where the time is encoded on the client's authentication ticket; using time as a constraint prevents system attackers from reusing old authentication tickets. The Windows W32Time service uses the Simple Network Time Protocol (SNTP) to synchronize system clocks and isn't as accurate as NTP in regard to adjusting the clock. Registry settings for the Windows W32Time service are in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Parameters subkey. To synchronize your Windows 2003 authoritative time source with an external entity, you'll need to use the `w32tm` command to add the time source:

```
w32tm /config /syncfromflags:manual /manualpeerlist:<NTP_Server_IP_or_FQDN>
```

Now update the server:

```
w32tm /config /update
```

You can synchronize the clocks on Windows 9x and Windows NT systems using the net time command:

```
net time \\<NTP_Server_IP_or_FQDN>> /set /yes.
```

You can time synchronize Linux systems by editing the /etc/ntp.conf file. Find the server statement, and edit it so it has the IP address of your network's NTP server:

```
server xxx.xxx.xxx.xxx
```

You can additionally use the ntpdate command to synchronize the local clock with the Windows W32Time service. To get an idea of the information obtained from an (S)NTP server using the debug option, execute `ntpd -d x.x.x.x`. To set the time of your Linux box on boot, edit /etc/rc.d/rc.local and append `ntpdate` to the file. Additionally, to avoid socket errors, you should disable the NTP daemon.

```
ntpdate -s <x.x.x.x>
```

If you're using Windows XP as your VM host, you can enable SNTP for guest VM synchronization by editing the registry entry `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config`. Set the `AnnounceFlags` DWORD value to 5. Next, enable the SNTP server by editing `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer\`. Set the `Enabled` DWORD value to 1. Restart `w32time` using `net stop w32time & net start w32time`.

If you're using a Linux host, edit /etc/ntp.conf to allow external host synchronization. The first line in the file sets the host as the time authority, and the second line denies access to everyone. The third line allows VMs on a given subnet the ability to synchronize, and the last line is for the local host.

```
server 127.127.1.0
restrict default notrust lowpriotrap nopeer nomodify
restrict 192.168.100.0 mask 255.255.255.0 # local hosts
restrict 127.0.0.1 mask 255.0.0.0 # local config
```

Note For the best overall security and performance of your network, you should use an NTP server and point all hosts to it. To properly build and secure an independent NTP time synchronization server for your network, please refer to <http://www.ntp.org>. In addition, for a complete introduction to Microsoft's implementation of NTP (Windows W32Time service), search Google for *Windows Time Service* or *wintimeserv.doc*.

For this install of the SMB/CIFS protocol, we'll assume you don't have Samba installed, so we'll proceed by showing how we compile our own binaries. If you want to see if you have Samba installed, you can execute `rpm -qa | grep samba`. Download Samba from <http://www.samba.org> and save it to a convenient location, or snag it at the command line using `wget`:

```
wget http://us1.samba.org/samba/ftp/samba-latest.tar.gz
```

After installing Samba, you should familiarize yourself with its three daemons, as listed in Table 8-3.

Table 8-3. *Samba Daemons*

Daemon	Primary Purpose
nmbd	Provides name registration and resolution and aids in network browsing
smbd	Manages file and print services and local authentication
winbindd	Required for membership in a Windows NT 4 or Active Directory domain or in a trust relationship

Now follow these steps:

1. Extract the Samba download:

```
tar zxvf samba-latest.tar.gz
```

2. You don't want to be the unwitting victim of a maliciously altered Samba download, so verify the PGP signature of your gzipped TAR file to confirm its authenticity. You may need to browse the Samba FTP directory for the matching signature source file of your download. We'll show how to use `wget` to download the source file and the Samba public PGP key to the extracted Samba directory, such as `/root`:

```
wget http://us1.samba.org/samba/ftp/samba-3.0.10.tar.asc
```

```
wget http://us1.samba.org/samba/ftp/samba-pubkey.asc
```

3. Import the Samba public key into your system using GNU Privacy Guard (GnuPG). GnuPG is a free replacement for Pretty Good Privacy (PGP). GnuPG doesn't use patented technology and is open-source technology. PGP uses the patented International Data Encryption Algorithm (IDEA) block cipher.

```
gpg --import samba-pubkey.asc
```

4. Verify Samba's integrity by using `gpg --verify <file_signature> <file_name.tar.gz>`. Your system should minimally respond with "Good Signature from Samba Distribution Verification Key."

```
gpg --verify samba-pubkey.asc samba-3.0.10.tar.asc samba-latest.tar.gz
```


5. If the configure script isn't present in the source directory in which Samba was extracted, /samba-3.0.10/source, you'll need to build it using ./autogen.sh. If you downloaded Samba directly from <http://www.samba.org>, the configure program should be present. We'll show how to use it to build the binaries. Being that we want our Linux server to be integrated into a Windows 2000 domain as a client with DFS capabilities, we'll need to specify several additional programs. You may not need every program we list here, but it simplifies the installation and configuration processes later. Moreover, many of the applications in the list are installed by default; therefore, be sure to research your Samba version ahead of time. For instance, Samba 3 includes DFS support. In a production environment, you'll want to install only what's necessary to satisfy your needs. Here's what we're installing:

- acl-support
- ads
- krb5
- ldap
- msdfs
- windbind
- smbwrapper
- smbmount
- utmp

6. To get an idea of all the optional programs Samba makes available to you, execute ./configure --help. To configure your server, execute the following command:

```
./configure --with-acl-support --with-ads➤
--with-krb5=/usr/kerberos --with-ldap➤
--with-msdfs --with-windbind --with-smbwrapper➤
--with-smbmount --with-utmp tee | config.log
```

Notice that screen output is being piped to a log file with the tee command. tee allows configure to write to the screen and to the log file. If configure runs into errors, you can easily skim the captured output to troubleshoot your distribution of Linux. You can find additional information in config.log.

Note Make sure the DNS is properly configured. Samba and Kerberos depend on it to function correctly. Without a properly configured DNS server, you'll run into many problems that could have easily been avoided.

7. If you run into Kerberos errors during the make process, verify that your version of Samba is compatible with the version of Kerberos you're attempting to use. If not, in a unique directory (for example, /krb5), download a compatible version from MIT by using `wget`. Strictly put, you don't necessarily have to use Kerberos with Samba to serve DFS shares; however, you'll have to rely on Samba's password files.

```
wget http://web.mit.edu/kerberos/dist/krb5/1.4/krb5-1.4-signed.tar
```

Tip Your VMs have the ability to revert to a previous state, which will save you a considerable amount of configuration time in the event of a botched installation. Take a moment to create a snapshot, redo log, or simple offline backup before starting major configuration changes—you'll thank yourself.

8. Extract the files from the downloaded TAR:

```
tar xvf krb5-1.4-signed.tar
```

9. Extract the Kerberos tarball. The extraction process will build the directory `krb5-1.4` and create the file `krb5-1.4.tar.gz.asc`.

```
tar zxvf krb5-1.4.tar.gz
```

10. You can check the authenticity of your Kerberos download by using the detached PGP signature from MIT's Web site at <http://web.mit.edu/kerberos/www> and by using GPG.

11. Navigate to the `src` directory:

```
cd /krb5tree/krb5-1.4/src
```

12. Run `configure`, and execute `make`:

```
./configure --enable-dns --enable-dns-for-kdc --enable-dns-for-realm --enable-kdc-replay-cache --enable-krb4 --enable-shared; make
```

13. Test the build by running the built-in regression tests:

```
make check
```

14. Finally, install the binaries:

```
make install DESTDIR=</Directory_Path>
```

After completing the installation of Kerberos, you'll need to configure the master key distribution center (KDC), which is responsible for issuing Kerberos tickets. For this, you'll need to edit two configuration files, `/etc/krb5.conf` and `/usr/local/var/krb5kdc/kdc.conf` (or `/var/kerberos/krb5kdc/kdc.conf` for Red Hat RPM-installed systems). Both files need to have realm information correctly configured for host names and your realm name. You can satisfy

the constraints of basic realm by substituting `EXAMPLE.COM` and `example.com` with your domain name. Be sure to follow the case of the examples in each configuration file. Be sure that the KDC sample entry matches the host name of the master KDC server.

Note For a complete discussion of the configuration of a Kerberos realm, refer to the “Kerberos V5 Installation Guide” at <http://web.mit.edu/kerberos/www/krb5-1.2/krb5-1.2.6/doc/install.html>. The guide also includes instructions on creating a logging stanza for your configuration file and security best practices.

Follow these steps:

1. Use the `kdb5_util` command on the master KDC to create the realms database. The `-s` option creates the stash file for the master key. The utility will be in your installation directory. This example is for Red Hat:

```
/usr/kerberos/sbin/kdb5_util create -s
```

2. You now need to create or edit `kadm5.acl` to tell `kadmin` who has administrative access to the database. The file is in the format `Kerberos principal permissions optional target principal`. For a simple realm, you can use wildcards to grant access, such as `*/admin@TEST.COM *`.

3. Start the `kadmin` server:

```
service kadmin start
```

4. You'll need to add at least one administrative principal with `kadmin.local` to allow remote access:

```
kadmin.local -q addprinc <username>/admin
```

5. You now need to create the `kadmin` keytab using the `kadmin.local` command with `ktadd`. In a nutshell, this process gives administrators the ability to access the database. Be sure the path used with `ktadd` reflects your installation. If you're working on an RPM-based system, such as Red Hat, you may not need to perform this step. The `-k` option saves the extracted keytab to the specified `kadm5.keytab` file, which is also used in the `kdc.conf` file. Type `exit` to escape from `kadmin.local`. Once again, be sure to use the directories that reflect your installation, like so:

```
/usr/sbin/kadmin.local
ktadd -k /usr/local/var/
krb5kdc/kadm5.keytab => kadmin/admin kadmin/changepw
```

6. Start the Kerberos daemons:

```
/sbin/service krb5kdc start
/sbin/service kadmin start
```

7. Create host keys by adding principals with `ktadd`. Creating a principal for the master KDC will allow you to move the master KDC to a slave KDC, so add it first:

```
/usr/sbin/kadmin.local
addprinc -randkey host/<FQDN>
```

8. To complete the configuration, create a keytab for the master KDC so it can decrypt tickets:

```
/usr/sbin/kadmin.local
ktadd host/<FQDN>
```

9. Now that Kerberos is installed and configured, you can add principals for network hosts using `kadmin.local`. We'll use the recommended `randkey` command for the host key:

```
/usr/sbin/kadmin.local
addprinc -randkey host/<FQDN>
```

10. You can add principals for network users similarly by using `kadmin.local`. You'll need to specify the password for the user in the creation process:

```
/usr/sbin/kadmin.local
addprinc <userid>
```

11. You can view the server's list of principals in the database while using `kadmin` by executing `listprincs`. In addition, you can test the server's functionality by using `kinit`, `klist`, and `kdestroy`, as outlined in Table 8-4.

Table 8-4. Kerberos Test Commands

Command	Action Performed
<code>kinit</code>	Obtains a ticket and stores it in the credential cache
<code>klist</code>	Views the credential cache
<code>kdestroy</code>	Destroys the cache

12. To configure an XP user and host machines to use a Kerberos 5 server, you'll need to install the tools from the Support folder located on the installation CD-ROM to gain access to the `ksetup` command. Next select Start ► Programs ► Windows Support Tools ► Command Prompt. The first statement configures computer account information, and the second statement specifies the change password protocol. If your Kerberos KDC doesn't allow password changes, users won't be able to change their passwords using the three-finger salute.

```
ksetup /addkdc <DOMAIN_NAME> <FQDN_of_KDC>
ksetup /addpasswd <DOMAIN_NAME> <FQDN_of_KDC>
```

13. The user's local account must be mapped to use the Kerberos server as well:

```
ksetup /mapuser <username>@<DOMAIN_NAME> <username>
```

14. Configuring Linux systems and users to employ a Kerberos realm minimally requires a Kerberos client and a valid `krb5.conf` configuration file. Typically, you can use the same `krb5.conf` file as that of the server. For the Kerberos client, Linux systems using RPMs need `krb5-libs` and `krb5-workstation` packages. Create the host account on the KDC:

```
/usr/sbin/kadmin.local
addprinc -randkey host/<FQDN>
```

15. Extract the keys on the client by first using `kadmin` and then using `ktadd`:

```
/usr/sbin/kadmin.local
ktadd -k /etc/krb5.keytab host/<FQDN>
```

16. Create the user account:

```
/usr/sbin/kadmin.local
addprinc <userid>
```

Note If you want XP services to use Kerberos 5 servers for authentication, refer to “Kerberos Interoperability” in the Windows XP Professional Resource Kit or online at http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prdp_log_tjil.asp. Additionally, if you need more help with client configuration, use Google to find the “Kerberos System Administrator’s Guide” online.

17. Depending on the directory you use to extract the files, you’ll need to specify the root directory of your Kerberos install for your Linux distribution to avoid errors with configure in Samba, such as `--with-krb5=/krb5-1.04`.

18. Create the Samba binaries with the `make` command:

```
make | tee make.log
```

19. Run `make install` to install the compiled binaries.

Now that Samba is installed, you’ll need to configure the service to start automatically. You can do this in many ways, so we’re merely offering one way. To start Samba as a daemon, edit `rc.local` and add the command syntax to start Samba for your Linux distribution:

```
vi /etc/rc.d/rc.local
service smb start
```

Note For the comprehensive resource regarding Samba, refer to the “Official Samba-3 HOWTO and Reference Guide.” It’s freely available in PDF from <http://www.samba.org/samba/docs/Samba-HOWTO-Collection.pdf>.

You must configure Samba prior to using it. Though this just entails creating and editing a text file, configuring the service tends to be the tricky part for most Samba newbies (and veterans!). Generally, most problems are a result of typos and not placing the file in the correct directory. In general, you can place the file in `/usr/local/samba/lib`. Being that this directory varies based on your particular installation, you should check your documentation. Conversely, you can run `testparm` to find the expected installation point. On some Red Hat systems, you may find that the file already exists in the `/etc/samba` directory. A simple configuration consists of only a handful of parameters. Rather than giving a complete discourse on the Samba configuration file, which is well beyond the scope of this book, we'll explain only the relevant parts.

In the first part, the configuration file needs to minimally consist of two stanzas: `global` and `dir_share`. The basic setting instructs Samba to become a member of a workgroup, to share the directory `/data/share`, and to give it a label of `dir_share`.

```
[global]
workgroup = DOMAIN_NAME
netbios name = HOST_NAME
security = share
encrypt passwords = yes
[dir_share]
comment = Shared Directory
path = /data/share
read only = no
oplocks = no
guest only = yes
```

Create the shared directory, and configure permissions to suit your needs. To speed up the troubleshooting process, set permissions to give everyone complete access and control to the directory. You can always tighten the security after you verify the functionality of the server.

```
mkdir /data/share
chmod 777 /data/share_name
```

For testing purposes, create a Samba user account for root with legacy authentication using `smbpasswd`. The first user added initializes the encrypted password file. Test your Samba configuration, and start the service. If you get errors, check your configuration file. If you continue to get errors, revert your VM and start again.

```
smbpasswd -a root
testparm
service smb start
```

Now that the service is running, you can test the server's ability to offer its share to the network. The following command will enumerate any shares, the share type, and associated comments:

```
smbclient -U root -L localhost
```

Take a moment to further test your VM-hosted Samba server by connecting to it from a Windows VM. At the CLI, try to map a drive to the shared directory:

```
net use * \\<samba_FQDN>.x.x.x\dir_share /user:domain_name\root
```

If your host fails to connect to the Samba server, try substituting the IP for the host name to eliminate issues with DNS. Verify that the directory share is created and the permissions are set. Finally, make sure you're connecting to the share name and not the shared directory.

You can manage Samba from the command line or through a GUI. The traditional interface is the Samba Web Administration Tool (SWAT). SWAT offers robust help and allows you to easily configure the `smb.conf` file. SWAT is a daemon that runs under `inetd` or `xinetd`. For RPM installations, simply install your OS's SWAT RPM and move onto configuring Samba. If you rolled your own binaries, you'll need to edit the services file and add `swat 904/tcp`. You'll also need to configure `inetd.conf` or `xinetd.conf` based on the type of superdaemon your Linux distribution uses. For `inetd.conf`, add the following statement:

```
swat    stream    tcp    nowait    root    /usr/samba/swat    swat
```

For `xinetd`-based systems, such as Red Hat, create a file titled `swat` and place it in `/etc/xinetd.d`. The contents of the file need to contain the following:

```
# description: SWAT is the Samba Web Admin Tool. Use swat \
# to configure your Samba server. To use SWAT, \
# connect to port 901 with your favorite Web browser.
service swat
{
    port = 901
    socket_type = stream
    wait = no
    only_from = localhost
    user = root
    server = /usr/samba/swat
    log_on_failure += USERID
    disable = no
}
```

To use SWAT, you'll need to either reboot your VM or restart the superdaemons. Point your Web browser to `http://<FQDN>:901/`. If you can't use Samba with your host's FQDN, fix DNS before proceeding. Implementing SWAT will rewrite your existing `smb.conf` file, so back it up first; for the sake of easing Samba administration, stick to either the CLI or the GUI.

Tip Whether you have an affinity for GUIs or the CLI, you'll want to download and install Webmin for your Linux system from <http://www.webmin.com>. Webmin provides a powerful modular-based Web interface for system administration, including BIND, Samba, Apache, NFS, Clustering, Backup, and Virtualmin—just to name a few.

With the basic install of Samba out of the way, we'll show how to configure Samba as a domain controller (DC). Samba is Samba—that is to say, it's not a Windows NT 4 DC, and it's not a Windows Active Directory DC. Samba performs functionally the same as an NT 4 DC and can integrate with a Windows Active Directory domain. Domains offer security and facilitate

using shared network resources by using single sign-on techniques. You set up security for Microsoft and Samba domains through a security ID (SID). SIDs are generated when an object, such as a user account, computer account, or group, is added to a domain. You can further enhance security by using the SID in conjunction with access control lists (ACLs).

Note At the time of this writing, you can join Samba to an Active Directory domain. However, Samba doesn't support Active Directory. Also, for more detailed information about SIDs and relative IDs (RIDs), refer to http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/techref/en-us/Default.asp?url=/Resources/Documentation/windowsserv/2003/all/techref/en-us/w2k3tr_sids_how.asp.

Configuring Samba as a Windows NT primary domain controller (PDC) simply requires adding domain supporting stanzas to the `smb.conf` file and creating a few directory shares. Being that Samba doesn't have the ability to communicate with a Microsoft backup domain controller (BDC) to synchronize domain-based data, you should create only Samba BDCs. Let's look at an example PDC `smb.conf` file:

```
[global]
workgroup = DOMAIN_NAME
netbios name = HOST_NAME
security = share
encrypt passwords = yes
domain master = yes
local master = yes
preferred master = yes
passdb backend = tdbsam
os level = 65
preferred master = yes
security = user
domain logons = yes
logon path = \\%L\profiles\%u
logon drive = H:
logon script = logon.bat
logon home = \\%L%\u\winprofile\%m
domain admin group = root administrator
time server = yes
```

The global section configures Samba as a domain controller and as a master browser; it also handles logons. Additionally, the stanza directs the logon path for roaming profiles and logon scripts, and it sets the home directory. The entries are all fairly self-explanatory; we do, however, want to discuss a couple of lines in particular: `logon path`, `logon home`, `logon script`, and `domain master`. Table 8-5 lists several variables that can be easily used in Samba's configuration file to aid in its administration of users.

Table 8-5. *Samba Configuration File Variables*

Variable	Function Performed
%I	Client's IP address
%m	Client's NetBIOS name
%M	Client's DNS name
%u	Username
%H	Home directory of %u
%g	Primary group of %u
%S	Current share's name
%P	Current share's root directory
%h	Server's DNS host name
%L	Server's NetBIOS name

The `logon path` statement is in UNC format and is required if you want to implement Windows 2000 and Windows XP roaming profiles. The variables in the statement specify the server name (%L) and the username (%u). The `logon home` statement is also in UNC format; it specifies the server name (%L) and specifies the username (%u) for the user's home directory; %m defines the NetBIOS name. The `logon script` statement directs clients to use a login script file. As a word of caution, the script needs to be in MS-DOS format. If you use Vi to create the script, run the colon command `:set ff=dos` to set the format of the file before inserting script commands. Alternatively, you could create the file on a Windows system and then copy it to the Samba share. If a path is specified for the file, it's relative to the path specified in the `netlogon` stanza. In the example, we're using a batch file. The `domain master = yes` statement sets the server as the PDC. If the value is set to no, it's a BDC.

The `netlogon` section is required by Windows clients during the logon process; without it, logon attempts will fail. This section also directs the location of logon scripts and policy files.

```
[netlogon]
path = /netlogon
read only = yes
write list = ntadmin
```

The `profiles` section defines the share required by Windows 2000/Windows XP roaming profiles. Being that profiles need to be updated by the user account, notice that `read only` is set to no. This is opposite of the `yes` setting in `netlogon`.

```
[profiles]
path = /profiles
read only = no
create mask = 0600
directory mask = 0700
```

The `homes` section generates a share for a user to connect. The share correlates to the user's home directory in `/etc/passwd`. This process doesn't create the home directory but merely facilitates network connectivity and browsing.

```
[homes]
read only = no
browseable = no
guest ok = no
map archive = yes
```

The `dir_share` section is a generic data share for users to exchange files; it can be labeled as anything.

```
[dir_share]
comment = Shared Directory
path = /data/share
read only = no
oplocks = no
guest ok = no
```

The Samba configuration file in this section doesn't necessarily employ best-practice standards for security, and you should follow up by implementing access control options to protect data and verify directory share permissions. If you're unfamiliar with how to control access to Samba shares, set up roaming profiles, or troubleshoot the problems you'll run into, refer to the "Official Samba-3 HOWTO and Reference Guide" online. Whenever you make an edit to the Samba configuration file, be sure to restart its service.

Adding client VMs to the Samba domain requires that the VMs be registered with the Linux host and the Samba service; this procedure is analogous to creating computer accounts in a Windows domain. The registration process is similar to creating a typical user account, save one caveat. Computer account names need to have the same name of the VM and end with `$`. Follow these steps:

1. Create the Linux computer system account with the two parts Samba requires: the username and user ID. In the following code, the `-d` option designates that `home_dir` be set to `/dev/null`, meaning that a home directory isn't created. The `-g` option sets the group the account will belong to; the default is 1. You can use the group ID to add the identification of VMs from physical systems but using different numbers. The `-s` option sets the account's login shell to `/bin/false`, which prevents logins. The `-M` option prevents the system from creating a home directory for the computer account. Notice that `$` is appended to the name.

```
useradd -d /dev/null -g 100 -s /bin/false -M <Computer_Name>$
```

2. Create the Samba service account. You don't need to do anything special for this step. The `useradd` command did all the hard work. Notice that the computer name ends with `$`. Once you've created the computer account, you can join the VM to the domain using the same methods you typically use for Windows-based operating systems.

```
smbpasswd <Computer_Name>$:<Computer_Name>$
```

To provide some redundancy for a Samba PDC, you'll need to create a BDC. A Samba PDC/BDC-configured network is more akin to an NT-type network. Communications between the domain controllers is best performed by utilizing LDAP on the backend to synchronize password databases. Configure your PDC as the master LDAP server and each BDC as slave LDAP servers.

Creating a Samba BDC requires a bit of preparation:

1. The domain SID must match on the PDC and the BDC. Execute `net rpc getsid` to retrieve and store the SID in `secrets.tdb`.
2. The LDAP administration password needs to be configured by executing `smbpasswd -w mysecret`.
3. The `smb.conf` file needs to reflect either the `ldap suffix` or `ldap idmap`.
4. User database synchronization must take place between the PDC and BDC for both local and Samba accounts; you can use the LDAP master-slave relationship for this.
5. The `netlogon` share requires replication between the PDC and BDC. This a good task for a login script.
6. For clients to find the domain controllers, you'll need to either enable NetBIOS over TCP/IP or add the appropriate service records to DNS.
7. You must edit the `slapd.conf` file to reflect the master LDAP and slave LDAP relationship.

If you're content with an NT-style domain and aren't integrating your Samba servers into an Active Directory infrastructure, and if you're unfamiliar with configuring LDAP for PDC and BDC servers, many sites on the Internet have excellent, well-documented configurations. To start with, however, visit <http://www.samba.org/samba/docs/man/Samba-Guide/happy.html#ldapsetup>.

Using Samba with Kerberos Authentication

You can now take Samba to the next level by using a Kerberos authentication server. To configure this option, you'll need to modify the `smb.conf` configuration file to reflect the name of the realm and the name of the authenticating server. In the `global` section, you need to add `realm = <Realm_or_Domain_Name>`, `security = server`, and `password server = <FQDN_of_Kerberos_Server>`. When you use a Kerberos server for authentication, you don't need to use the `smbpasswd` file; however, if you want nondomain members to log on, you'll need the file. When you put it all together, your configuration file for Samba should look similar to Listing 8-1.

Listing 8-1. Samba Configuration File

```
[global]
realm = <Realm_or_Domain_Name>
workgroup = DOMAIN_NAME
netbios name = HOST_NAME
security = server
```

```
password server = <FQDN_of_Kerberos_Server>
encrypt passwords = yes
domain master = yes
local master = yes
preferred master = yes
passdb backend = tdbsam
os level = 33
preferred master = yes
domain logons = yes
logon path = \\%L\profiles\%u
logon drive = H:
logon script = logon.bat
logon home = \\%L%\u\winprofile\%m
domain admin group = root administrator
time server = yes
[netlogon]
path = /netlogon
read only = yes
write list = ntadmin
[profiles]
path = /profiles
read only = no
create mask = 0600
directory mask = 0700
[homes]
read only = no
browseable = no
guest ok = no
map archive = yes
[dir_share]
comment = Shared Directory
path = /data/share
read only = no
oplocks = no
guest ok = no
```

Adding Samba to Active Directory

Adding a Samba server as a member server to an existing Windows 2000 or Windows 2003 mixed-mode domain requires your VM to have an account in the domain. In addition, you'll need to add or alter several lines in the `smb.conf` file to use Kerberos authentication. You'll need to configure the file to reflect the name of the domain and the name of the domain controller. In the global section, you need to add `realm = <Domain_Name>`, `security = ADS` and `password server = <FQDN_of_DC>`. When you use Active Directory for authentication, you don't need to use the `smbpasswd` file; however, if you want nondomain members to log on, you'll need the file. When you put it all together, your configuration file for Samba should look similar to Listing 8-1. As you probably already noticed, this process is similar to Samba using a

Kerberos server. The exception is the `security = setting`. In addition, being that Active Directory creates DNS records for KDCs, it's not necessary to modify the `kbr5.conf` file to reflect the name of the authenticating server.

Before attempting to add the system to the domain, ensure that your current account has the ability to authenticate to it:

```
kinit <USERNAME>@<DOMAIN_NAME>
```

To create the computer account for your Samba VM, execute the `net ads` command with administrative privileges:

```
net ads join -UAdministrator
```

Execute the `net ads` command again to join the VM to the domain:

```
net ads join "organizational_unit"
```

Verify that your Samba VM is part of the domain by opening Active Directory Computers and Users to see if the computer account exists in the organizational unit you specified. If you run into problems, make sure you're using the correct case for your usernames, domain, and passwords, because Kerberos is case sensitive.

Setting Up Samba DFS Shares

Samba has the capability to serve Microsoft DFS shares. You know from our earlier discussion that DFS makes files and directories appear to be in a single directory tree on a single server when they're really distributed across many disparate servers. By aggregating many network file resources in one file system, network users can more easily find documents by using the single point of access DFS creates.

Being that you want to configure Samba to serve DFS shares, you'll first need to create a DFS root directory on the server as a share point. You then need to modify the Samba configuration file to offer the share.

The DFS share can be any directory on your Samba server. When creating the directory, make sure it fits with your administrative practices. For simplicity, we'll show how to create a share off the `/` directory:

```
mkdir /dfs
```

The root user account should own the directory, and you should set the permissions to 755. If you created the directory using an administrative account other than root, use `chown` to reflect root as the directory owner:

```
chown root:root /dfs  
chmod 755 /dfs
```

You can add subdirectories in the DFS share to grant users the ability to physically store files on the Samba server:

```
mkdir /dfs/client_files
```

Creating additional shares in the dfs directory will treat a shared resource like a normal share. Being that creating simple shares isn't the purpose of DFS, we'll now show how to use DFS for what it's supposed to do—logically group disparate network resources into a single reference or directory tree. Linux groups shares by using *symbolic links*. You can think of a symbolic link as a shortcut to a resource or as a pointer. You'll use the `ln` command to create the links. Before you start creating links, first edit the `smb.conf` file and add the requisite code to enable DFS. For example, use `vi /etc/samba/smb.conf`. Be sure to substitute the correct path for your particular install of Samba.

Now follow these steps:

1. Create a stanza in the configuration file defining the DFS root:

```
[dfs]
path = /dfs
```

2. Enable DFS by adding `host msdfs = yes` to the global stanza:

```
[global]
host msdfs = yes
```

3. Finally, restart Samba:

```
service smb restart
```

4. It's best to create the symbolic links from within the DFS root directory:

```
cd /dfs
```

5. Create a link to a share on a file server called `vmfs01` called `data`:

```
ln -s 'msdfs:vmfs01\data' data
```

6. If you shared an entire drive on a server, you use the `ln` command similarly:

```
ls -s 'msdfs:vmfs01\d' vmfs01-d_drive
```

As you've probably observed, the usage is similar to UNC, and the command follows the form `ln -s msdfs:<server_name>\<server_share> <link_name>`. The `-s` option means you're creating a symbolic link. As a few pointers, remember to use lowercase letters when creating shares, and don't worry about some commands reporting DFS links as broken links, such as the `file` command. Samba knows that DFS symbolic links are on remote systems and will direct clients accordingly.

For easy access to the DFS root, Windows clients can map a drive to the DFS root, and Linux clients can mount the DFS root. From within Windows Explorer, select Tools ► Map Network Drive; then select the drive letter to assign and specify the UNC path. For Linux clients, you can use the `smbmount` command:

```
smbmount //<samba_server>/<dfs_root_directory> /<local_dfs_mount_point>➡
-o username=<username>
```

or the mount command:

```
mount -t smbfs -o username=<username> //<samba_server>/<dfs_root_directory>
```

Introducing AFS

AFS is the distributed file system research project started at Carnegie Mellon University from which the modern commercial release of AFS has its roots. The project used *Andrew* in its name to honor the school's founder Andrew Carnegie. Like DFS, AFS is a client-server distributed file system that presents a unified view of shared directories on servers and workstations without regard for the physical location of shared resources. AFS is now supported by IBM Pittsburgh Labs where a branch of the commercial AFS source has been released into the public domain as OpenAFS for development and maintenance.

ArlaAFS is another free AFS product. You can read more about ArlaAFS by visiting <http://www.stacken.kth.se/projekt/arla/>. ArlaAFS has several cool features but isn't as heavily documented as OpenAFS, so we'll base our discussions on OpenAFS.

Note You can compile AFS into Samba using the `--with-dfs` option (`--with-afs` is an older implementation); don't confuse this with Microsoft DFS, `--with-msdfs`.

AFS relies on the typical client-server relationship, and several versions of the software are available for varying operating systems, from Windows to Unix. The file servers utilizing AFS are responsible for file storage, file management, file requests, and security. Clients in the AFS architecture provide the end user with a way to interact with the system. A site running its own AFS system is called a *cell*. Cells are independent of other sites and are governed by the site administrator. An administrator can join a site cell to other cells where the local filesystems are joined into a global filesystem. End users accessing a global filesystem still see a single view of the file system and need to know only the location of a file within the file tree, not the location of the file server housing the files; the namespace provided by AFS maintains a uniform file tree no matter the access point of a user.

Every site participating in the global namespace agrees to follow the same set of naming conventions for pathnames. Creating the AFS filesystem requires you to consider the needs of the Cache Manager and not just the layout of the file tree itself. When creating the namespace, you should adhere to the following:

- Pathnames begin with `/afs`.
- The cell name is the second component is the `/afs` pathname, such as `/afs/<cell_name>`.

- Tertiary and deeper levels in the AFS file tree are site administered. However, best-practice conventions dictate several third-level directories, as listed in Table 8-6.

Table 8-6. *Suggested Tertiary AFS Pathnames*

Directory	Purpose
common	Shared access to all files and programs for all authorized cell users
public	Reserved for lookup and read access to anyone having access to the filesystem, including foreign users
service	Used to help cells coordinate sharing
sys_type	Storage for client and server binaries
usr	Home directories local cell users
wsadmin	Reserved for the package program, which facilitates client configuration and updates

A *local cell* or *home cell* refers to the cell an end user initially authenticates, and all other cells are referred to as *foreign cells*. We'll later discuss how to make foreign cells visible in the local file tree. Joining a global filesystem doesn't remove local administrative control or bypass ACLs. Users must have the appropriate permissions to use information in other cells. Access to the global filesystem takes place through the user's local Cache Manager. The Cache Manager resides on the user's computer and is responsible for retrieving files and maintaining the cache.

An AFS server runs several services to make the distributed file system function properly:

- The file server is responsible for delivering, saving, and storing files for AFS clients.
- The authentication server maintains the authentication database, provides secure network communications by verifying user identities, and facilitates identity verification for participant transactions.
- The protection server controls access to directories and files for users and groups.
- The volume server manages volume tasks, such as moving volumes between servers and balancing workload between participating systems.
- The volume location server maintains the volume location database that records the location of volumes on file servers and creates the illusion of transparent file access for clients.
- The backup server allows the administrator to back up volume data to tape by maintaining the backup database.
- The basic overseer server monitors the other AFS services to ensure they're functioning. This service aids administrators in system administration.
- The update server distributes AFS software updates and configuration information to file servers to maintain software release homogeneity.
- The salvager is used in the event of the file server or volume server failing to repair inconsistencies.
- The NTP daemon synchronizes system clocks so that the AFS distributed database functions correctly for all system participants.

The main administrative unit in AFS is the *volume*. Volumes logically correspond to directories in the AFS file tree and group files stored in the tree's directory on a single partition. A volume is a conceptual container that's used to store related files and resides on a system partition; it can have space quotas applied. Volumes tend to be small in size to aid in administration, such as backing up volumes or the state of volumes or replicating volumes to other servers to increase availability. A *mount point* associates a directory with a volume, and the directory is referred to as the volume's *root directory*. A mount point is like a symbolic link in the AFS file tree that associates a volume with the files in a directory.

Performance for AFS is increased by using replication and caching. Replication places a read-only copy of files on a different server. Read-only attributes afford AFS the ability to balance load across several servers. On the client side of things, caching temporarily stores the location of AFS files. This increases performance because network requests don't have to be generated every time the client needs access to the same file; the local Cache Manger already knows the location of the file and can fetch it for the end user.

AFS provides security by using mutual authentication and ACLs. This is the process of a server and client proving their identity to each other by knowing something in advance, such as a password. If a client isn't on the list of permitted users, access is denied. In addition to servers protecting resources using mutual authentication, users can create their own ACLs to manage who has access to their personal files.

Before creating the first cell for your site, you should consider a few issues, such as the cell name, differences between AFS and system file and directory security, volume administration, and client-server administration.

Whether you choose to be a part of a global namespace or not, your first AFS cell should adhere to standard naming conventions for several reasons:

- Changing the cell name later is difficult and administratively taxing.
- Your home cell name is the second component in the AFS pathname and is what makes the cell unique in the AFS global namespace. Unique cell names avoid path and file-space conflicts.
- Cell performance can be adversely affected because of the way programs and processes work. For instance, some commands execute in the cell of the client computer's membership.
- The cell name is involved with security because the authentication server uses the cell name with the user's password for encryption keys. Changing a cell name means the authentication database passwords won't match passwords created by the login utility; users will be locked out of the file space until updates are made to every machine's `ThisCell` and `CellServDB` files.

As you know, ACLs provide file and directory security with AFS, and even the end user can use AFS to grant or deny access. Unlike Unix-like operating systems, AFS doesn't solely rely on mode bits for security. AFS disregards directory mode bits and uses the first set of owner mode bits for security in conjunction with the ACL. The ACL has seven permissions that can be granted.

Being that all the files in one directory inherit identical security settings from the ACL, if strict security is required for a specific file, a separate directory must be created. AFS furthers security using blanket permissions with system groups: `system:anyuser` (literally any user of the filesystem) and `system:authuser` (authorized filesystem users only). The `system:anyuser` group is excellent for making information publicly available to anyone visiting a filesystem, such as via notices or documentation, and `system:authuser` is good for making common applications and general file resources available to local users, such as via text editors' site documentation.

AFS volumes should be created in a manner that eases administration. To that end, be aware that a directory in the file tree can be mounted to only one volume. When creating volumes, the directories at the top three levels of the file tree should be configured with separate volumes. Though you can create as many volumes as you want, this however doesn't mean every directory in the file tree needs its own volume. For instance, you wouldn't necessarily want the subdirectories a user creates within their home directory to be on separate volumes.

Note When creating volumes, you'll see they're assigned a space quota of 5000KB blocks. In addition, the ACL of the volume's root directory is set to grant all permissions to the `system:administrators` group.

When naming volumes, you should be aware of some restrictions; the following list is by no means comprehensive, though:

- Including extensions, the maximum volume name length is limited to 31 characters.
- Read/write volume names can be 22 characters long.
- Don't manually add `.readonly` and `.backup` extensions to volume names. The volume server automatically adds extensions.
- `root.afs` and `root.cell` must exist and be mounted at `/afs`.

Tip When naming volumes or dealing with OpenAFS in general, refer to the <http://www.openafs.org> Web site for help. Complete installation and administration documentation is available with thorough explanations and helpful advice.

Administering AFS servers requires you to take into account the four roles a file server performs. If you have more than one server, you can distribute these roles across several servers; otherwise, you'll configure a single server as a simple AFS server. Table 8-7 outlines the four roles any server can perform.

Table 8-7. *The Four Roles of an AFS File Server*

Role	Responsibility
Simple file server	Stores and delivers files to clients
Database server	Runs replication processes: authentication, backup, protection, and volume location
Binary distribution server	Distributes server binaries to other servers
System control server	Distributes common server configuration files to other servers and is a time source

You can create a cell with a single server or a cell with a server for each AFS role. Creating a cell with a single server will assign the file server and database server roles to the machine and will be the focus of our AFS implementation.

You can redirect common client files to AFS to conserve disk space if necessary. More important, you can manage client configuration files by using the package program that defines global configuration settings. The package program compares a client's hard disk with a configuration file for differences and then updates the client's disk by copying information from AFS. In addition, you can direct the package program to delete files on the client and reboot the machine if necessary. The package program employs a prototype file to determine which files reside on the client disk and to identify those that are linked to the site's AFS. Some benefits of using the package program are all site machines are automatically configured and reconfigured, and organization disk structures are maintained across a site.

Implementing AFS

AFS is an enterprise-class file system providing a common namespace without regard for physical boundaries to its clients. In a really reductive sense, you can think of it as creating your own file manager that spans all systems in the network. Files can be stored on disparate physical machines and viewed as if they were available in the local file tree; moreover, the administrator or the end user can apply a varying level of security measures to the system. You can download OpenAFS from the Internet for Linux or Windows at <http://www.openafs.org/release/latest.html>.

The following steps provide a basic overview of a Windows install; they're followed by a Linux installation and configuration overview:

1. When launching the executable, you'll be required to choose the default language before it self-extracts and starts the installation wizard. Be sure you have an NTFS-formatted partition available before continuing.

2. You'll first be presented with the Choose Components screen; at a minimum, install the AFS client and AFS server. If this is your first experience with OpenAFS, install all the available options, as shown in Figure 8-5. You'll want to explore them later. In a VM environment, you can always start over with little time penalty. If you're curious about the purpose of each option, the Description field briefly explains them.

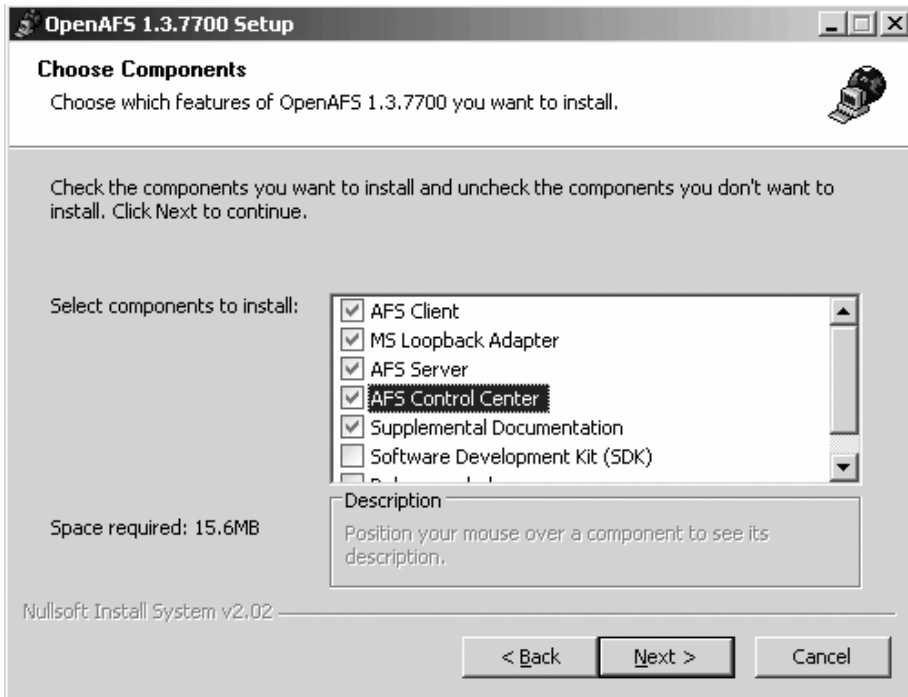


Figure 8-5. *OpenAFS component options*

3. After choosing the default installation directory for OpenAFS, you'll need to specify the location of the CellServDB configuration in order to contact the AFS file server. Use the default for the server, Use Package CellServDB File.
4. Being that you don't have a preconfigured CellServDB file, on the Cell Name Configuration screen select Next.
5. On the AFS Credentials screen, leave the defaults and select Install. The installation takes several moments, especially on a VM. Upon completion of the install, you'll be asked to reboot your server.

With OpenAFS installed, you can move onto configuring AFS for your Windows system server. You may receive a message stating that AFS failed to start, which is fine, seeing how AFS isn't configured. The configuration process starts with the AFS configuration wizard, which automatically starts upon login. Select Next on the Quick Start screen, and follow these steps:

1. On the Cell and Server Information screen (see Figure 8-6), select *This Will Be the First Server in a New AFS Cell*. For the cell name, you'll need to choose a unique ID, such as your domain name, and supply a good server password. The AFS server cell's principal account uses this password. AFS servers obtain tokens as principals, and the authentication server's ticket granting service (TGS) uses the password to encrypt tickets that clients present during mutual authentication. If you have questions as to what you should do, the Help button offers excellent information. For instance, you'll find that you should choose a cell name fewer than 64 characters using lowercase letters ending in a conventional suffix (.com, .edu, .mil, and so on). Underscores and numbers are acceptable, but you should keep in mind the conventions of cross-platform operating system portability.

AFS® Server Quick-Start Wizard

AFS Server Quick-Start Wizard

Current Step:

Cell and Server Information

An AFS cell is a collection of servers that provide a uniform namespace for file access.

☐ Make this host a server in an existing AFS cell
☒ This will be the first server in a new AFS cell

Cell Name
 Enter the name of the AFS cell.
 Cell name:

Server Password
 Enter the password for the AFS principal. All AFS servers get tokens as this principal.

Name:
 Password:
 Verify password:

Figure 8-6. Cell and server information

2. The wizard continues the install with the Administrative Information screen, where you must supply the information for the cell's administrative account. As shown in Figure 8-7, leave the default for the AFS user ID.

The screenshot shows the 'AFS® Server Quick-Start Wizard' window. On the left is a dark sidebar with the title 'AFS Server Quick-Start Wizard' and a circular arrow icon. Below the icon, it says 'Current Step:' followed by a progress bar with a triangle pointing to 'Administrative Information'. The main area is titled 'Administrative Information' and contains the text: 'The information below is needed to create the administrative account for this new cell.' There are two sections: 'Account Name and Password' and 'AFS UID'. The 'Account Name and Password' section has three text boxes: 'Name' (containing 'admin'), 'Password' (containing 'xxxxxxx'), and 'Verify password' (containing 'xxxxxxx'). The 'AFS UID' section has the text 'Enter the user ID for the AFS administrative account:' followed by two radio buttons. The first radio button is selected and labeled 'Use the next available AFS UID'. The second radio button is labeled 'Use this AFS UID:' followed by a text box containing '0'. At the bottom are four buttons: 'Cancel', '< Back', 'Next >', and 'Help'.

Figure 8-7. *Administrative information*

3. Being that you're configuring your system as the first server in the cell, the wizard will continue the installation after you opt to configure your computer as an AFS file server. You'll then need to verify that the server will also be configured as a database server and backup server. If you configure the database server as a backup server, all database servers must also be configured as backup servers.

4. You'll next be required to dedicate one NTFS volume to the AFS partition creation process; the AFS partition delivers files and programs to AFS clients. The two are virtually the same except that the wizard installs specialized metadata for tracking AFS structures. Ideally, the partition will be an unutilized volume on your server. As depicted in Figure 8-8, you should end the AFS partition name with the server's physical drive letter.

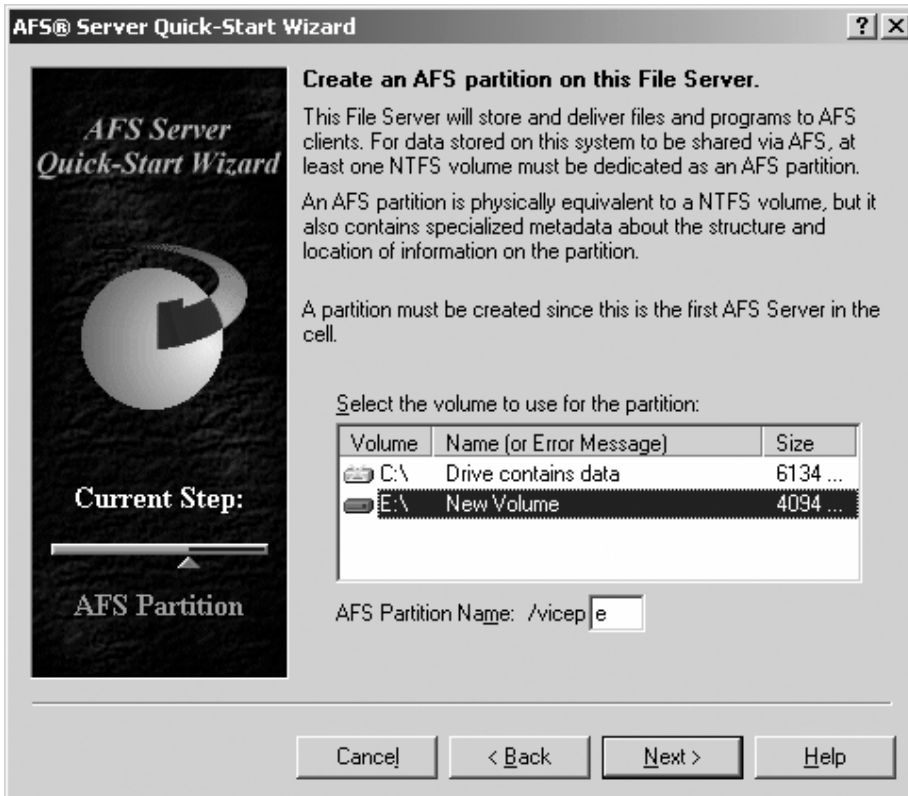


Figure 8-8. AFS partition creation

Note All AFS servers must have one partition designated for storing AFS volumes, and all AFS volumes must be designated as AFS partitions. You can create AFS partitions only on empty NTFS volumes.

5. Select Next on the root.afs and root.cell creation screen. This is an automatic procedure for the first server in the cell along with replicating the cell's root volumes on the next screen.
6. On the System Control Server screen, elect to make your server the system control server. This provides the common configuration files to AFS server system control clients. When you install the Control Center with the AFS server or client, you don't have to configure it.
7. The wizard will end with a summary screen, as shown in Figure 8-9, where you have the ability make changes or continue the configuration. Assuming you're satisfied with your choices, select Configure.



Figure 8-9. Host cell final configuration screen

After you reboot your server, you'll need to create user accounts and AFS directories for your clients to use. This process is straightforward and requires you to have a good plan in place for your directory tree structure. You can start by creating the volumes listed in Table 8-6 for a best-practice implementation.

Start by creating your first user. Select Start ► Programs ► OpenAFS ► Control Center ► Account Manager. Log in with the admin password, select the Users tab, and click Create. As shown in Figure 8-10, supply a username and password. You can leave the User ID selection as the default. If the account is going to be an administrator account, you can add it to the system:administrators group by clicking Groups.

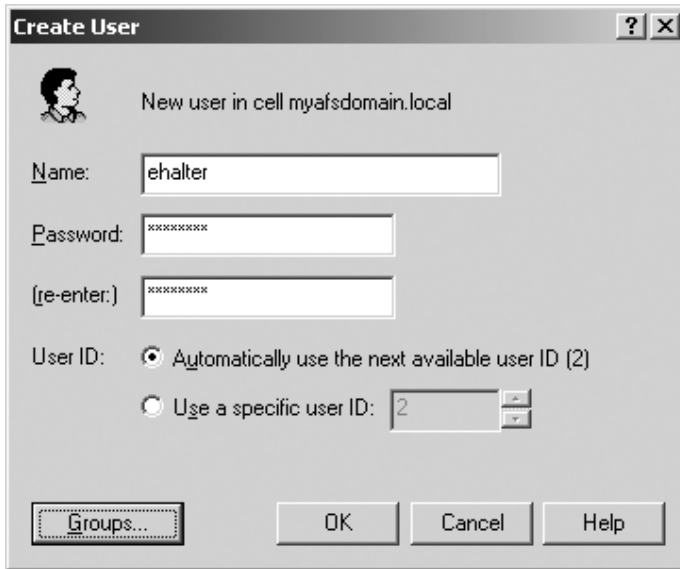


Figure 8-10. *Creating user accounts*

When you're done creating user accounts, close the Account Manager, and select Start ► Programs ► OpenAFS ► Control Center ► Server Manager. Log into the Server Manager (see Figure 8-11), and select the Volumes tab. Create volumes for common, public, service, sys_type, usr, and wsadmin. When creating the volumes, you can select Backup, which creates a copy of the volume within the site and appends .backup to the volume name. If necessary, change volume quotas as well.

With a remedial configuration for testing AFS completed, you can move onto client connectivity. You'll need to use the same executable as that of the server. However, you'll just install the AFS client. As shown in Figure 8-12, on the Choose Components screen, select AFS Client, MS Loopback Adapter, and Supplemental Documentation.

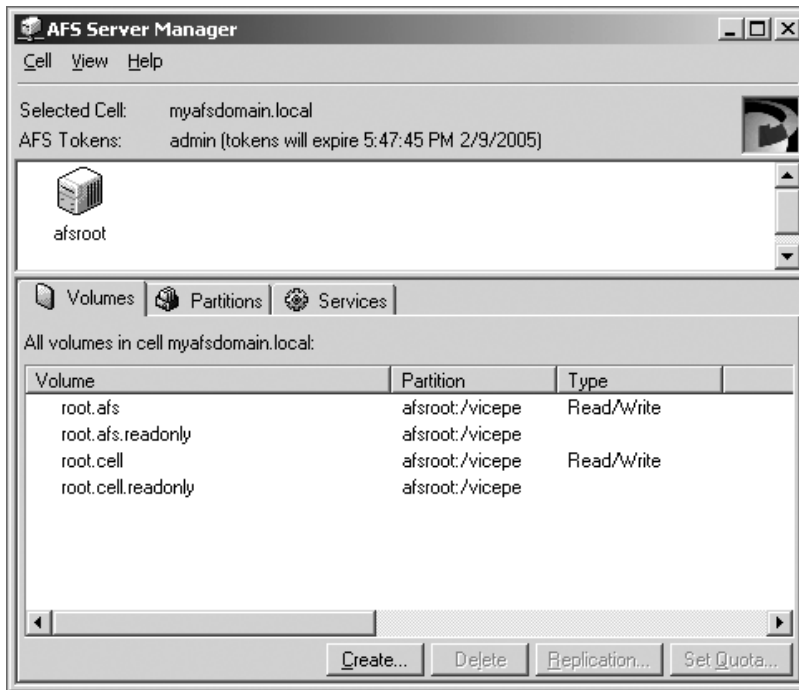


Figure 8-11. The Server Manager

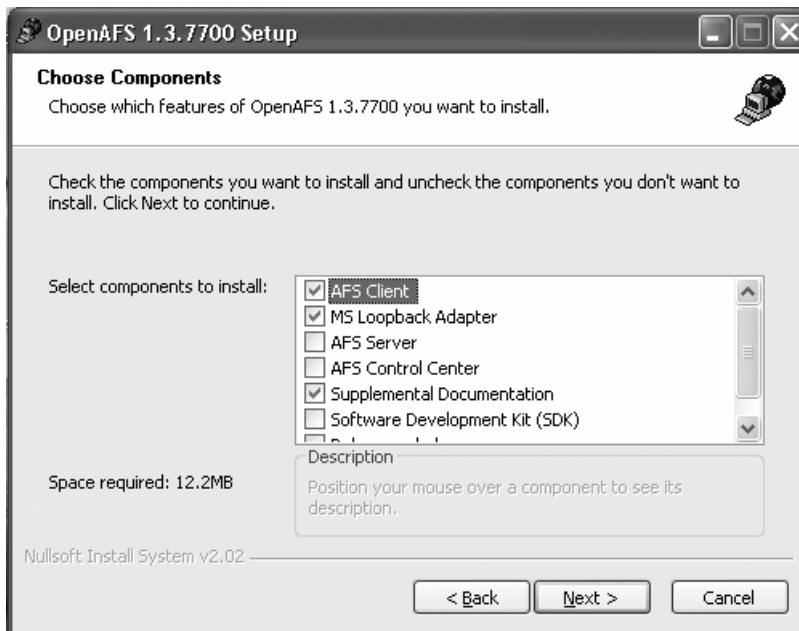


Figure 8-12. Client installation components

Now follow these steps:

1. Install the OpenAFS software in the default location, and on the CellSrvDB Configuration screen use the default setting for the CellSrvDB file.
2. On the Client Cell Name Configuration screen, shown in Figure 8-13, enter your cell's AFS name and leave the default options. The cell's name should be similar to your site's domain name, such as `apress.com`. Next, select Install on the AFS credentials screen, and reboot your system upon completing the software install.

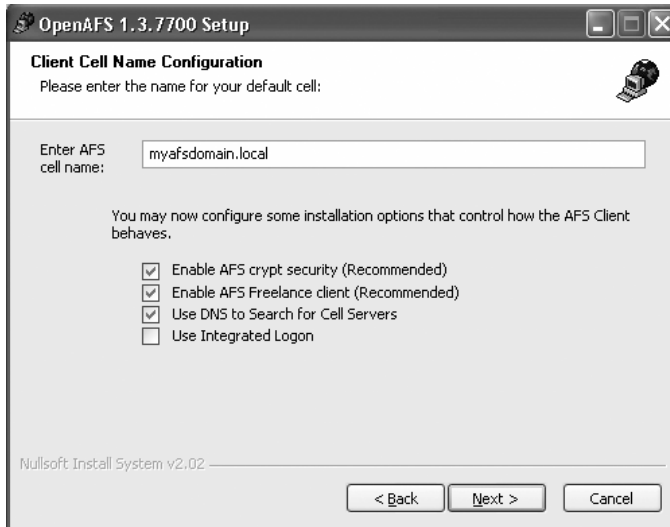


Figure 8-13. *Client cell name configuration*

3. Once your system reboots, it's time to finish configuring the AFS client. An open lock with a red X should be visible in your system tray. Double-click the lock to open the AFS client; the interface should look similar to Figure 8-14.

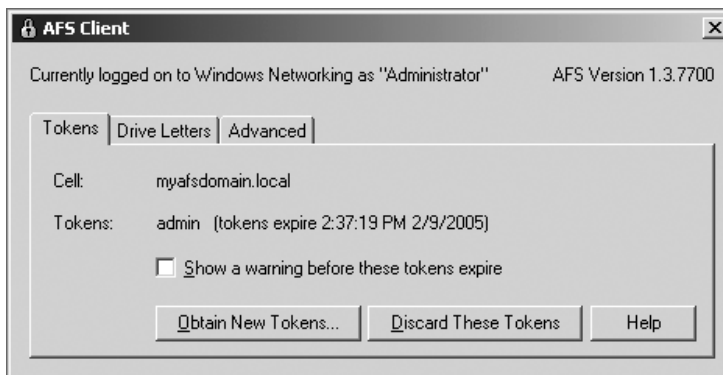


Figure 8-14. *AFS client configuration*

4. Select Add on the Drive Letters tab. We'll configure the client to use the H drive to connect to the usr share on the AFS server. Notice that in Figure 8-15 we're using the absolute path of the share on the AFS Path line. The Submount option will supply a friendly name for the mapped drive letter. We chose the word *home*.

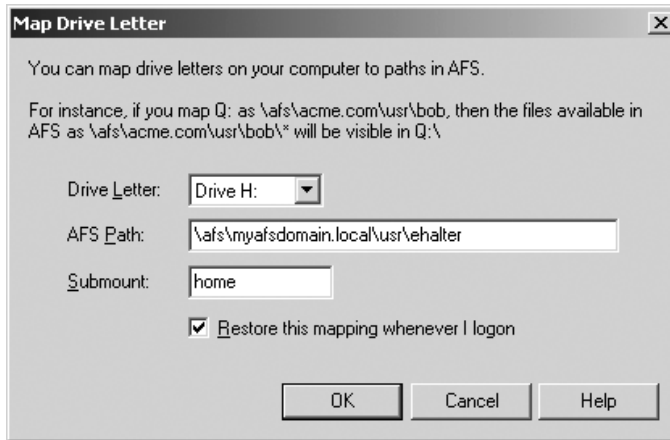


Figure 8-15. *AFS client drive mapping*

5. Select the Advanced tab, and click Configure AFS Client. Make sure the cell name correctly reflects your AFS configuration. On the Preferences tab, add your server's name, and on the AFS Cells tab, add your cell. When you're done with the Advanced settings, select the lock from the system tray and log in by requesting new AFS tokens, as shown in Figure 8-16. When you're done, you should see the drive letter mapping in Windows Explorer.



Figure 8-16. *Obtaining new AFS tokens*

Installing AFS for Linux Systems

To install AFS on a Linux system, you can download the software from the OpenAFS Web site. If you're new to OpenAFS, your best shot at getting a server up and running is to install the RPMs for your particular distribution of Linux. As a heads-up, we'll take a shotgun approach to installing AFS and assume you're installing AFS in a test environment; you'll have to think on your feet and make operating system/path adjustments to the steps as we blast through the configuration.

To get started, download all the packages for your flavor of Linux. If necessary, recompile the AFS kernel for your distribution of Linux. Make sure your server has sufficient disk space to be an AFS server for each role it will be playing. If you're setting up a VM as an AFS server, you can always add virtual disks later. Before getting started, make sure you've created a dedicated partition for AFS and mounted it as a directory in the format `/vicepxx`, where `xx` is one to two lowercase letters. Also, start the AFS initialization script. If you decide to use the RPMs, you'll need to recompile the AFS kernel for your distribution of Linux.

Note You may receive error messages during the configuration of your AFS server by using the `-noauth` option. If this happens, continue with the configuration, because this is reasonable being that `-noauth` disables authorization checking. Additionally, make sure your systems are time synchronized using a reliable protocol, such as NTP.

Follow these steps:

1. Verify that the necessary local directories for AFS to function were created during the install process, and start the AFS initialization script:

```
ls /usr/afs
ls /usr/vice
ls /usr/vice/etc
./etc/rc.d/init.d/afs start
```

2. Create at least one AFS partition or logical volume. The partition should be mounted at a directory named in the format `/vicepxx`, where `xx` is one to two lowercase letters. In our example, we created a second VM SCSI disk for AFS:

```
mkdir /vicepa
```

3. Edit `/etc/fst` to map the directory to the AFS disk partition:

```
/dev/sdb1 /vicepa ext2 defaults 0 2
```

4. Create the file system on the partition:

```
mkfs -v /dev/sdb1
```

5. Mount the partition:

```
mount /dev/sdb1 /vicepa
```

6. Change to the bosserver script directory, and execute it with the `-noauth` option. The `-noauth` option disables authentication to your cell and compromises security. Make sure you adequately protect your system during this phase of the configuration process. You can set your VMs to host-only mode or use a NAT device. If you unplug your patch cable or disable your NIC, make sure name resolution continues to function:

```
cd /usr/afs/bin
./bosserver -noauth
```

The script creates several directories and sets the appropriate permissions on the directories for the root user account. The directories created are as follows:

```
/usr/afs/db
/usr/afs/etc/CellServDB
/usr/afs/etc/ThisCell
/usr/afs/local
```

7. Verify bosserver created `/usr/vice/etc/ThisCell` and `/usr/vice/etc/CellServDB`:

```
ls -l /usr/vice/etc
```

8. Assign your cell name to the AFS server:

```
cd /usr/afs/bin
./bos setcellname <server_FQDN> <cell_name> -noauth
```

9. Use `bos setcellname` to set the cell name:

```
cd /usr/afs/bin
./bos setcellname <server_FQDN> <cell_name> -noauth
```

10. Issue `bos listhosts` to verify the name configuration:

```
./bos listhosts <server_FQDN> -noauth
```

11. Configure the four AFS database services to start: authentication server, backup server, protection server, and VL server:

```
./bos create <server_FQDN> kaserver simple /usr/afs/bin/kaserver➤
-cell <cell_name> -noauth
./bos create <server_FQDN> buserver simple /usr/afs/bin/buserver➤
-cell <cell_name> -noauth
./bos create <server_FQDN> ptserver simple /usr/afs/bin/ptserver➤
-cell <cell_name> -noauth
./bos create <server_FQDN> vlserver simple /usr/afs/bin/vlserver➤
-cell <cell_name> -noauth
```

12. Configure basic cell security, and create the authentication database for `admin` and `afs`. You'll be asked to supply and verify the passwords for each after passing the `create` command:

```
kas -cell <cell_name> -noauth
create afs
create admin
```

13. Use `kas setfields` to enable the administrator to use privileged `kas` commands, and then execute `quit`:

```
setfields admin -flags admin
quit
```

14. Add `admin` to the `/usr/afs/etc/UserList` file to use privileged `bos` and `vos` commands:

```
./bos adduser <server_FQDN> admin -cell <cell_name> -noauth
```

15. Use `bos addkey` to configure the server encryption key:

```
./bos addkey <server_FQDN> -kvno 0 -cell <cell_name> -noauth
```

16. Execute `pts createuser` to create the admin protection database entry. You should use the same local user ID for `root` as `admin`:

```
./pts createuser -name admin -cell <cell_name> -id <AFS_UID> -noauth
```

17. Use `pts adduser` to add `admin` to the `system:administrators` group:

```
./pts adduser admin system:administrators -cell <cell_name> -noauth
```

18. Run the `bos restart` command to restart the server processes with the newly created encryption key:

```
./bos restart <server_FQDN> -all -cell <cell_name> -noauth
```

19. Initialize the file server processes, file server, salvager, and volume server, and verify that they start:

```
./bos create <server_FQDN> fs fs /usr/afs/bin/fileserver
/usr/afs/bin/volserver
/usr/afs/bin/salvager -cell <cell_name> -noauth
./bos status <server_FQDN> fs -long -noauth
```

20. Create the `root.afs` volume:

```
./vos create <server_FQDN> /vicepa root.afs -cell <cell_name> -noauth
```

21. Use `bos create` to start the update server, making it a binary distribution point:

```
./bos create <server_FQDN> upserver simple "/usr/afs/bin/upserver➤
-crypt /usr/afs/etc -clear /usr/afs/bin" -cell <cell_name> -noauth
```

Your AFS server is now configured as a binary distribution server, database server, file server, and system control server. You can configure your server as a client machine to help aid in administration. You'll need to perform several tasks to add AFS client functionality to your server. Please keep in mind that you'll need to adjust virtually every step for the distribution of Linux you're using. We're merely providing you with an installation summary.

Follow these steps:

1. To begin, you'll need to download and install several packages from the OpenAFS Web site: `openafs`, `openafs-kernel`, `openafs-kernel-source`, `openafs-compat`, and `openafs-client`. If necessary, recompile the AFS kernel for your distribution of Linux. Next, make sure PAM is configured to allow log in. Edit `pam.d/login` to reflect AFS's needs. Make sure you change the path of the statement to meet the requirements of your operating system:

```
auth sufficient /lib/security/pam_afs.so try_first_pass ignore_root
```

2. Execute `./configure` and then make from within the `open-afs-kernel` directory.
3. Copy the new module in the `modload` directory:

```
cp MODLOAD-XXXS/libafs-XXXS.o /usr/vice/etc/modload/
```

4. From the `modload` directory, add the module to the AFS `SymTable`:

```
cd /usr/vice/etc/modload
./afsmodule -f SymTable -g libafs-XXXX.o
```

5. Edit `ThisCell` to add your local cell information, or copy it from the server. You'll probably get better results by copying the file from the server to your workstation:

```
cp /usr/afs/etc/ThisCell ThisCell
```

6. Add local cell information to the `CellServDB` file. The format of the file is as follows:

```
>cell_name      #organization
IP_Address      #machine_FQDN
```

A typical example might look like the following:

```
>apress.com      #Apress (home cell)
192.168.100.1    #db1.apress.com
192.168.100.2    #db2.apress.com
```

7. Create a local AFS cache file on your hard disk to increase performance for frequently accessed files. Assuming you have enough memory, you could create the cache in RAM. To take advantage of the file, you'll also need to initialize the Cache Manager:

```
mkdir /usr/vice/cache
# echo "/afs:/usr/vice/cache:25000" > /usr/vice/etc/cacheinfo
```

8. Create the `/afs` directory for the Cache Manager to mount the filesystem:

```
mkdir /afs
```

9. Copy the `afsd` file from the `/usr/vice/etc` directory to `/etc/sysconfig`. Omit the `.conf` file extension:

```
cp /usr/vice/etc/afs.conf /etc/sysconfig/afs
```


10. Edit the `afsd` options file to set the startup values. You can adjust the startup values according to your needs, which will vary if you use a memory-based cache rather than a disk-based cache. You'll need to refer to the AFS documentation to tweak this. If you don't perform this step, the Cache Manger will supply its defaults:

```
vi /etc/sysconfig/afs
-stat 3500 -daemons 5 -volumes 150
```

11. Initialize the AFS client, and log in:

```
./afs start
klog USERNAME
```

12. You should be able to navigate to the AFS filesystem as you would any other directory on your computer using `cd`.

To make a foreign cell visible in a local cell's file tree, you must mount the foreign site's `root.cell` at the second level in the AFS file tree, mount AFS at `/afs` on client computers, and add a database entry in `CellServDB` for the foreign cell's database server. Follow these steps:

1. Use the `fs mkmount` command to mount the cell's `root.cell` volume at the second level:

```
fs mkmount <directory> <volume_name> -cell <cell_name>
```

2. Make the `afsd` program mount AFS at `/afs` by using a text editor to modify `cacheinfo`. The `afsd` program automatically mounts the first directory listed in `cacheinfo`. Failing to mount clients at `/afs` will prohibit clients from viewing the filesystem of foreign cells that are conventionally mounted using best practices. The first entry in the `cacheinfo` file looks like the following:

```
/afs:/usr/vice/cache:25000
```

3. Add a database server entry to `CellServDB` to client machines that need access to the foreign cell's filesystem. One method is to use a text editor.

```
vi /usr/vice/etc/CellServDB
```

Remember that the format of the file is as follows:

```
>cell_name    #organization
IP_Address    #machine_FQDN
```

A typical example might look like the following:

```
>apress.com    #Apress (home cell)
192.168.100.    #db1.apress.com
192.168.100.2   #db2.apress.com
```

If you want to make your filespace visible to foreign cells, remember that foreign users are granted default permissions of the `system:anyuser` group by a directory's ACL. Generally, the permissions are limited to `l` (lookup) and `r` (read). If you want to grant foreign users additional access, you can modify the directory's ACL or give the foreign user a local account. In addition, you should follow best practices and create the upper level of the AFS filespace as suggested on the OpenAFS Web site. You'll also want to visit the OpenAFS Web site to read up on how to increase security and increase the usability of your cell by making documentation and binaries available to AFS clients.

Summary

In this chapter, we discussed how to configure and deploy your own custom filespace with DFS and AFS virtual file systems. In addition, we covered how to integrate open-source operating systems with DFS in a Microsoft domain. We touched on basic security issues and best practices for distributed file systems. In Chapter 9, you'll explore another common virtualization technology used today, server clustering. We'll introduce how to deploy server clusters and discuss server clustering management techniques.



Implementing Failover Clusters

For many organizations today, data must be available practically every second of every day. Unfortunately, computers, networks, and storage fail, no matter how much you pay for them. This is where clustering comes in. Simply put, a *cluster* consists of two or more computer systems acting and managed as one. Clients access the cluster using a single host name or IP address, and one of the systems in the cluster answers their request. We'll explain cluster configurations shortly.

The purpose of cluster technology is to eliminate several single points of failure. When data availability is paramount, clustering is ideal. Consider the following single points of failure you can eliminate by using a failover cluster:

- Network card failure
- Processor failure
- Motherboard failure
- Power failure
- Cable failure
- Network failure

With a cluster, you can basically eliminate nearly any hardware failure associated with a single computer. If hardware associated with one system fails, the other system will automatically take over. Here are some of the typical reasons for implementing a cluster:

- To increase the availability of mission-critical applications such as Oracle database servers and Exchange mail servers
- To increase the availability of file and print servers
- To reduce downtime caused by hardware failures and routine maintenance

In this chapter, we'll start by covering the necessary buzzwords you need to understand when planning and implementing failover clusters, and then we'll show you examples of failover clusters on two popular OS platforms: Windows Server 2003 Enterprise Edition and the Linux High Availability Project software running on Red Hat Enterprise Advanced Server 3.0. Since the point of this chapter is to get you familiarized and started with failover clustering, we won't cover every possible nuance and gotcha with failover clustering (countless books that are hundreds of pages long already do this). Instead, we'll show you how to successfully implement both Windows and Linux failover clusters.

Introducing Failover Clustering

Failover clustering typically employs some form of shared storage even though it's not a requirement with Linux clusters. Shared storage usually exists on an external shared storage bus (Fibre Channel, SCSI, or iSCSI), with access to the shared storage managed either by the cluster service software or by Distributed Lock Manager (DLM) software. The technology that determines how shared storage is accessed varies by OS and clustering application. Many OS clustering technologies, such as Windows Server 2003 clusters, manage shared disk access themselves. Others, often from third-party vendors, incorporate a DLM. To put all of this into context, Figure 9-1 shows a simple two-node cluster.

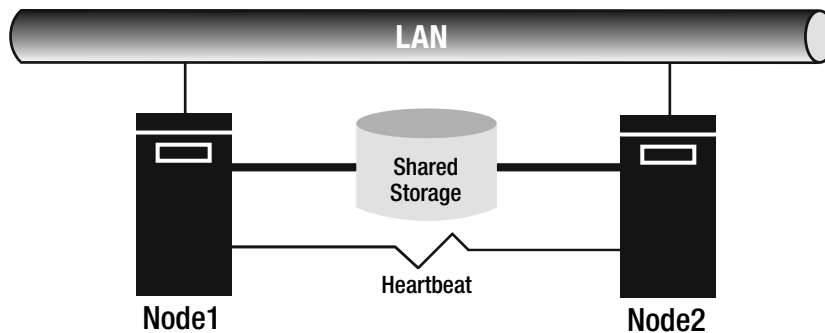


Figure 9-1. *Two-node failover cluster*

In Figure 9-1, two physical computers share a common storage source. Using the common shared storage makes data management in the cluster much easier. This is because with data living in one central location, you don't have to worry about keeping data synchronized on each node, assuming each node locally keeps its own copy of the cluster's data. Most often the shared storage has some form of built-in redundancy, such as with the use of hardware RAID. With applications that require both read and write data access and high availability, failover clustering is almost always a good fit. Shared storage can provide for the read/write access to a central data store, and the clustering application or service can ensure that if a failure is detected, the second (or next) node in the cluster assumes control of the virtual server being managed by the cluster.

If that last paragraph was a mouthful, let's break it down a little further. Failover clusters allow you to run one or more “virtual servers” on top of two or more physical computers. The physical computers in the cluster are most often referred to as either *nodes* or *real servers*, depending on the clustering application and operating system. The virtualized computer running on the cluster is usually called the *virtual server* or *virtual host*. A virtual server is controlled by only one node at a time. The node controlling the virtual server will lock the shared disk resources used by the virtual server so that no other node in the cluster attempts a write operation at the same time as the virtual server.

Also notice that Figure 9-1 has a connection labeled *heartbeat*. This connection is typically either an Ethernet connection via a crossover cable or a serial interface connection between the two nodes. For larger clusters, the heartbeat is typically connected via a dedicated switch.

Defining Essential Terms

Like nearly all aspects of IT, clustering comes with its own unique vocabulary. The following are the most common terms you'll hear:

- **Cluster communications:** This is the means by which two or more cluster nodes communicate. Communications can occur through a private (cluster node-only) network segment or mixed (cluster nodes and clients) network segment.
- **Cluster disk:** This is the physical hard drive that's shared by two or more cluster nodes using a shared data bus, which can be a SCSI or a Fibre Channel bus.
- **Private disk:** This is the hard disk on each physical node that's used to store the node's OS, applications, and swap file.
- **Cluster node:** This is the physical system participating in the cluster. The node is often referred to as a *real server* (RS) in Linux cluster implementations.
- **Heartbeat:** This is the periodic communication between cluster nodes using User Datagram Protocol (UDP) datagrams in order to determine if a node is running; each cluster node will listen for the heartbeat of all other nodes. The heartbeat connection can be established over the public LAN, a dedicated crossover cable, or a dedicated serial interface connection.
- **Quorum resource:** In Windows and Red Hat Linux clustering, this is the disk or disk partition that contains the cluster's management and configuration data and its recovery log.
- **Virtual server:** This is typically a collection of cluster nodes that appear as a single server to clients. Like all network servers, a virtual server will have a network name and IP address.
- **Failover:** This is the process of a virtual server moving from one physical node in the cluster to another. Failover is usually caused when another node in the cluster fails to receive heartbeat packets from the virtual server's active host node.
- **Failback:** This occurs when a virtual server's original or designated preferred host returns online following a failure. When a failback occurs, a virtual server is automatically moved by the cluster service back to its preferred host.

With an understanding of some of the terminology, you'll now examine the available cluster architectures.

Introducing Cluster Architecture

Failover clusters are characterized as either active-passive or active-active. The following sections describe the differences between these two configurations.

Active-Passive Clusters

In an *active-passive* cluster configuration, one node has complete control of the shared resources, and the other node monitors the active node for failure. If the active node fails, then a passive node will assume control of the shared resources, which typically is a disk storage array, possibly incorporating hardware RAID. Figure 9-2 shows the basic configuration of an active-passive cluster.

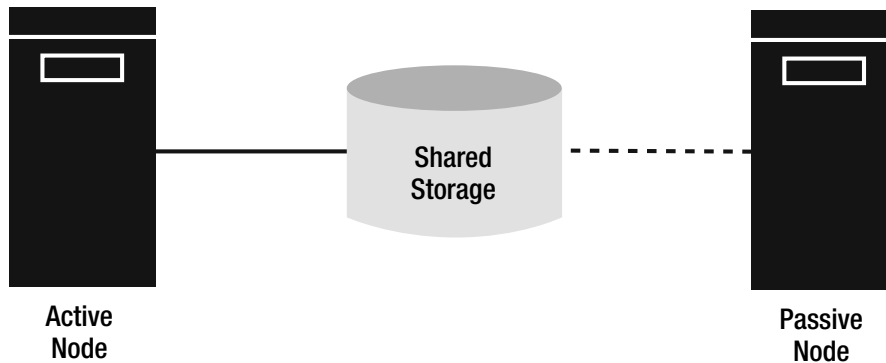


Figure 9-2. *Two-node active-passive cluster*

Both nodes are always physically connected to the shared storage, but only the node that controls the storage can access it. This is known as *shared nothing* architecture, which means that no shared resources can be accessed by multiple nodes simultaneously. In other words, only the active node can perform reads and writes to the shared disk. With the active-passive approach, one cluster node does nothing except monitor the active node for failure. For some organizations, this approach is more difficult to justify economically, since without a failure the passive node does little more than consume electricity.

Active-Active Server Clusters

In an active-active cluster configuration, two or more cluster nodes actively host virtual servers. Active-active clusters are usually an easier sell to the bean counters because each node in the cluster is actively hosting a virtual server and isn't merely sitting idle.

Tip An easy way to think of an active-active cluster is as two or more active-passive clusters combined.

For example, suppose you want to have failover capability for both your Exchange and SQL servers. You could store each server's data on separate physical disks in a shared SCSI disk array. Connect both servers to the array, and then install and configure the Microsoft Cluster Service. You can then look at your SQL server as the standby Exchange server, and vice versa. If one fails, the other server will act as both servers. In essence, you have an active-passive Exchange server with node 1, for example, being the active node and an active-passive SQL server with node 2 being the active node. When looking at the cluster as a whole, both nodes are active, giving you an active-active cluster. Figure 9-3 shows this configuration.

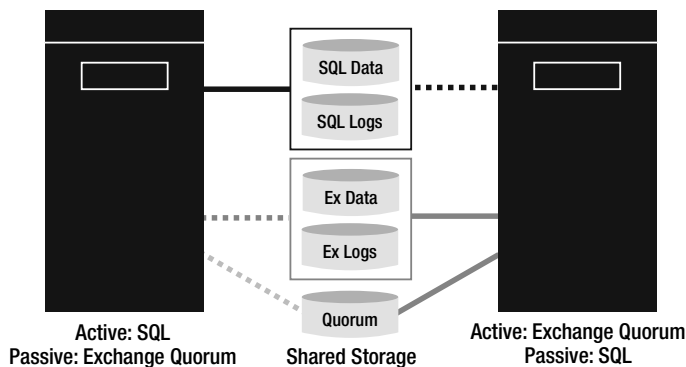


Figure 9-3. Exchange-SQL active-active cluster

Introducing N-tier Clustering

Many cluster application vendors describe their cluster architectures as either *N*-to-1 or *N*-plus-1. In an *N*-to-1 architecture, one node in the cluster is designated as the passive node, leaving it available to handle failover if an active node in the cluster fails. Figure 9-4 shows a three-node *N*-to-1 cluster. Notice that Node1 is active for Oracle and Node2 is active for Apache. If either active node fails, Node3 will assume its role. In this architecture, Node3 is always designated as the passive node, meaning that the primary active node returns online following a failure, and the service will failback to the primary node. While this approach offers simplicity, having automatic failback means that the failed service will be offline twice—once during the initial failover and again during the failback.

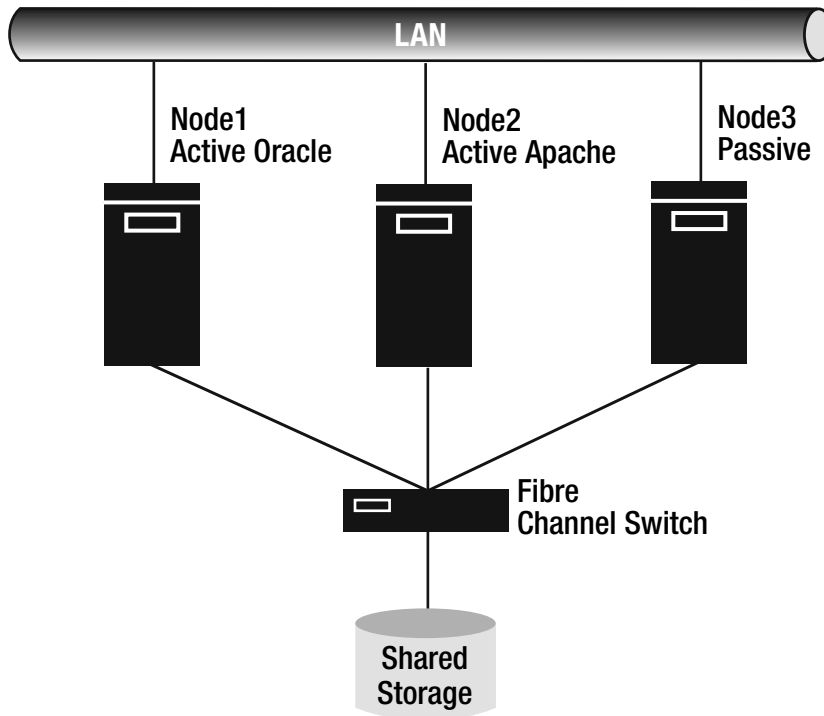


Figure 9-4. *Three-node N-to-1 cluster*

N-plus-1 clustering offers a different approach. With *N*-plus-1, a standby (passive) node can assume control of a primary node's service when the primary active node fails. However, when the active node returns to service, it then assumes the role of the passive node. This means that in time, the active node for each service managed by the cluster may be completely different than at the time the cluster was originally set up. However, automatic failback isn't an issue with this approach, thus providing for better overall availability.

Working with Failover Cluster Products

Now that we've covered the general architecture of failover clusters, we'll show a sample of some of the available clustering products. Table 9-1 lists some of the most common clustering solutions available for both Windows and *nix platforms.

Table 9-1. *Available Failover Cluster Products*

Service/Application	OS	Maximum Number of Nodes	Vendor URL
Linux-HA 1.x	Linux, FreeBSD, Solaris	2	http://www.linux-ha.org
Linux-HA 2.x	Linux, FreeBSD, Solaris	8 or more	http://www.linux-ha.org
Microsoft Cluster Service	Windows Server 2003 Enterprise/Datacenter	8	http://www.microsoft.com
Microsoft Cluster Service	Windows 2000 Datacenter Server	4	http://www.microsoft.com
Microsoft Cluster Service	Windows 2000 Advanced Server	2	http://www.microsoft.com
PolyServe Matrix Server	Windows 2000/2003, SuSE/Red Hat Linux	16	http://www.polyserve.com
Red Hat Cluster Suite	Red Hat Enterprise Advanced Server	8	http://www.redhat.com
Veritas Cluster Server	AIX, HP-UX, Solaris, Windows	32	http://www.veritas.com

As you can see, you have many choices when it comes to configuring failover clusters. While the configuration steps of the various clustering products differ, architecturally they're nearly identical. With that in mind, next we'll cover some general cluster planning considerations.

Planning for Failover Clusters

Failover cluster planning begins with choosing the right model as the foundation for the cluster service. Before getting to cluster model selection, let's take a moment to review the fundamental hardware requirements for failover clustering. The hardware requirements are as follows:

- At least one external shared disk connected to each node through a PCI SCSI, Fibre Channel controller (required in most commercial implementations), or iSCSI via a Gigabit Ethernet controller
- At least one NIC (two recommended) in each node
- Appropriate cabling and termination for shared storage

Note You must consider several issues when connecting cluster nodes to shared storage in a SAN. Chapter 12 will discuss these issues in detail.

Take the following points into consideration when planning a shared storage configuration:

- For Windows clusters, each disk must be configured as a basic disk, with each partition formatted using NTFS.
- Each SCSI drive and SCSI adapter must have a unique SCSI ID.
- All drives used by the cluster must be physically attached to each node in the cluster.
- Prior to installing the cluster service, power up each node individually and verify that the node can access the shared storage.

Tip The SCSI ID on most host adapters is set to 7 by default, so you'll need to change the ID of the second node's adapter in two-node clusters with shared SCSI-attached storage.

When planning the cluster network configuration, consider the following:

- Each virtual server must have a unique host name.
- Each network adapter and virtual server should have its own static IP address.
- For Microsoft clusters, all nodes in the cluster must be joined to the same domain, and a domain user account must be created for use by the Cluster Service.

Choosing the Right Model

Failover clusters are built using the format from any of five different models. The model you choose will depend on the current and future needs of your organization. The following sections will outline each model, its pros and cons, and how it's best used.

Single-Node Model

The single-node model is the simplest of all cluster models. While the phrase *single-node cluster* may seem like an oxymoron, trust us—it's not. Figure 9-5 shows the concept of a single-node cluster.

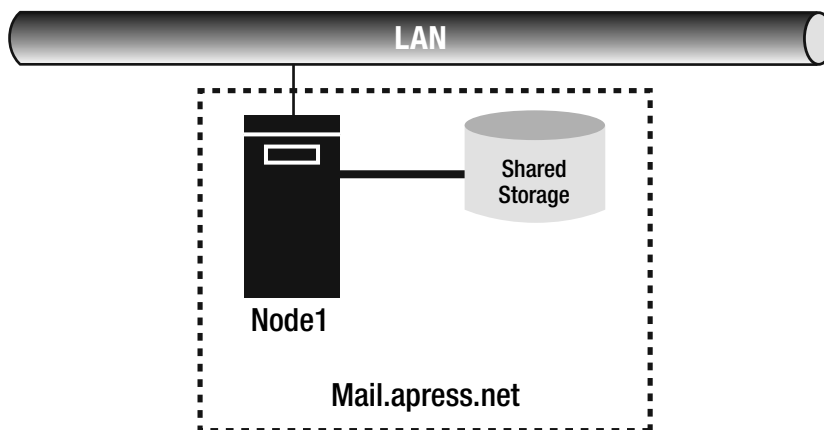


Figure 9-5. *Single-node virtual mail server cluster*

The single-node model has the following advantages:

- It can be the first stage of an eventual multinode cluster implementation.
- It allows you to take advantage of a cluster service's automatic-restart ability for failed applications.

Here are some reasons why you wouldn't want to use this model, though:

- No other nodes are available for failover.
- It has a single point of failure.
- You'd need to invest in external Fibre Channel or SCSI storage, even though internal storage for a standard server is much cheaper.

Taking advantage of the automatic-restart ability of clustering will keep your applications and services up more consistently. Also, if you're working with a small budget, the single-node model could be the first step in the eventual rollout of a multinode cluster.

Caution Not all applications can run in a virtual server on a single-node cluster. Always check the application's documentation, and test the configuration before placing the single-node cluster in your production environment.

Dedicated Secondary Node (Active-Passive) Model

The standard active-passive model provides the failover capability not found in the single-node model. This model type was shown earlier in Figure 9-2.

As you can see, with this model, the active node manages all cluster virtual servers, and the passive node stands by in the event the active node fails. Upon failure of the active node, the passive node will then become active, servicing all cluster virtual servers.

This model offers the following advantages:

- Provides failover capability
- Removes a single node's associated hardware as single points of failure

Here are two reasons why some avoid this strategy:

- The passive node doesn't manage any cluster resources or groups until the primary node fails.
- This requires investment in hardware that's utilized only when the primary node fails.

Note Remember that all cluster nodes must meet the minimum hardware and software requirements of all applications run in virtual servers. Each cluster node must have the hardware resources required to run all applications that have the capability to failover to it.

Some have trouble justifying a server whose job is essentially to stand by in the event that another server fails. Because of this, many implement active-active clusters instead, which will be described momentarily.

Dedicated Secondary Node (Active-Passive) Split Model

The active-passive split model operates similarly to the standard active-passive model. The primary difference between the two is that with the split model, the active node services both clustered and nonclustered applications.

Applications that don't support failover can't use the cluster's shared disks. Instead, all application data must be installed on a cluster node's local disk. Remember that with this approach, if the cluster node fails, so does the application. However, some applications can't be clustered, and this approach allows you to get more use out of the active cluster node.

High Availability with Static Load Balancing (Active-Active) Model

Many prefer the static load-balanced active-active model because it entails running clustered applications on each cluster node. This model makes it much easier to justify your cluster expenses. Figure 9-3 earlier showed the typical active-active cluster configuration.

With active-active clusters, each node in the cluster must be able to take on the resources required by virtual servers active in any other node. For example, if a clustered application requires 256MB of RAM, each cluster node must have 256MB of available memory in case it has to host the application. Remember that if a node fails, another node in the cluster may be slowed down by having to service too many applications. However, most would rather have two slow servers for a brief period of time than one failed server.

Here's a summary of the primary reasons for using the active-active cluster model:

- No servers are “wasted” as standby servers.
- Two applications can be statically load balanced over two nodes, as opposed to running both applications on a single node.
- Multiple instances of the same application can be statically load balanced over several nodes. This technique is helpful when a single server, no matter how large, can't handle the load placed on it by its clients.
- When all nodes are up, application performance is typically better than running multiple applications on a single node.

The lone disadvantage to this approach is that all nodes must have adequate hardware resources to assume the role of another node if needed.

Hybrid Model

As with other network concepts, the term *hybrid* implies a mix, or combination, of other models. The typical hybrid model involves active-active clusters that also service nonclustered applications. This model allows you to use your cluster nodes as network servers for other applications, as opposed to using them exclusively as a part of the cluster.

Once you've chosen your cluster model, your next course of action is to configure the cluster node hardware.

Configuring Cluster Hardware

In a perfect world, when you build a cluster you'll start with new servers with identical hardware configurations. While this isn't always possible, you should try to match the hardware configurations of each node as closely as possible.

To begin, any PCI SCSI or Fibre Channel HBAs that connect to shared cluster storage should be installed in identical PCI slots on each node. This is critical for clustered applications to see the data path to the shared storage the same way, regardless of which node the application is actively running on.

Once the shared storage is configured, you should at this point validate that each node can properly access the storage. You can best accomplish this by powering on one node at a time and validating connectivity to the storage. You should also configure a mount path or drive letter to the storage at this time as well. Each node must use the same mount path or drive letter to access the shared storage in order for the cluster to properly function.

With the shared storage ready to go, your final task will be to configure the network. Ideally, each node will have at least two network interfaces. One interface should be dedicated for the heartbeat network, and a second interface should be configured for client access and for failover of the heartbeat network. If the heartbeat crossover cable fails, for example, and the cluster isn't configured to use a second network for the heartbeat, each node may assume that the other node is down and attempt to take control of the virtual server. This can result in both nodes in a cluster trying to service the same virtual server and thus both writing to the same shared storage. This problem is known as *split brain* and can be avoided by having at least two potential networks for the cluster heartbeat.

With the network and storage set up, your final task is to install and configure the clustering service or application to be used to provide for failover clustering. Next, we'll cover configuring the Microsoft Cluster Service (MCS). Later, we'll show how to configure Linux failover clusters.

Setting Up Microsoft Server Clusters

Microsoft refers to its implementation of failover clustering as *server clustering*. With so many technologies in IT having so many different names, why not do the same with clustering? Well, at least that's how Microsoft sees it.

In the following sections, we'll cover some of the vocabulary and configuration that's unique to MCS and will then show you the steps for setting up a Windows server cluster. Finally, we'll wrap things up by covering the general management of Windows clusters.

Looking Under the Hood

With the foundations of clustering behind you, you're now ready to dive a little deeper into the service itself. The Cluster Service actually consists of several components, collectively working as one entity. At this point, you're probably thinking, "Why should I bother with the details? I just want to use the service!" If you like details, that's great. If not, read this anyway. The reason for being aware of the Cluster Service components isn't to help you now, necessarily, but to make your life easier when things go wrong. We'll provide you with brief descriptions of the core components of MCS.

Resources and Groups

Resources and groups are the principal elements of server clusters. Resources define objects controlled by the Cluster Service. Some examples of cluster resources are as follows:

- **IP address:** Provides IP address of virtual server
- **Network name:** Provides virtual server NetBIOS name
- **Physical disk:** Defines access to shared disk for use by a virtual server
- **Print spooler:** Defines print queues for network-attached printers

Notice that with the IP address and network name resources, we used the term *virtual server*. When you place an IP address resource and network name resource in the same group, you have a virtual server. As long as the resources are online, clients on the network can access shares and even ping the virtual server. Most of the groups you configure will be defined by their virtual server attributes.

Now let's get to groups. Resources that work in conjunction with one another are generally placed into groups. For example, when clustering SQL server, you place all SQL-related resources, including an IP Address and Network Name, in a single group. This will give you a SQL virtual server. Like resources are placed into a group because groups define failover. If a resource in a group fails, then the entire group is moved to another node, where the Cluster Service will attempt to restart all resources.

Tip Since failover occurs at the group level, a best practice is to configure a separate group for each virtual server. This way, the failure of one virtual server will not affect other virtual servers in the cluster.

Later when you learn about configuring server clusters, you'll see how groups and resources appear in the Cluster Administrator tool.

Checkpoint Manager

The Checkpoint Manager is the service component that performs registry *checkpointing*. Registry checkpointing is the process that allows the Cluster Service to verify that a cluster-unaware resource can successfully failover to another node. Cluster-unaware resources are typically application components that don't natively support clustering. You can use many cluster-unaware applications in a server cluster by configuring the Generic Service resource. We'll cover resource configuration shortly.

You've already read that configuration changes to the cluster are written to the quorum log. Since the cluster can contain several nodes, all being online or offline at different times, it needs a way to determine the updates written to the quorum log that are required by each node. This is the other purpose of the Checkpoint Manager. By logging checkpoints against the quorum disk, the Generic Service resource can easily update a node's registry configuration data as soon as it's brought online, thus keeping the configuration of all nodes in the cluster synchronized.

Communications Manager

The Communications Manager, also known as the Cluster Network Driver, is the service component that's responsible for managing node-to-node communications. The Communications Manager directs the flow of heartbeat and remote procedure call (RPC) communications and also ensures that each node in the cluster is notified when resources are brought online and go offline.

Configuration Database Manager

Most cluster configuration information is stored in the quorum log and registry checkpoint files on the quorum resource. In managing this information, the Configuration Database Manager ensures that any new node joining the cluster receives the most recent cluster configuration information.

Aside from managing information on the quorum resource, the Configuration Database Manager also stores a cluster configuration database in the registry on each cluster node. The configuration database stores information on the cluster, as well as all of its resources and groups.

Node Manager

Each group can have a preference list that outlines the order of nodes it prefers to run on. When a node fails, the Node Manager determines the node that will take ownership of the group based on the group's preference list.

Resource DLLs

Resource dynamic link libraries (DLLs) provide a means for the Cluster Service to communicate with other applications supported by the Cluster Service. Applications running on a cluster are categorized as either cluster-aware or cluster-unaware based on how they interact with the Cluster Service resource DLLs.

Cluster-aware applications use the Cluster application programming interface (API) to communicate with the Cluster Service and its related resources. Cluster-aware applications natively support failover. Cluster-unaware applications are those that aren't coded to interface with the Cluster API. To allow them to support failover, Microsoft provides a generic resource DLL that the applications can use, provided they meet the clustering requirements.

Resource Monitor

A *resource monitor* is a component that provides for communication between the Cluster Service and a resource DLL. By default, only a single resource monitor is run on each node, which allows the node to communicate with the Cluster Service using RPCs. The Cluster Service is designed so all cluster-aware applications can share a single resource monitor. Since each resource monitor runs as its own process, the flaw in this architecture is that any failure by a single process running inside the resource monitor can cause all processes running inside the monitor to freeze. To prevent this, some third-party cluster-aware applications use their own resource DLLs and are configured to run in their own resource monitor.

Other Service Components

The Cluster Service also has several other components worth mentioning. The following list describes the remaining components:

- **Event Processor:** Starts the Cluster Service on each node and allows event messages to be passed between cluster nodes
- **Event Log Manager:** Ensures that all cluster-related events are logged onto each cluster node
- **Failover Manager:** Determines ownership of groups when a failover occurs
- **Global Update Manager:** Propagates cluster configuration updates to all nodes in the cluster
- **Log Manager:** Writes cluster configuration changes to the quorum log on the quorum resource
- **Membership Manager:** Tracks all members (nodes) in a cluster, and updates each node when a node fails or comes back online
- **Object Manager:** Maintains a database of all cluster objects, including nodes, resources, and groups
- **Resource Manager:** Starts and stops resources and initiates group failovers

Before implementing a cluster, it's best to plan for how the cluster's resources and groups should be configured. You'll look at these issues next.

Planning Resource and Group Configuration

While it's a simple task, as you'll soon see, to create several resources and place them into groups, prior planning can make you look smart down the road when the inevitable failure occurs. You can break group configuration into seven phases:

1. Create a list of all server-based applications.
2. Sort the list by which applications do and don't support failover.
3. Verify licensing.
4. List all nonapplication resources where failover is desired.
5. Document dependencies for each resource.
6. Make preliminary group assignments.
7. Make final group assignments.

The following sections outline the considerations you must make during each phase.

Creating a List of All Server-Based Applications

In this phase, you document all applications running on each server that will participate in the server cluster. Your initial reaction probably would be to document the applications you plan on configuring for failover. The problem with this approach is that an undocumented application running on a cluster node can consume memory that you didn't consider during planning. Remember that each cluster node must have substantial physical resources to accommodate all applications that might failover onto it. If you have an application that isn't clustered, running on a node consuming 100MB of physical RAM, you could run into problems during a failover. When you scope out the physical hardware requirements of each cluster node, having each application documented alongside its approximate resource consumption will allow you to arrive at accurate estimates. While the application vendor can provide numbers on physical memory and storage usage, you'll most likely have to arrive at information such as CPU and bandwidth consumption through load tests.

Sorting the List

Once you have the list compiled of all applications, you can then sort the list to differentiate the applications that will use the MCS failover feature from all others. For any application to support failover, it must be able to do the following:

- Communicate using TCP/IP
- Allow you to specify where its data is stored

Some applications may support failover that don't require 100 percent availability. Your list should single out the applications that will be configured for failover from those that won't.

Verifying Licensing

Before configuring any application for clustering, you should verify its licensing requirements. An application running on a single server will most likely require one license. If that same application is configured to run on a two-node cluster, it might now need three licenses: one license for each node and one license for the virtual server. As far as the OS is concerned, you should also note the licensing currently employed. If per-seat client-access licensing is being used, then the licensing is valid for clients accessing applications on either node. If the application uses per-server or concurrent-use licensing, then you should ensure that ample licenses exist to handle the peak load placed on the application from the server's clients.

Tip Per-server licensing isn't pooled between cluster nodes and can't failover. Make sure you have ample licenses on each cluster node, whether it's an active or a passive node.

Listing All Nonapplication Resources Where Failover Is Desired

List the file shares and print spoolers you want to have failover capability. Remember that file access and printer access still consume resources and need to be accounted for in order to ensure accurate planning.

Documenting Dependencies for Each Resource

You must carefully outline relationships between resources. The required connections between resources will allow you to make proper grouping decisions. For example, if you're clustering an Exchange server, you want to make sure the Exchange services, network name, IP address, and shared disks are all grouped together. Exchange can't run if it has no place to store its data. This holds true for all other applications. When grouping resources, consider the following guidelines:

- Each resource and all its dependencies must be grouped together.
- Each resource can be in only one group.

Since each resource can exist in only one group, you may find that your total number of groups will be smaller than expected.

Note Physical disk resources are assigned by physical disk, not by disk partitions. This means that each shared physical disk can reside only in a single group.

Making Preliminary Group Assignments

With dependencies properly documented, your preliminary group assignments should already be partially complete. At this point, don't forget yourself as the administrator. You have to manage these groups. Managing a few large groups is always easier than managing several small groups. While you should try to avoid having too much in a single group, such as a file server, IIS server, and Exchange server, try not to get too granular with your grouping either.

A common example of a means to conserve grouping is to group resources by department, such as TrainingFileWeb, which could be a group encompassing file server and IIS resources for the department's intranet server.

Making Final Group Assignments

With preliminary group assignments in place, you should now work to name each group based on its intended purpose. For example, you'd want to name a DHCP server group as DHCPServer or name the group of resources for SQL server as SQLServer. While this may look like common sense to you now, we can't tell you how many groups we've seen named with the default values of Disk Group 1, Disk Group 2, or Disk Group 3. Names like Disk Group aren't helpful.

Another common practice with final grouping is to note resource dependencies within the group. Remember that if a resource fails, such as an IP address resource, everything that depends on that resource will fail as well. A common practice of visually grouping dependencies is to use a dependency tree. Figure 9-6 shows a sample tree. Notice that the print spooler resource has three dependencies. This means that failure of the physical disk resource will cause the print spooler to fail as well.

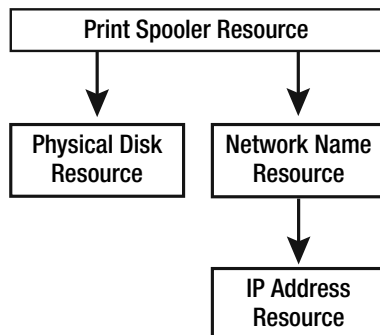


Figure 9-6. *Print spooler resource dependence tree*

Viewing dependency trees provides one more means of verifying the validity of your group configuration choices. Now that you've looked at how to logically put together a cluster, you'll see the actual steps involved.

Installing the Windows Server 2003 Cluster Service

In this section, you'll look at the general procedure for configuring the Windows Server 2003 Cluster Service in a two-node environment.

Prior to installing the Cluster Service, you must first satisfy several prerequisites. To begin, you must have the following:

- A configured and online domain controller.
- A cluster user account that's a member of the Domains Admins global group or is at a minimum set as a local administrator on each cluster node.
- Shared storage must be configured and online. Each node uses the same drive letters or mount paths to access the shared storage.
- The cluster public and heartbeat networks should be configured.

The Cluster Service can operate with a single NIC for both public and heartbeat traffic; however, two NICs are the recommended minimum. Once you've met the prerequisites, you're now ready to install the Cluster Service.

Caution Microsoft strongly recommends only one node be online during the installation and initial configuration of the first cluster node. Once the first node is configured in the server cluster, then power up each subsequent node and join it to the cluster.

To install the Cluster Service, follow these steps:

1. Power down all but one node in the cluster.
2. On Node1, click Start ► Administrative Tools ► Cluster Administrator.
3. When the Cluster Administrator opens, you should see the Open Connection to Cluster dialog box appear. In this dialog box, select Create New Cluster in the Action menu and click OK. If you don't see the Open Connection to Cluster dialog box, in Cluster Administrator, click File ► New ► Cluster.
4. The New Server Cluster Wizard should now open. Click Next.
5. Enter a name for the cluster in the Cluster Name field, and click Next.
6. In the Computer Name field, leave the default name selected, and click Next.
7. The Cluster Administrator will now analyze the cluster configuration. If all previous steps were completed successfully, you should see nothing but checkmarks in this window (see Figure 9-7). If a problem is encountered, the wizard will give you a general description of the problem. Once you correct the problem, click the Re-analyze button to perform the cluster analysis again. When the analysis completes successfully, click Next.

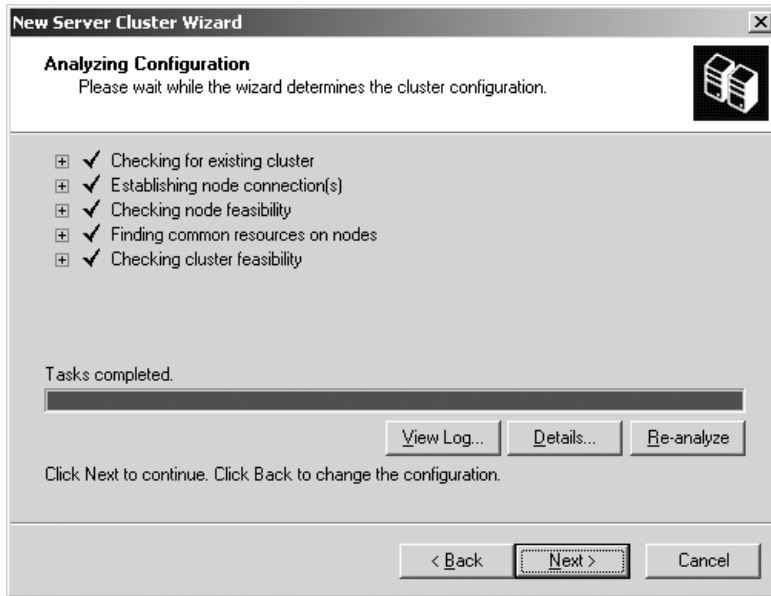


Figure 9-7. Successful cluster configuration analysis

8. In the IP Address field, enter an IP Address for the cluster. In our example, we used 192.168.0.20. Once an address is entered, click Next.
9. Now enter the username and password of the Cluster Service user account (created earlier), and click Next.
10. View the settings in the Proposed Cluster Configuration dialog box, and click Next to create the cluster. To change the location of the quorum disk, click the Quorum button.
11. This wizard will now create the cluster. When it finishes, click Next.
12. Click Finish to close the New Server Cluster Wizard window.

You should now see the cluster shown in the Cluster Administrator, with Node1 listed as the only cluster node. Perform these steps to add a node to the cluster:

1. Power up Node2. Once the node finishes booting, in the Cluster Administrator MMC on Node1, right-click the cluster object, select New, and then click Node.
2. When the Add Nodes Wizard opens, click Next.
3. You should now see the Select Computers window. In the Computer Name field, enter **Node2** (or whatever you set as the host name of Node2), and then click the Add button. Then click Next.
4. Once the wizard finishes analyzing the cluster configuration, click Next.

5. Enter the password for the Cluster Service user account, and click Next.
6. Verify that the cluster configuration settings are correct, and click Next.
7. The wizard will now add Node2 to the cluster. When it finishes, click Next.
8. Click Finish to close the Add Nodes Wizard.

You should now see both Node1 and Node2 displayed in the cluster's configuration. At this point, the cluster is now online and ready for anything you can throw at it. At this point, you can configure the cluster's resources and groups and install any cluster-aware applications.

Using the Cluster Administrator

You can perform nearly all administrative functions using the Cluster Administrator. You can run it locally on any cluster node or remotely on a Windows 2000, Windows XP, or Windows Server 2003 system. When you open the Cluster Administrator, you have the ability to decide which cluster you'd like to administer, allowing you to manage all clusters in your organization without having to leave your seat. While technology may make us smarter, it also makes us fatter.

Figure 9-8 shows the Cluster Administrator user interface. Notice in the example that two groups exist. One group is Cluster Group, which is the default group that's configured when the Cluster Service is installed. This group hosts the cluster quorum resource and is responsible for maintaining the cluster's configuration. The second group is Mail and hosts an Exchange 2003 virtual server. With the Mail group selected, you can see that the group contains 11 resources. The group has two physical disks, network name and IP address resources, and seven Exchange server resources.

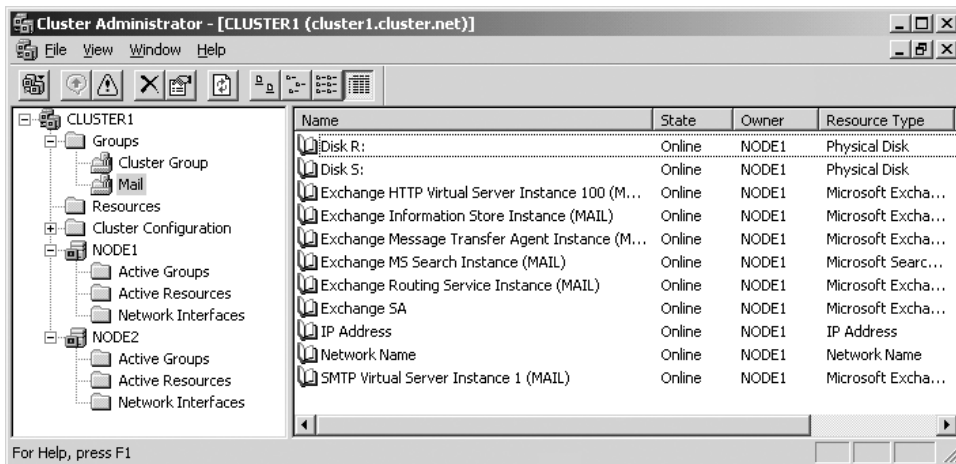


Figure 9-8. Cluster Administrator UI

The first object displayed in the interface is the cluster itself. Figure 9-8 shows Cluster1. Below the cluster are folders and placeholders for the cluster groups, resources, configuration, and each node.

The Groups folder lists the groups on the cluster, as well as the status of each group. For groups and resources, you'll see a down arrow next to an object that's offline and an *X* accompanying a failed object. This graphical representation allows you to quickly scan to see that your cluster is fully operational.

The next folder shown in the Cluster Administrator is the Cluster Configuration folder. This folder allows you to do the following:

- View a list of resources that can be installed, such as IP address, DHCP, local quorum, and so on
- View and configure networks used by the cluster
- View a listing and status of the network interfaces on each cluster node

Finally, the Cluster Administrator provides information on the operation of each cluster node. Under each node object, you can do the following:

- View the active groups running on the node
- View the active resources running on the node
- Check the status of each network interface on the node

The Cluster Administrator is simple to use, so we won't waste more of your time trying to explain the obvious and will get down to business. Next, we'll give you step-by-step procedures for performing some of the most common clustering administrative tasks.

Moving a Group

If you need to perform maintenance on a cluster node or simply want to test failover in the cluster, then you'll need to know how to move a group. To move a group, you simply right-click the group to be moved and then select Move Group. After doing this, you'll see each resource in the group taken offline, then moved to the next node in the cluster, and then brought back online. Once the move is complete, you'll see the name of the next node in the cluster listed as the owner of the group.

Adding Resources to a Group

When you need to create virtual servers, this will involve adding resources to a group. To have a virtual server, you need to start with a physical disk, an IP address, and the network name resources in a group. If a physical disk is in another group, you can move the disk resource into the new group by first taking the resource offline and then dragging and dropping it into the new group. Note that groups can be created by right-clicking the cluster object, selecting New, and then clicking Group.

Now to add resources to a group, follow these steps:

1. Right-click the group in which you want to add a resource, and click New and then Resource.
2. In the New Resource dialog box, enter a name and description for the resource, select the resource to be created in the Resource Type drop-down menu, and then click Next.
3. Now select the possible owners for the resource (by default all nodes are selected). The list of possible owners will determine the nodes to which the resource's group can failover. Once you've selected the owners, click Next.
4. If other resources must be online in order for the new resource to operate, then select the appropriate dependencies from the Available Resources field and move them to the Resource Dependencies field. When finished, click Next.
5. Finally, you'll need to enter the specific resource's parameters. This window will vary by resource type. For example, for a new IP address resource, you'd have to enter an IP address, subnet mask, and network interface for the resource. Once you've entered the resource parameters, click Finish to create the resource.
6. When the resource is created, it will appear in the group in an offline state. To bring the resource online, right-click the resource and select Bring Online.

As you've seen, navigating the Cluster Administrator tool is relatively simple. The tough part with clustering is always bringing the shared storage online and ensuring that it can be seen the same way by each cluster node. With Microsoft clustering out of the way, next you'll turn your attention to Linux clustering.

Tip For more information on Microsoft clustering, point your Web browser to <http://www.microsoft.com/windowsserver2003/technologies/clustering/default.aspx> to access the Windows 2003 clustering home page. This site is loaded with clustering white papers and how-to articles.

Setting Up Linux Failover Clusters

To set up failover clusters on Linux operating systems, you can go in three different directions: purchase a clustering solution from a third-party vendor, purchase a clustering solution by an operating system vendor, or use an open-source solution available on the Internet.

In the next sections, we'll cover failover clustering using the Red Hat Cluster Suite and also show how to configure clustering using the leading open-source solution from the Linux High Availability Project.

Setting Up the Red Hat Cluster Suite

For enterprise-class Linux clusters, many organizations have turned to Red Hat. With the Red Hat Cluster Suite, organizations can deploy up to an eight-node failover cluster. Best of all, this solution is fully supported and documented by Red Hat. While you can get some open-source failover cluster products for free, many prefer the peace of mind that comes with using a purchased and supported software product. Unlike with Red Hat Advanced Server 2.1, newer Red Hat OSs don't come with the cluster software for free. Instead you'll have to buy it as a separate add-on from Red Hat.

The Red Hat Cluster Suite offers pretty much everything you'd want in a failover cluster, including the following:

- SCSI and Fibre Channel shared storage support
- Simple configuration and administration
- Shared cluster quorum
- Support for all major Linux applications
- Support for failover domains

Let's examine these features in more detail.

SCSI and Fibre Channel Shared Storage Support

The Red Hat Cluster Suite supports both SCSI and Fibre Channel shared storage. This allows you to configure a simple two-node cluster with a shared SCSI-attached storage array or scale up to eight nodes by connecting to shared storage through a SAN. Also, with the SAN integration support, it's easy to connect your Red Hat cluster to your existing storage infrastructure.

The Red Hat Cluster Suite also supports Global File System (GFS), which provides for better integration with storage networks. GFS supports simultaneous reads and writes to a single shared file system in a SAN. For more on GFS, turn to Chapter 12.

Simple Configuration and Administration

Red Hat clusters can be quickly deployed and configured on Red Hat Enterprise Advanced Server systems using the Cluster Configuration tool, which is shown in Figure 9-9.

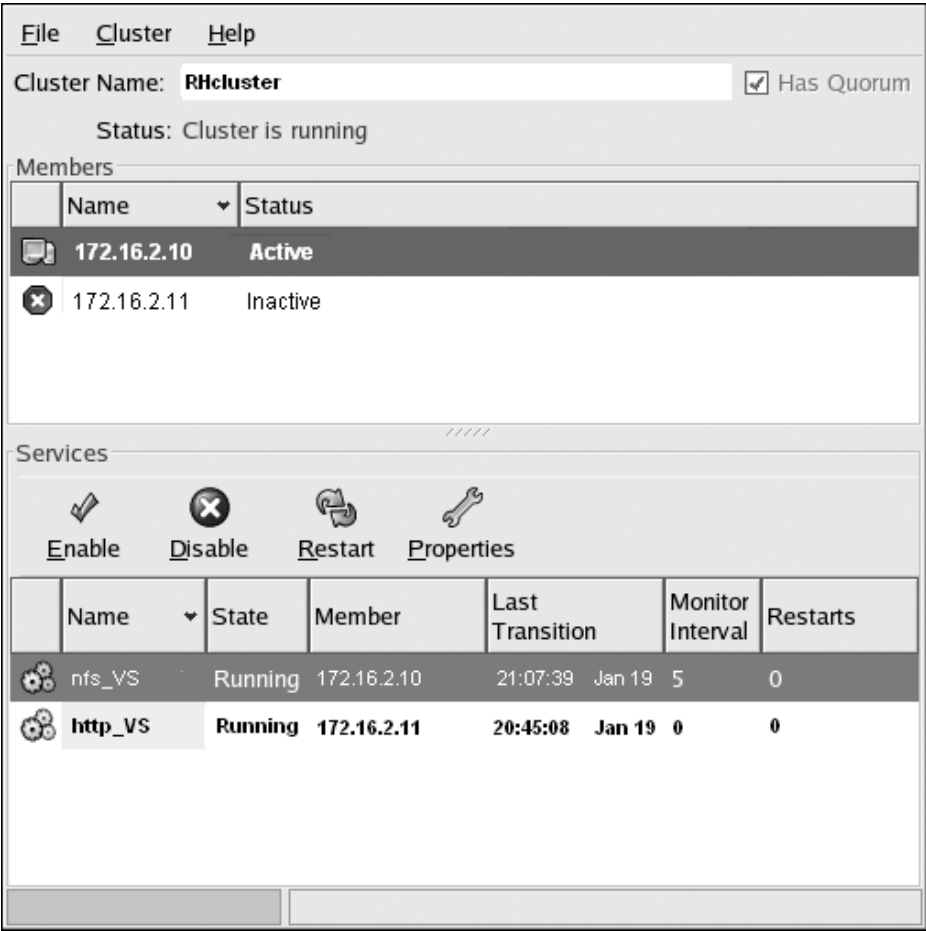


Figure 9-9. Red Hat Cluster Configuration tool

While we realize that many Linux enthusiasts scoff at the notion of GUIs, the bottom line for many businesses is cost of ownership, and if a GUI tool leads to more reliable and faster deployment of failover clusters, then so be it. In fact, Red Hat supports changes to its cluster .xml configuration files only through the Cluster Manager. If you want to make changes via a text editor, you're on your own! Who would have thought this came from Red Hat five years ago?

Shared Cluster Quorum

As with Microsoft clusters, Red Hat failover clusters use a shared quorum drive to maintain updates to cluster configuration information. This way, if changes are made to the cluster

while one node is offline, the offline node will receive the updates as soon as it boots back up. The availability of the shared quorum disk provides for much more reliable and easier-to-manage clusters.

Supports All Major Linux Applications

The Red Hat Cluster Suite fully supports applications and services such as Oracle, Apache, NFS, and MySQL. In fact, Red Hat even offers complete step-by-step procedures on how to configure these applications and services to run on your Red Hat Linux cluster. This level of support, as well as extensive documentation, for many makes the Red Hat Cluster Suite well worth the price.

Support for Failover Domains

Failover domains act as containers for clustered services. Architecturally, they're similar to groups with Windows clusters. When you configure "restricted" failover domains, you can assign ownership of a failover domain to specific nodes in the cluster. This gives you control of how services failover in the event of a failure. Without configured failover domains, services would arbitrarily failover to any available node in the cluster. This feature will give you much greater control of how services and service failures are managed by the cluster.

As you can see, the Red Hat Cluster Suite is a robust clustering platform, with quite a bit to offer. If you're looking for more information on Red Hat clustering, refer to the `RH_Clustering.pdf` document located in the `RedHat` folder on the companion CD or point your Web browser to the Red Hat Clustering home page at <http://www.redhat.com/software/rha/cluster>.

Using Linux-HA Clusters

Linux High Availability (Linux-HA) clustering is the result of the Linux High Availability Project (<http://www.linux-ha.org>). This project started as the standard for two-node Linux failover clusters, and at the time of publication is heading toward completing code to support up to 16-node or higher cluster configurations.

Linux-HA clusters can be configured on nearly any Linux distribution. Since we don't know which Linux distribution you have, in this section we'll show how to configure a two-node cluster using open-source software from <http://www.ultramoney.org>. UltraMonkey provides the software and documentation of Linux-HA on Red Hat distributions. Since this particular cluster format can operate with or without using shared storage, you don't have to do any preparation to get each cluster node to properly see the shared storage in order to install and configure the cluster service. If you want to synchronize data between local storage on each cluster node, you can use tools such as `Mirrordir`, which you can download from <http://www.rpmsfind.net>. Another tool that provides for remote replication of file data is `Rsync`, which is available at <http://rsync.samba.org>.

Note For sharing disk resources in a Linux-HA failover cluster, you can configure GFS on the cluster nodes. GFS supports shared SCSI, iSCSI, and Fibre Channel storage. For more information on GFS, point your Web browser to the Open GFS home page at <http://opengfs.sourceforge.net>.

The key to operating Linux-HA clusters is Heartbeat. Heartbeat is the monitoring service that will allow one node to monitor the state of another and assume control of the cluster's virtual IP address if the primary node fails. This setup will allow you to configure a simple active-passive failover cluster. In the next section, we'll show how to configure a sample UltraMonkey implementation of Linux-HA, which was installed on Red Hat Enterprise Advanced Server 3.0. Figure 9-10 shows our setup.

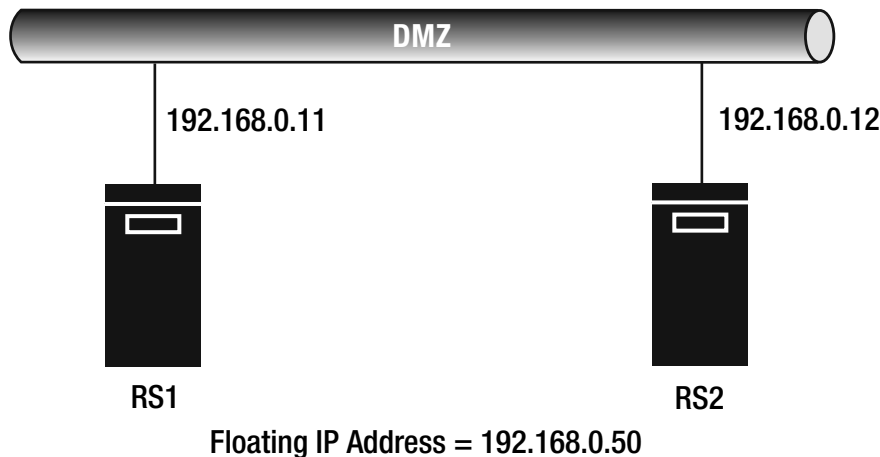


Figure 9-10. *Linux-HA two-node failover cluster*

This configuration could allow for the failover of Web servers, such as providing redundancy and fault tolerance for an Apache HTTP server.

Initial Configuration

Before installing the high-availability software, you must first configure each node's host name and TCP/IP settings. With both hosts configured, you'll then need to connect them to the Internet to download the UltraMonkey software from <http://www.ultramoney.org>. You'll find a single folder available on the Web site to download all the Red Hat installation RPMs. Download each available RPM for your Red Hat distribution before continuing.

Once you have the available software, you're ready to set up the cluster. The first step in the installation process is to upgrade `mkinitrd`. To do this, follow these steps:

1. Assuming you're logged on and you've downloaded the RPMs as root, change to the `/root` directory.
2. Next, use RPM to install the latest `mkinitrd` package. For example, use `rpm -hFv mkinitrd-3.5.13-1.um.1.i386.rpm`.

With `mkinitrd` upgraded, the next step is to upgrade the kernel. To do this, download the kernel RPM from <http://www.ultramonkey.org> that's appropriate for each node's hardware architecture and then use the RPM to install the kernel upgrade. Here are examples for different host architectures:

- **AMD Athlon:** `rpm -Fhv kernel-2.4.21-20.EL.um.1.athlon.rpm`
- **i686:** `rpm -Fhv kernel-2.4.21-20.EL.um.1.i686.rpm`
- **i586:** `rpm -Fhv kernel-2.4.21-20.EL.um.1.i586.rpm`

With the kernel upgraded, you'll need to reboot each node for the changes to be applied, but prior to doing so you can use the RPM to install the remaining packages to configure Linux-HA.

Note The remaining packages to be installed depend on the following packages: `perl-Digest-HMAC`, `perl-Digest-SHA1`, and `perl-Parse-RecDescent`. You can download these packages from <http://www.rpmfind.net> and then install them using the `rpm -Uvh` command.

You can now install the remaining configuration and management packages that were downloaded from UltraMonkey. These packages include the following:

- `heartbeat-1.0.4-2.rh.el.um.1.i386.rpm`
- `heartbeat-ldirectord-1.0.4-2.rh.el.um.1.i386.rpm`
- `heartbeat-pils-1.0.4-2.rh.el.um.1.i386.rpm`
- `heartbeat-stonith-1.0.4-2.rh.el.um.1.i386.rpm`
- `ipvsadm-1.21-1.rh.el.1.i386.rpm`
- `libnet-1.1.0-1.rh.el.1.i386.rpm`
- `perl-Authen-SASL-2.03-1.rh.el.um.1.noarch.rpm`
- `perl-Convert-ASN1-0.16-2.rh.el.um.1.noarch.rpm`
- `perl-IO-Socket-SSL-0.92-1.rh.el.um.1.noarch.rpm`
- `perl-Mail-IMAPClient-2.2.7-1.rh.el.um.1.noarch.rpm`
- `perl-Net-SSLeay-1.23-1.rh.el.um.1.i386.rpm`
- `perl-XML-NamespaceSupport-1.08-1.rh.el.um.1.noarch.rpm`
- `perl-XML-SAX-0.12-1.rh.el.um.1.noarch.rpm`
- `perl-ldap-0.2701-1.rh.el.um.1.noarch.rpm`

You can install all the previously listed RPMs collectively using a single `rpm -Uvh` command by including each package as a parameter in the command syntax. Note that the package versions listed were the most recent at the time of this book's publication and may have different version numbers by the time you download them.

Once all the packages are installed, run the `reboot` command to reboot each node. After both nodes reboot, you'll need to copy the following files from `/usr/share/doc/heartbeat-1.0.4` to the `/etc/ha.d` folder:

- `ha.cf`
- `haresources`
- `authkeys`

With the files in place, you'll now need to configure them. Let's begin with `authkeys`.

Note When you edit these three files, each line will be commented out with a `#`, so editing the files is merely an exercise in removing a few `#` statements and substituting the file's sample parameters with settings particular to your specific cluster.

authkeys File Configuration

You must configure `authkeys` to set the heartbeat authentication protocol and password for each node. This file should be identical on both nodes. When you configure the file, you need to set the `authm` parameter as well as the authentication method and password. In the sample `authkeys` file shown in Listing 9-1, the most secure authentication method (SHA-1) was selected, with a password of `P@ssword!`. We left in the file's original comments, as they also provide additional information on the configuration and content of the file.

Listing 9-1. *authkeys*

```
#
#   Authentication file.  Must be mode 600
#
#
#   Must have exactly one auth directive at the front.
#   auth    send authentication using this method-id
#
#   Then, list the method and key that go with that method-id
#
#   Available methods: crc sha1, md5.  Crc doesn't need/want a key.
#
#   You normally only have one authentication method-id listed in this file
#
#   Put more than one to make a smooth transition when changing auth
#   methods and/or keys.
```

```
#
#
#     sha1 is believed to be the "best", md5 next best.
#
#     crc adds no security, except from packet corruption.
#         Use only on physically secure networks.
#
auth 2
#1 crc
2 sha1 P@ssword!
#3 md5 Hello!
```

Once `authkeys` is edited, your next task is to change it to mode 600. To do this, run the command `chmod 600 /etc/ha.d/authkeys`. With `authkeys` configured, you can then move onto configuring the `ha.cf` file.

ha.cf File Configuration

The `ha.cf` file establishes the names of the nodes in the cluster, as well the configuration of the heartbeat packets. Listing 9-2 shows the content of our configuration file. In our sample, we used the interface `e0` for the heartbeat with nodes named `RS1` and `RS2`.

Listing 9-2. *ha.cf*#

```
#     There are lots of options in this file.  All you have to have is a set
#     of nodes listed {"node ...}
#     and one of {serial, bcast, mcast, or ucast}
#
#     ATTENTION: As the configuration file is read line by line,
#                 THE ORDER OF DIRECTIVE MATTERS!
#
#                 In particular, make sure that the timings and udpport
#                 et al are set before the heartbeat media are defined!
#                 All will be fine if you keep them ordered as in this
#                 example.
#
#
#     Note on logging:
#     If any of debugfile, logfile, and logfacility are defined then they
#     will be used. If debugfile and/or logfile are not defined and
#     logfacility is defined then the respective logging and debug
#     messages will be logged to syslog. If logfacility is not defined
#     then debugfile and logfile will be used to log messages. If
#     logfacility is not defined and debugfile and/or logfile are not
#     defined then defaults will be used for debugfile and logfile as
#     required and messages will be sent there.
#
#     File to write debug messages to
```

```
#debugfile /var/log/ha-debug
#
#
#       File to write other messages to
#
logfile      /var/log/ha-log
#
#
#       Facility to use for syslog()/logger
#
logfacility   local0
#
#
#       A note on specifying "how long" times below...
#
#       The default time unit is seconds
#           10 means ten seconds
#
#       You can also specify them in milliseconds
#           1500ms means 1.5 seconds
#
#
#       keepalive: how long between heartbeats?
#
keepalive 2
#
#       deadtime: how long-to-declare-host-dead?
#
deadtime 30
#
#       warntime: how long before issuing "late heartbeat" warning?
#       See the FAQ for how to use warntime to tune deadtime.
#
warntime 10
#
#
#       Very first dead time (initdead)
#
#       On some machines/OSes, etc. the network takes a while to come up
#       and start working right after you've been rebooted. As a result
#       we have a separate dead time for when things first come up.
#       It should be at least twice the normal dead time.
#
initdead 120
#
#
#       nice_failback: determines whether a resource will
```



```

#      automatically failback to its "primary" node, or remain
#      on whatever node is serving it until that node fails.
#
#      The default is "off", which means that it WILL fail
#      back to the node which is declared as primary in haresources
#
#      "on" means that resources only move to new nodes when
#      the nodes they are served on die. This is deemed as a
#      "nice" behavior (unless you want to do active-active).
#
nice_failback on
#
#      hopfudge maximum hop count minus number of nodes in config
#hopfudge 1
#
#
#      Baud rate for serial ports...
#      (must precede "serial" directives)
#
#baud    19200
#
#      serial    serialportname ...
#serial /dev/ttyS0      # Linux
#serial /dev/cuaa0      # FreeBSD
#serial /dev/cua/a      # Solaris
#
#      What UDP port to use for communication?
#      [used by bcast and ucast]
#
udpport      694
#
#      What interfaces to broadcast heartbeats over?
#
bcast eth0      # Linux
#bcast eth1 eth2  # Linux
#bcast le0      # Solaris
#bcast le1 le2   # Solaris
#
#      Set up a multicast heartbeat medium
#      mcast [dev] [mcast group] [port] [ttl] [loop]
#      [dev]          device to send/rcv heartbeats on
#      [mcast group]  multicast group to join (class D multicast address
#                      224.0.0.0 - 239.255.255.255)
#      [port]         udp port to sendto/rcvfrom (no reason to differ
#                      from the port used for broadcast heartbeats)
#      [ttl]          the ttl value for outbound heartbeats. This affects
#                      how far the multicast packet will propagate. (1-255)

```

```

#      [loop]          toggles loopback for outbound multicast heartbeats.
#                      if enabled, an outbound packet will be looped back and
#                      received by the interface it was sent on. (0 or 1)
#                      This field should always be set to 0.
#
#
mcast eth0 225.0.0.1 694 1 0
#
#      Set up a unicast / udp heartbeat medium
#      ucast [dev] [peer-ip-addr]
#
#      [dev]           device to send/rcv heartbeats on
#      [peer-ip-addr]  IP address of peer to send packets to
#
#ucast eth0 192.168.1.2
#
#
#      Watchdog is the watchdog timer.  If our own heart doesn't beat for
#      a minute, then our machine will reboot.
#
#watchdog /dev/watchdog
#
#      "Legacy" STONITH support
#      Using this directive assumes that there is one stonith
#      device in the cluster.  Parameters to this device are
#      read from a configuration file. The format of this line is:
#
#      stonith <stonith_type> <configfile>
#
#      NOTE: it is up to you to maintain this file on each node in the
#      cluster!
#
#stonith baytech /etc/ha.d/conf/stonith.baytech
#
#      STONITH support
#      You can configure multiple stonith devices using this directive.
#      The format of the line is:
#      stonith_host <hostfrom> <stonith_type> <params...>
#      <hostfrom> is the machine the stonith device is attached
#      to or * to mean it is accessible from any host.
#      <stonith_type> is the type of stonith device (a list of
#      supported drives is in /usr/lib/stonith.)
#      <params...> are driver specific parameters. To see the
#      format for a particular device, run:
#      stonith -l -t <stonith_type>
#
#

```

```
#      Note that if you put your stonith device access information in
#      here, and you make this file publicly readable, you're asking
#      for a denial of service attack ;-)
#
#
#stonith_host *      baytech 10.0.0.3 mylogin mysecretpassword
#stonith_host ken3   rps10 /dev/ttyS1 kathy 0
#stonith_host kathy  rps10 /dev/ttyS1 ken3 0
#
#      Tell what machines are in the cluster
#      node      nodename ...      -- must match uname -n
node    RS1
node    RS2
#
#      Less common options...
#
#      Treats 10.10.10.254 as a psuedo-cluster-member
#
#ping 10.10.10.254
#
#      Started and stopped with heartbeat.  Restarted unless it exits
#                                     with rc=100
#
#respawn userid /path/name/to/run
```

At this point, you're almost there. The last step is to configure haresources.

haresources File Configuration

The haresources file specifies the services or applications that failover from node to node when the heartbeat isn't detected. This file is crucial, as it will determine what services will automatically start on the passive node when the passive node detects that the active node is no longer available. This file will also specify the virtual IP address of the virtual server that's hosted by the cluster. While the file's comments offer several examples, ultimately you'll probably just need to configure a single line of the file to allow for the proper failover of a service. In our example, this was the line that was added to the haresources file:

```
RS1      192.168.0.50 httpd
```

Adding this line to the file accomplishes the following tasks:

- Specifies that RS1 is the default active node
- Configures 192.168.0.50 as the IP address of the virtual server
- Sets the HTTP service to start after a failover

Listing 9-3 shows the complete haresources file, along with all default comments.

Listing 9-3. *haresources*

```
#
#      This is a list of resources that move from machine to machine as
#      nodes go down and come up in the cluster. Do not include
#      "administrative" or fixed IP addresses in this file.
#
# <VERY IMPORTANT NOTE>
#      The haresources files MUST BE IDENTICAL on all nodes of the cluster.
#
#      The node names listed in front of the resource group information
#      is the name of the preferred node to run the service. It is
#      not necessarily the name of the current machine. If you are running
#      nice_failback OFF then these services will be started
#      up on the preferred nodes - any time they're up.
#
#      If you are running with nice_failback ON, then the node information
#      will be used in the case of a simultaneous start-up.
#
#      BUT FOR ALL OF THESE CASES, the haresources files MUST BE IDENTICAL.
#      If your files are different then almost certainly something
#      won't work right.
# </VERY IMPORTANT NOTE>
#
#
#      We refer to this file when we're coming up, and when a machine is being
#      taken over after going down.
#
#      You need to make this right for your installation, then install it in
#      /etc/ha.d
#
#      Each logical line in the file constitutes a "resource group".
#      A resource group is a list of resources that move together from
#      one node to another - in the order listed. It is assumed that there
#      is no relationship between different resource groups. These
#      resource in a resource group are started left-to-right, and stopped
#      right-to-left. Long lists of resources can be continued from line
#      to line by ending the lines with backslashes ("\").
#
#      These resources in this file are either IP addresses, or the name
#      of scripts to run to "start" or "stop" the given resource.
#
```

```
#      The format is like this:
#
#node-name resource1 resource2 ... resourceN
#
#
#      If the resource name contains an :: in the middle of it, the
#      part after the :: is passed to the resource script as an argument.
#      Multiple arguments are separated by the :: delimiter
#
#      In the case of IP addresses, the resource script name IPAddr is
#      implied.
#
#      For example, the IP address 135.9.8.7 could also be represented
#      as IPAddr::135.9.8.7
#
#      THIS IS IMPORTANT!!
#
#      The given IP address is directed to an interface that has a route
#      to the given address. This means you have to have a net route
#      set up outside of the High-Availability structure. We don't set it
#      up here -- we key off of it.
#
#      The broadcast address for the IP alias that is created to support
#      an IP address defaults to the highest address on the subnet.
#
#      The netmask for the IP alias that is created defaults to the same
#      netmask as the route that it selected in the step above.
#
#      The base interface for the IPalias that is created defaults to the
#      same netmask as the route that it selected in the step above.
#
#      If you want to specify that this IP address is to be brought up
#      on a subnet with a netmask of 255.255.255.0, you would specify
#      this as IPAddr::135.9.8.7/24 .
#
#      If you wished to tell it that the broadcast address for this subnet
#      was 135.9.8.210, then you would specify that this way:
#          IPAddr::135.9.8.7/24/135.9.8.210
#
#      If you wished to tell it that the interface to add the address to
#      is eth0, then you would need to specify it this way:
#          IPAddr::135.9.8.7/24/eth0
#
```

```

#      And this way to specify both the broadcast address and the
#      interface:
#      IPaddr::135.9.8.7/24/eth0/135.9.8.210
#
#      The IP addresses you list in this file are called "service" addresses,
#      since they're the publicly advertised addresses that clients
#      use to get at highly available services.
#
#      For a hot/standby (non load-sharing) 2-node system with only
#      a single service address,
#      you will probably only put one system name and one IP address in here.
#      The name you give the address to is the name of the default "hot"
#      system.
#
#      Where the nodename is the name of the node which "normally" owns the
#      resource. If this machine is up, it will always have the resource
#      it is shown as owning.
#
#      The string you put in for nodename must match the uname -n name
#      of your machine. Depending on how you have it administered, it could
#      be a short name or a FQDN.
#
#-----
#
#      Simple case: One service address, default subnet and netmask
#      No servers that go up and down with the IP address
#
#just.linux-ha.org      135.9.216.110
#
#-----
#
#      Assuming the administrative addresses are on the same subnet...
#      A little more complex case: One service address, default subnet
#      and netmask, and you want to start and stop http when you get
#      the IP address...
#
#just.linux-ha.org      135.9.216.110 http
#
#-----
#
#      A little more complex case: Three service addresses, default subnet
#      and netmask, and you want to start and stop http when you get
#      the IP address...
#
#just.linux-ha.org      135.9.216.110 135.9.215.111 135.9.216.112 httpd
#
#-----
#

```

```
#      One service address, with the subnet, interface and bcast addr
#      explicitly defined.
#
#just.linux-ha.org      135.9.216.3/28/eth0/135.9.216.12 httpd
#
#-----
#
#      An example where a shared file system is to be used.
#      Note that multiple arguments are passed to this script using
#      the delimiter ':' to separate each argument.
#
#node1 10.0.0.170 Filesystem::/dev/sda1::data1::ext2
#
#      Regarding the node-names in this file:
#
#      They must match the names of the nodes listed in ha.cf, which in turn
#      must match the `uname -n` of some node in the cluster. So they aren't
#      virtual in any sense of the word.
#
RS1      192.168.0.50 httpd
```

Once this file is configured, you're now ready to start the service.

Starting the Heartbeat Service

With the heartbeat parameters configured, you run the following command to start the service on each node: `/etc/init.d/heartbeat start`. To test failover, run `/etc/init.d/heartbeat stop` on the active cluster node. You can also disable the network service on the active node. In a few moments the passive node will assume the IP address of the cluster virtual server and will start the HTTP service. At this point, you have a working Linux failover cluster!

Summary

As you've seen, failover clustering gives you the ability to run a virtual server on top of another system. If the physical system hosting the virtual server fails, another system in the cluster will assume control of the virtual server. For clients, this provides a high level of reliability to crucial data. Now you have a server that isn't dependent on any one physical system to run.

With solutions available on both Windows and Linux and plenty of industry pressure to offer more failover clustering solutions, it's likely that clusters will become easier to deploy and much more common in the coming years. Another form of clustering, load-balanced clustering, provides a different level of virtualization than with failover clustering. With load-balanced clustering, a virtual server can be hosted by up to 32 nodes simultaneously, providing for a way to handle a high volume of traffic that would normally be impossible for a single server to handle. You'll look at this form of virtualization next in Chapter 10.



Creating Load-Balanced Clusters

Midsize to enterprise-class organizations employ *load-balancing clusters* to do just what their name entails—balance a load. In case you're thinking “what?” right about now, let's take a more detailed look at load-balanced clusters. With load balancing, several physical computers collectively act as one or more logical computers for the purpose of distributing client requests amongst multiple physical servers. In addition to distributing client requests, or a server's load, the load-balancing cluster also provides for fault tolerance of data access.

To further understand the role of load-balancing clusters, consider the following load-balanced cluster implementations:

- Web servers
- FTP servers
- Streaming media servers
- VPN servers
- Terminal servers

Each of the previously mentioned load-balanced cluster implementations has the following similarities:

- Clients require read-only data access.
- One server may not be able to handle the load presented by the client access.
- The availability of access is crucial.

When clients require read and write data access, such as with access to an e-mail server, then failover clustering is typically implemented. Chapter 9 described this clustering configuration. Aside from the type of data access load-balanced clusters provide, the way they allow access is unique as well. With load-balanced clusters, you could have 30 servers in the cluster all providing access to data simultaneously. With the typical failover cluster, only a single physical server will serve as the data access point for the virtual server running on the cluster. To provide for data access across multiple physical servers, load-balanced clustering requires that each server, or *node*, in the cluster maintain its own local copy of the shared cluster data.

Figure 10-1 shows a typical load-balanced cluster. Note that the figure shows several physical computers acting as a single logical computer for the Web site `http://www.apress.com`. Each computer in the cluster will communicate with each other to decide which one will handle each specific incoming client request. When several client requests come to the cluster at the same time, all nodes will assume a portion of the load. So, if the load-balanced cluster has three computers, and nine client HTTP requests come to the cluster, each node handles three requests.

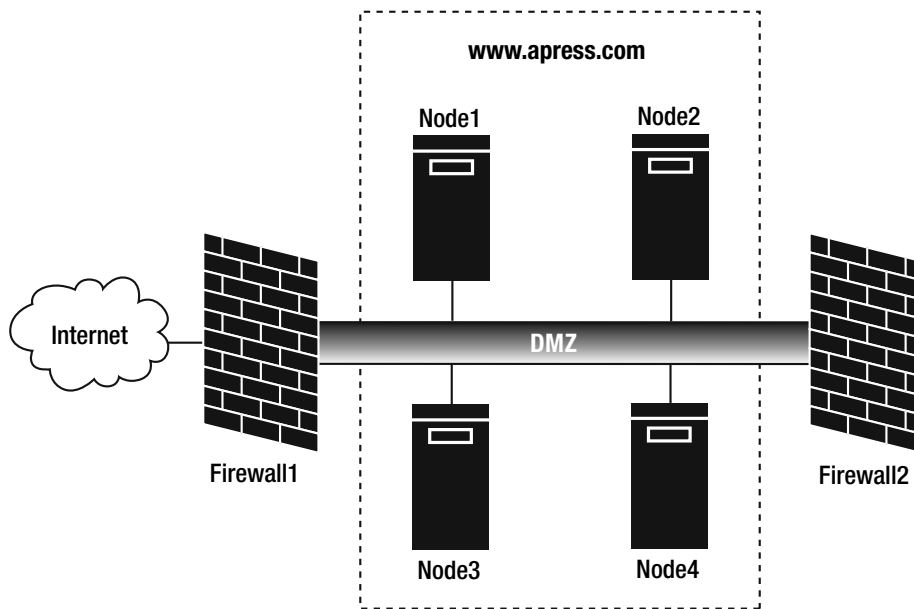


Figure 10-1. *Four-node load-balanced cluster*

In the example shown in Figure 10-1, if you were to connect to `http://www.apress.com`, you wouldn't be directed to a single server whose FQDN is `http://www.apress.com`. Instead, although you logically connect to `http://www.apress.com`, you're actually transparently redirected to one of the many servers that handle requests for the `http://www.apress.com` name. In one instance, you may connect to the first node in the cluster. The next time you connect, the third node in the cluster could answer your request.

One other feature that has added to the growing popularity of load-balanced clusters is their ability to scale. When planning a load-balanced cluster, your initial estimates don't have to be perfect with regard to the number of servers you'll need to effectively balance a load. As the load placed on the cluster grows, you can successfully balance the increased load by adding nodes.

To truly appreciate the necessity of load balancing, in the next section you'll examine an earlier technology that provided a form of static load balancing: round-robin DNS.

Round-Robin DNS: The Beginning

Other techniques allow you to distribute client requests to multiple servers. One of the most popular early techniques still employed in some instances today is *round-robin DNS*. The primary difference between round-robin DNS and load-balanced clustering is how they actually load balance. With load-balanced clusters, the servers in the cluster that are running communicate with each other to determine who will respond to client requests; thus, the term *dynamic* is often used to describe their operation.

Round-robin DNS is considered to be a static means of load balancing. With round-robin DNS, a DNS server maintains a list of several IP addresses for a single host name. For each client request, the DNS server responds with a different IP address, alternating through its list of possible addresses. Figure 10-2 shows the concept of this operation.

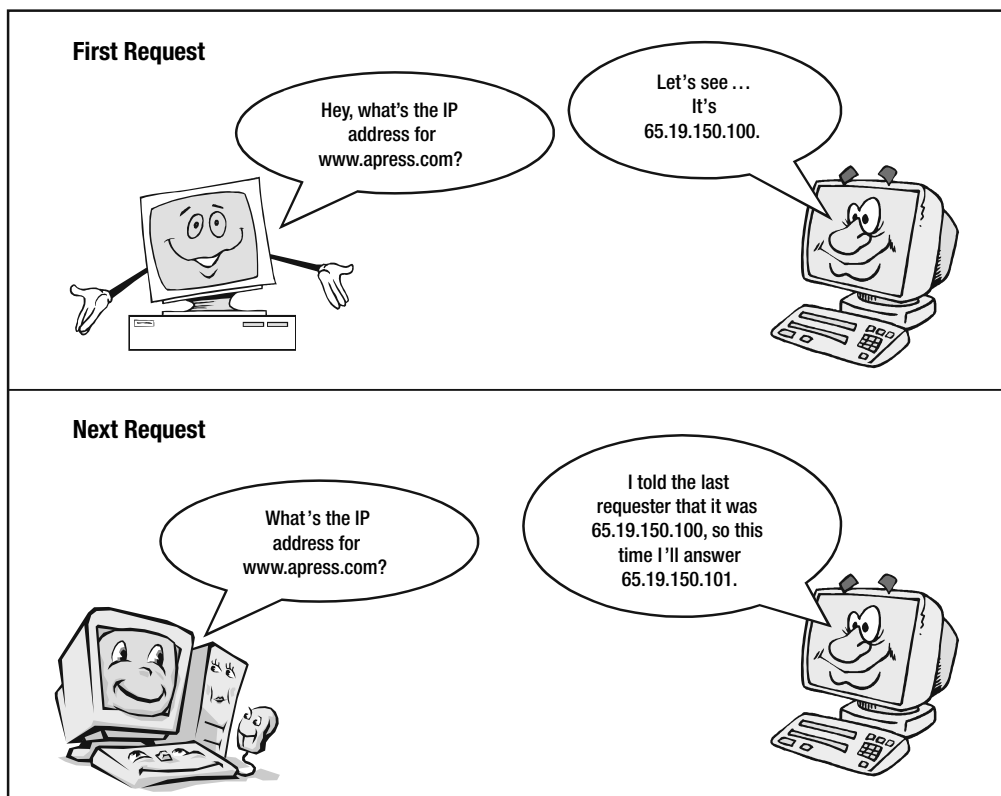


Figure 10-2. Round-robin DNS operation

Figure 10-3 shows the DNS configuration, with several IP addresses for the host name `www`.

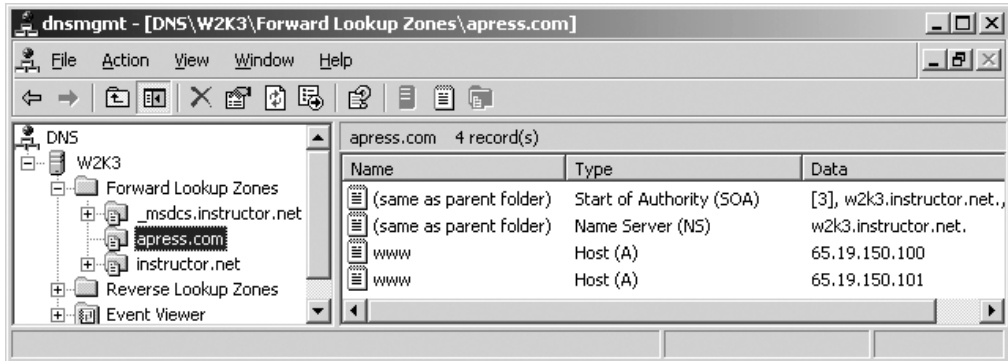


Figure 10-3. Round-robin DNS configuration

Although round-robin DNS is a means of load-balancing client requests over multiple servers, it has several flaws:

- If a server with an IP address that the round-robin DNS resolves to is down, clients will still be sent to the down server each time the DNS server responds with that address.
- IP address resolutions are cached by both clients and other DNS servers, meaning that once a client or a DNS server receives a host's IP address from the round-robin DNS server, it will continue to resolve all further requests to the same IP address for the life of the cache, thus diminishing the load-balancing effect of round-robin DNS.

Note Round-robin DNS is still used today as a means to statically load balance between two or more identical clusters. This is especially useful if the load-balanced cluster needs to scale beyond its maximum supported number of nodes. For example, a Windows Server 2003 load-balanced cluster supports a maximum of 32 nodes. If you need more nodes, you could configure two clusters (with 20 nodes each, for example) and use round-robin DNS to alternate client requests between the two clusters.

With this basic understanding of load-balanced clustering, you'll now learn about the common planning issues involved with deploying a load-balanced cluster. The chapter will then provide guidance on how to deploy two of the most popular load-balanced cluster configurations: Windows network load-balanced (NLB) clusters and Linux virtual server (LVS) load-balancing clusters.

Planning for Load-Balanced Clusters

Although having the flexibility to add nodes to the cluster gives you a safety net to work with, proper planning is still important. Load-balanced cluster planning has five key phases:

- Selecting applications
- Verifying licensing
- Analyzing risks
- Estimating server capacity
- Making technology decisions

Selecting Applications

Load balancing is generally application friendly, requiring little for an application to participate in the cluster. For an application to be able to run on a load-balanced cluster, it must do the following:

- It must use TCP/IP as its default network protocol.
- It must utilize a specific TCP or UDP port for communications.
- Several instances of the application must be able to run on multiple servers.

If the application running on the load-balanced cluster needs to share data with other nodes, the application needs to be able to synchronize the reads and writes to the data. Load-balanced clusters are ideal for read-only data access, typically running at the front end of your Web presence. When access is needed to shared data, the application running on the cluster node will typically call or write the data to a backend database or file share.

Tip Generally, any application running on a load-balanced cluster should be configured for read or possibly also read and execute access. To avoid data synchronization problems, load-balanced clustered applications should store shared data in a common repository, such as a backend database.

Verifying Licensing

As with planning failover clusters, you must verify that you have enough licenses to support the applications you plan to cluster. Although users may be accessing the application using a single virtual server name, odds are that a license will be required for each server on which the application is running. In addition, you may also need a license for each virtual server instance. Check with the application vendor to verify the specific requirements. Verifying licensing with the source is always the safest bet.

Analyzing Risks

In Chapter 9 you learned the fundamentals of risk assessment, seeing the numerous single points of failure that exist on any network. While the same principles apply to load-balanced cluster planning, you should also avoid falling into the following traps:

- Load balance only those applications that meet load-balanced clustering requirements.
- Don't look at load-balanced clustering as a replacement for backups.
- Don't forget about power protection.
- Don't forget about disk failure.

Many of the risks associated with load-balanced clusters are common for nearly all network servers. The important rule of thumb to remember is that load-balanced clusters make only data access fault tolerant, not the data itself. You should remember to still perform regular backups of some load-balanced cluster nodes and also consider implementing fault-tolerant disks on each node. RAID 1 or RAID 5 will provide an additional level of fault tolerance for each load-balanced node.

Caution Don't forget about the common network-related single points of failure mentioned in Chapter 9. Remember that a loss to a switch, router, hub, or cable can kill access to your entire cluster.

Estimating Server Capacity

After looking at application setup, software licensing requirements, and risks to the load-balanced cluster, you'll have enough configuration information to size the needed capacity for each cluster node. The following should be your primary considerations for each node:

- CPU
- RAM
- Disk space
- Bandwidth

You can most likely solve CPU- and RAM-related performance issues by simply adding nodes to divert some of the application processing to other systems. Whether on a Windows or Linux host, the load-balancing service in general will consume only between 250KB and 4MB of RAM, so your primary focus on RAM sizing should be in the consumption of each application running on the cluster.

When considering disk space requirements, remember that by default load-balanced clusters store application data locally on each cluster node. This means your primary concern with disk sizing should be the requirements of the applications themselves. Make sure you have enough local disk space on every node in the cluster in order to ensure sufficient storage space.

With bandwidth requirements, you need to look at the size of the throughput to the cluster nodes. Given the bandwidth of most Internet connections, connecting all nodes to a 100Mb switch is usually sufficient. If the bottleneck is resulting from your network (which normally occurs only for LAN-based client access to the cluster), another alternative is to add NICs to each node. You could deploy NIC teaming to double the available bandwidth of each cluster node. Also, you could consider adding a third NIC that could be exclusively dedicated for node-to-node communications. As the cluster scales, you can tier and add switches as needed.

Note Adding a second NIC to provide more bandwidth to a cluster node is helpful only when each interface has its own bandwidth. Multiple NICs should always be connected to the network via a switch and not a hub.

Now that you've looked at the general planning requirements for load-balanced clusters, in the next section you'll examine the specific methods for building load-balanced clusters. You'll start with Windows Server 2003 load-balanced clustering and then move onto building load-balanced clusters on Linux servers.

Building Windows Network Load-Balanced (NLB) Clusters

It's now time to learn how to set up and manage load-balanced clusters. In the following sections, you'll learn the essential terminology and theory that's specific to Windows NLB clustering and then see the procedures for configuring load-balanced clusters.

Enabling the NLB Service

By default, NLB is already installed as a network service. This means all you should have to do is enable the network service by checking its associated box. After enabling the service, you'll then need to configure the NLB parameters to set up a cluster.

Note NLB is nothing more than a network service. It isn't an operating system service that can be managed from the Services MMC.

If you need to see whether the NLB network service is installed on the network adapters of each cluster node, follow these steps:

1. Click Start ► Control Panel ► Network Connections, and then click the NIC to be used in the NLB cluster.
2. In the Status dialog box, click the Properties button.

3. You should see a check box for Network Load Balancing, as shown in Figure 10-4. To enable the NLB service, click the box.

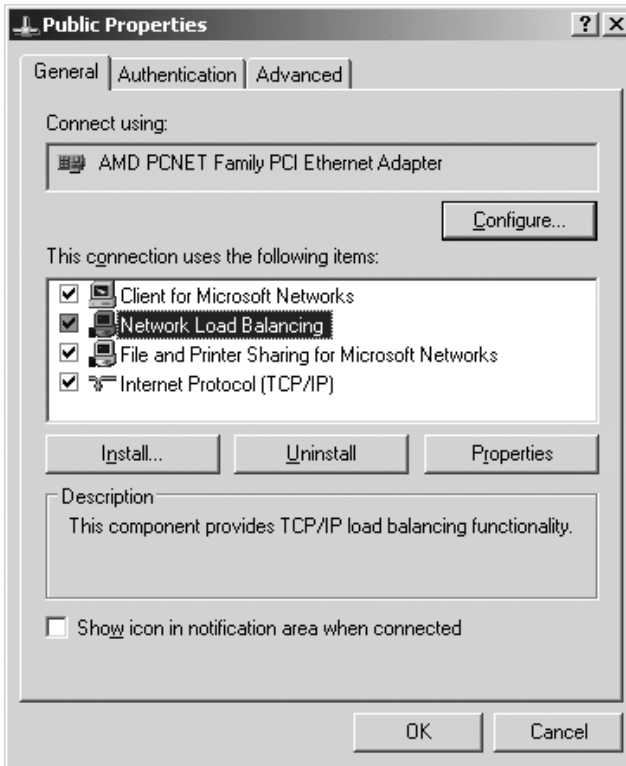


Figure 10-4. Enabling the NLB network service

With the service enabled, you'll now learn the essential terminology involved in configuring and managing the NLB service.

Understanding Unicast and Multicast

You can set the NLB cluster service to run in either *unicast* or *multicast* mode. Unicast is the default operation mode for the NLB service, but it's a valid choice only if you have two or more NICs in each NLB node. Otherwise, you'll have to use multicast. At this point, you're probably still scratching your head, so we'll first outline the differences between the two choices:

- **Unicast:** The NLB service changes the MAC address of the cluster network adapters to the same value for all hosts.
- **Multicast:** The NLB service adds a multicast MAC address of the cluster network adapters on all nodes in the cluster, leaving each adapter's existing MAC address unchanged.

Unicast is the default mode of operation, because not all switches support binding multiple MAC addresses to single NICs. Since with unicast you have a one-to-one ratio of MAC address to NIC, you'll need a second NIC in each node to facilitate node-to-node cluster communication. With every NIC having the same MAC address, it's not possible for one network interface to directly talk to another.

Caution Some NICs won't let the NLB cluster service change their MAC address. If your network cards don't allow the MAC to be changed, you'll have to either buy new NICs or configure the NLB service to use multicast.

If acquiring additional NICs isn't possible, then your choice must be multicast. Before just going with a multicast configuration, consider the logic behind paying thousands of dollars for each server OS and applications to run on the cluster, only to refuse to pay less than \$100 on an additional network adapter. If we still haven't talked you out of multicast, then we'd better tell you what else you should be aware of with its implementation.

For routers that don't accept Address Resolution Protocol (ARP) responses from the NLB service, you'll have to manually add static ARP entries to the router for each virtual IP address. With your routers configured properly, you shouldn't have any other significant issues with using multicast.

An alternative shortcut with multicast that usually circumvents the need to change your router or switch configurations is to connect all nodes in a multicast NLB cluster to a hub and then uplink the hub to a switch or router. Now the switch will see only the emulated MAC address on a single port (the one connected to the hub) instead of multiple ports. Figure 10-5 shows this configuration.

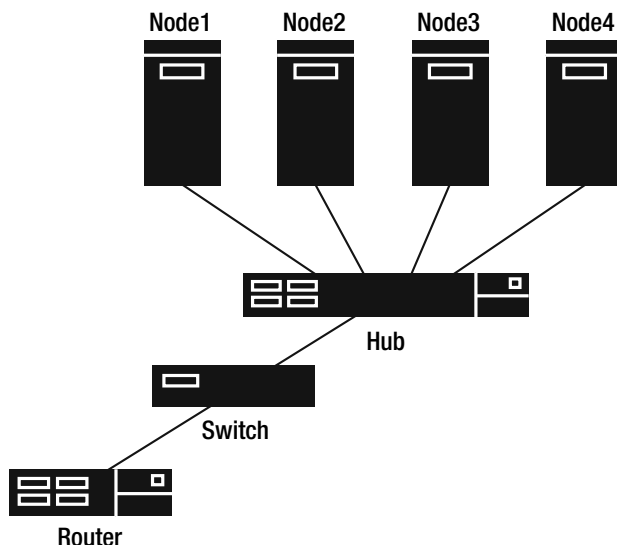


Figure 10-5. Using a hub to connect a multicast NLB cluster to the network

Without the hub in place, if the switch receives client requests to the NLB cluster (which includes the cluster's MAC address) and the switch doesn't recognize the MAC address, the switch will send the packet to all ports. This concept is known as *switch flooding*. Microsoft realized this flaw and with Windows Server 2003 added the IGMP Multicast setting to correct it. When in multicast mode, having the IGMP Multicast option selected ensures that the switch will route only NLB cluster packets to NLB cluster ports. This feature allows you to connect other systems to the same switch as the NLB cluster, without degrading performance, as would be the case with a hub. Later in the “Configuring and Managing Windows NLB Clusters” section, we'll show how to enable the IGMP Multicast setting.

Caution Configuring the IGMP Multicast option on an NLB cluster is useful only if the switch that the cluster is connected to supports this feature. Prior to planning for an IGMP Multicast cluster, consult with your switch vendor to ensure your particular switch model can support this feature.

Understanding Convergence

As with failover clusters, NLB cluster nodes communicate with each other through the use of heartbeats. To act and respond as a single entity, each NLB node must be aware of all other nodes in the cluster.

The process of all cluster nodes coming together as one is known as *convergence*. This process happens whenever a cluster node fails or a node configured to participate in the NLB cluster boots up. The actual convergence process accomplishes the following:

- Determines how many nodes exist in the NLB cluster
- Determines the node that will act as the default host (the host with the highest priority)

NLB cluster nodes are aware of each other by listening for heartbeats. If one node doesn't hear the heartbeat of another node within five seconds, that node is considered to have failed. When a node fails, convergence reoccurs and the NLB cluster is formed again with the remaining nodes. The convergence process doesn't interrupt client activity to any node on the cluster.

Note The timing and failure criteria for the cluster-node heartbeat are configurable parameters that you can change by editing the registry. We describe the process to make these changes in the “Using the Registry to Modify NLB Cluster Parameters” section later in this chapter.

You can validate convergence events by checking the Windows System Event log for events with the ID 28. Figure 10-6 shows this event. Checking for convergence events in the System Event log on one cluster node is an easy means to validate that all configured nodes are actively participating in the cluster.

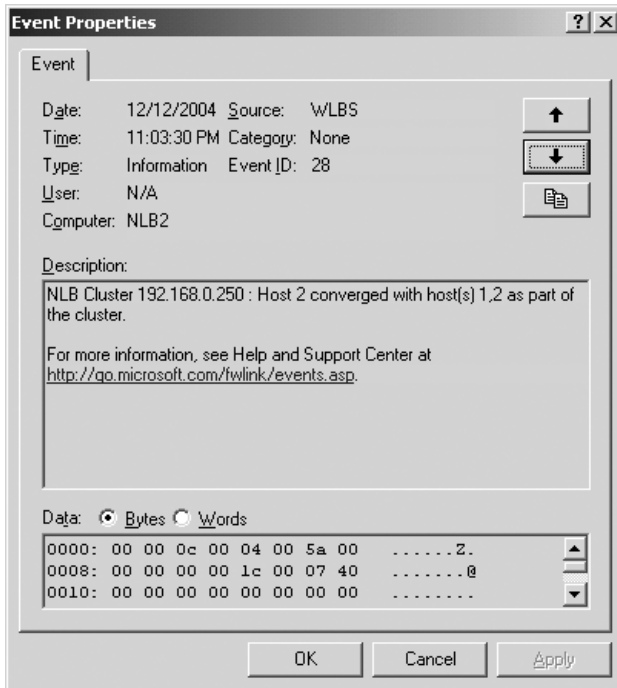


Figure 10-6. Convergence event stored in the System Event log

Setting Priority

Priority numbers serve two functions in an NLB cluster:

- They identify each node on the cluster.
- They prioritize the order in which cluster nodes should handle client requests.

Each cluster node must have a unique priority number. The assignable range of numbers is 1 to 32. With priority numbers, the lowest value has the highest priority. So if a cluster node has a priority of 1, it's the first to respond to client requests, if it isn't already busy. When several simultaneous attempts to access the NLB cluster are made, those requests are distributed amongst the many nodes in the cluster. Again, the initial request would be directed to node 1, followed by node 2, and so on.

Caution The priority number assigned to each cluster node must be unique. For example, you can't have two nodes configured to use priority 1 (the default value) in the same NLB cluster.

Setting Port Rules

You just read how cluster nodes can respond to client requests based on priority. What if you wanted two nodes in the cluster to respond to 80 percent of Hypertext Transfer Protocol (HTTP) requests and had four other nodes you wanted to use primarily for File Transfer Protocol (FTP)? This is where port rules come into play. With port rules, you have the ability to place a weight for each port on each node.

Using port rules allows you to divide how nodes handle client requests; they provide additional options for your load-balancing configuration and give you the ability to designate particular servers in the cluster to perform the majority of specific actions, based on the rules you'll assign. Still a little confused on port rules? Don't worry—we'll be spending plenty of time on them in the "Configuring Port Rules" section later in this chapter.

Understanding Remote Control

Remote control is a feature that allows you to remotely administer any NLB cluster node from the command line using the `wlbs.exe` command. This command allows you to perform tasks such as the following:

- Start and stop the NLB network service on the cluster node
- Pause and resume the NLB network service on the cluster node

Caution By default, NLB cluster nodes are remote controlled with the `wlbs.exe` command using TCP ports 1717 and 2504. If you don't want your Web server farm exposed to remote control from possible Internet users, you should strongly consider blocking these two ports on your firewall.

For those who prefer to remotely administer a network using GUI tools, Windows Server 2003 has a new tool for just that purpose.

Using the Network Load Balancing Manager

The Network Load Balancing Manager is a GUI tool that's new with Windows Server 2003. With this tool you can quickly configure one or several NLB cluster nodes simultaneously and even add nodes to an existing cluster. Figure 10-7 shows this handy tool.

The "Managing NLB Clusters with the Network Load Balancing Manager" section later in this chapter thoroughly covers NLB cluster administration with the Network Load Balancing Manager.

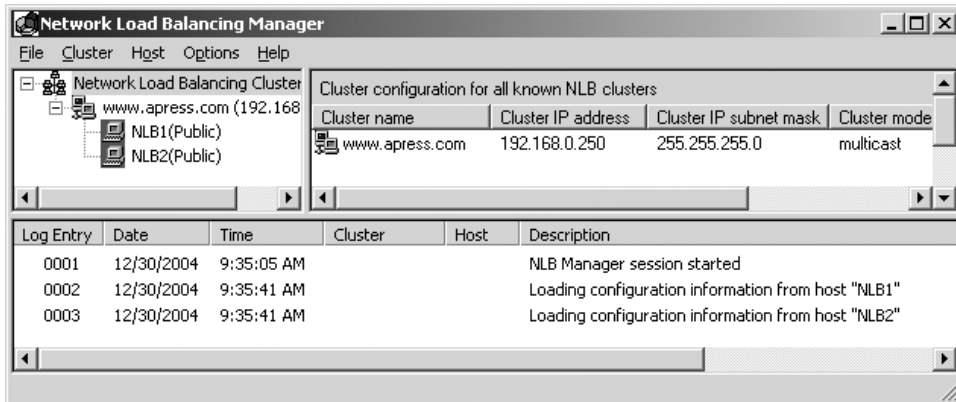


Figure 10-7. *The Network Load Balancing Manager*

Implementing Best Practices for NLB Cluster Implementations

Earlier in this chapter, we presented the fundamental steps for planning a successful load-balanced cluster deployment. To ensure optimal operation of your NLB cluster, you should strongly consider following each of these best practices:

- Verify that the applications and services to be load balanced on each cluster node are installed, are configured identically, and are running.
- Configure all cluster nodes so they collectively operate in either unicast or multicast mode (not both).
- Don't install NLB on a computer that's a part of a server cluster (not a Microsoft-supported configuration).
- Make sure TCP/IP is installed and configured on each cluster node, with a unique IP address for each node.
- Make sure TCP/IP is the only network protocol installed on the NLB cluster's network adapters.
- If possible, use two or more network adapters on each cluster node.
- Verify that configuration settings and port rules are consistent among all cluster nodes. You can most easily do this by configuring any global cluster setting using the Network Load Balancing Manager.
- Check to ensure that all NLB cluster nodes exist on the same network subnet.

With a handle on what you should do with NLB clusters, you'll now look at how you do it.

Configuring and Managing Windows NLB Clusters

Once you've selected the systems to participate in the cluster, you'll first need to enable the NLB service. If the NLB network service isn't installed (which it is by default), then you'll need to install it at this time. Here are the steps to install and enable the NLB network service:

1. Click Start ► Control Panel ► Network Connections, and then click the NIC to be used in the NLB cluster.
2. In the Status dialog box, click the Properties button.
3. If you see the Network Load Balancing service listed (refer to Figure 10-4 earlier), skip to step 8. If the service isn't present, then click the Install button to install the service.
4. Select Service, and click the Add button.
5. Select Network Load Balancing, and click OK.
6. If prompted, insert the Windows installation CD, and click OK.
7. Once the installation completes, you should see the Network Load Balancing service listed in the Local Area Connection Properties dialog box.
8. Check the box for the Network Load Balancing service, and click OK.

You'll need to repeat these steps on each node in the planned cluster.

Caution For unicast NLB clusters, make sure the network interface to be used for node-to-node cluster communication doesn't have the Network Load Balancing service enabled.

Performing the Initial NLB Node Configuration

Once the service is enabled, your work toward the completion of the NLB cluster implementation has just begun. The next steps you take depend largely on your rollout plan for the NLB cluster. Remember that the entire cluster must operate in either unicast or multicast mode. Without two or more NICs in every node, your only choice is multicast, because with unicast NLB, internal cluster communication isn't possible without two or more NICs per node. If you have multiple NICs in each node, then you can implement an NLB cluster that operates in unicast mode. Remember that unicast is the preferred method of operation and should be incorporated whenever possible.

Tip NLB cluster nodes don't dynamically update the IP address of the NLB cluster with their DNS server. Make sure you manually add an A (host) record for the NLB cluster in the proper DNS Forward Lookup Zone and a pointer record in the proper Reverse Lookup Zone.

The next two sections will take you through the steps for implementing unicast-mode and multicast-mode NLB clusters.

Note In the next two sections, we'll show you how to configure an NLB cluster one node at a time. If you want to configure all nodes simultaneously using the Network Load Balancing Manager, then skip ahead to the "Managing NLB Clusters with the Network Load Balancing Manager" section of this chapter.

Performing the Initial Configuration for a Multicast NLB Cluster Node

To configure a multicast NLB cluster, you'll need to perform these steps on each cluster node:

1. Click Start ► Control Panel ► Network Connections, and then click the NIC to be used in the NLB cluster.
2. In the Status dialog box, click the Properties button.
3. Under the General tab of the Local Area Connection dialog box, click Network Load Balancing and then click the Properties button.
4. You should now see the Network Load Balancing Properties dialog box (see Figure 10-8). Enter the IP address, subnet mask, and full Internet name (FQDN) for the NLB cluster.

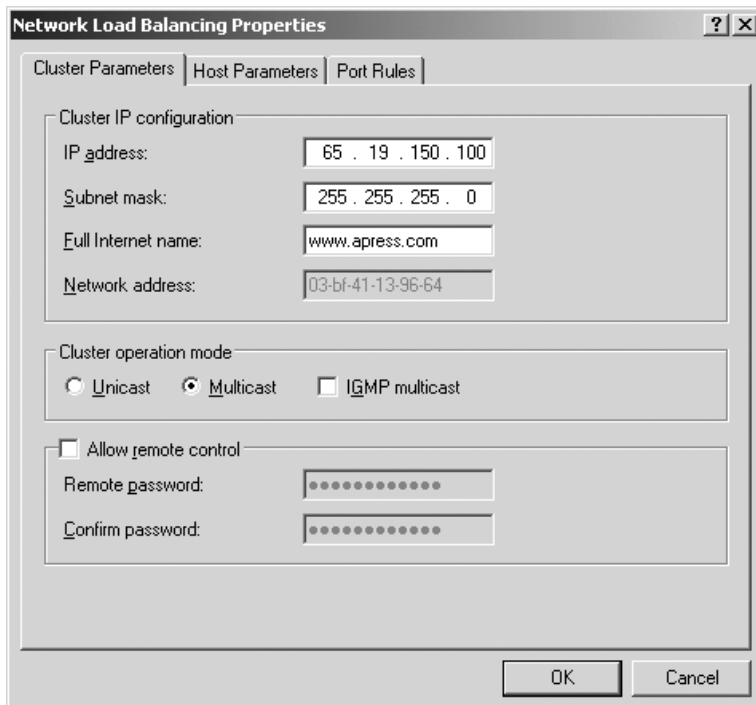


Figure 10-8. *NLB cluster properties*

5. For cluster operation mode, select Multicast. If desired, check the IGMP Multicast box.
6. To enable remote administration via `wlbs.exe`, check the Allow Remote Control box and enter a password.
7. Click the Host Parameters tab, as shown in Figure 10-9.

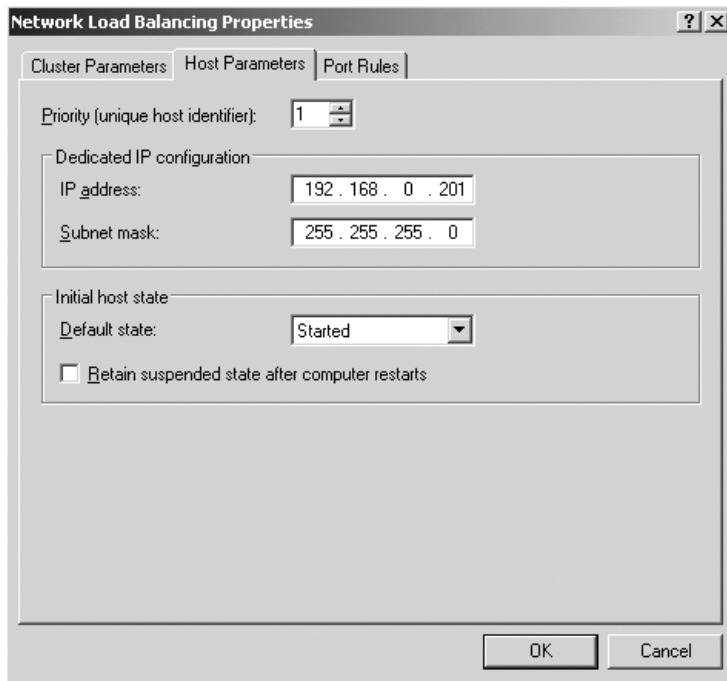


Figure 10-9. *NLB host parameters*

8. Select a priority for the cluster node between 1 and 32. As a general rule, it's best to give the most powerful server in the cluster a priority of 1, give 2 to the next most powerful, and so on. Remember, the number you select can't be the same as the configured priority for any other cluster node.
9. Enter the IP address and subnet mask of the host's network adapter. The values you enter must be identical to the IP settings for the adapter's first bound IP address.
10. If you'd like the node to be active in the cluster when you've completed the configuration, in the Initial Host State portion of the Network Load Balancing Properties window, select Started as the default state.
11. Click OK to close the Network Load Balancing Properties dialog box.
12. If prompted that you'll need to add the NLB cluster IP address to the TCP/IP component, click OK.
13. Click OK to close the network connection's dialog box.

With the NLB cluster parameters configured for the network interface, your final task in the initial node configuration is to add the NLB cluster's IP address to the network interface's listed IP addresses.

Here are the steps to complete the configuration of the NLB cluster node:

1. Click Start ► Control Panel ► Network Connections, and then click the NIC to be used in the NLB cluster.
2. In the Status dialog box, click the Properties button.
3. Under the General tab of the Local Area Connection Properties dialog box, click Internet Protocol (TCP/IP) and then click the Properties button. Make sure you don't accidentally check the TCP/IP box and thus disable it.
4. In the Internet Protocol (TCP/IP) Properties dialog box, click the Advanced button.
5. On the IP Settings tab, click the Add button in the IP Addresses field.
6. Enter the IP address and subnet mask of the NLB cluster, and click Add.
7. Your TCP/IP configuration should appear similar to what's shown in Figure 10-10. Click OK to close the Advanced TCP/IP Settings dialog box.

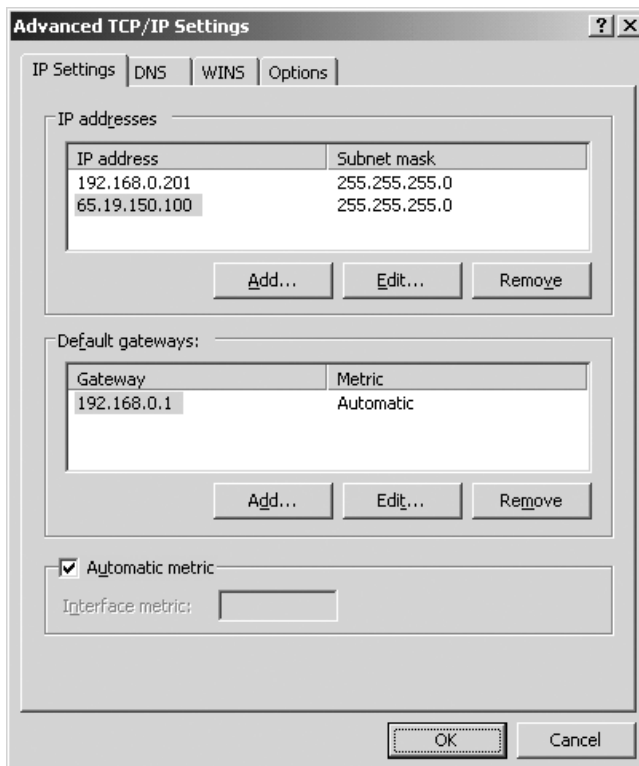


Figure 10-10. Advanced TCP/IP properties showing NLB cluster IP address

Caution Make sure the NLB cluster's IP address isn't the first listed IP address. The network interface's IP address should be listed first (the primary IP address), followed by the IP address of the NLB cluster.

8. Click OK to close the Internet Protocol (TCP/IP) Properties dialog box.
9. Click OK to close the Local Area Connection Properties dialog box.

For each additional node in the cluster, you'll need to repeat the previous nine steps. Once each node is configured, you should see an event with an event ID of 29 in the System Event log on each node. If you open the event (see Figure 10-6 earlier), you should see that each of the nodes is listed as having been converged. For example, for a four-node NLB cluster, you'd see a message in the event stating, "Host 1 converged as DEFAULT host with host(s) 1,2,3,4 as part of the cluster."

Now that you've done the initial setup of multicast cluster, next you'll learn how to configure a unicast NLB cluster. If you're interested in further configuring your multicast cluster, just skip ahead to the "Configuring Port Rules" section of this chapter.

Performing the Initial Configuration for Unicast NLB Cluster Nodes

Before beginning the initial configuration, remember that you'll need to have two available network adapters on each cluster node. One adapter will be used for node-to-node communications, and the other will be for cluster-to-client traffic. Before beginning the configuration, make a note of the TCP/IP address and subnet mask for the adapter to be used for node-to-node communications. Once you've determined the purpose for each NIC, you're ready to begin the network configuration.

To configure a unicast NLB cluster, you'll need to perform these steps on each node:

1. Click Start ► Control Panel ► Network Connections, and then click the NIC to be used in the NLB cluster.
2. In the Status dialog box, click the Properties button.
3. Under the General tab of the Local Area Connection dialog box, click Network Load Balancing and then click the Properties button.
4. You should now see the Network Load Balancing Properties dialog box (refer to Figure 10-8 earlier). Enter the IP address, subnet mask, and full Internet name (FQDN) for the NLB cluster.
5. For cluster operation mode, select Unicast.
6. Click the Host Parameters tab.
7. Select a priority for the cluster node between 1 and 32. Remember, the number you select can't be the same as the configured priority for any other cluster node.
8. Enter the IP address and subnet mask of the network adapter to be used for node-to-node communications. (This adapter can't have the NLB service enabled.) The values you enter must be identical to the IP settings for the adapter's first bound IP address.
9. If you'd like the node to be active in the cluster upon completion of the configuration, in the Initial Host State portion of the Network Load Balancing Properties window, select Started as the default state.

10. Click OK to close the Network Load Balancing Properties dialog box.
11. If prompted that you'll need to add the NLB cluster IP address to the TCP/IP component, click OK.

With the NLB cluster parameters configured for the network interface, your final task in the initial node configuration is to ensure that the NLB cluster's IP address is the same as the IP address of the interface on which NLB is configured.

Here are the steps to complete the configuration of the NLB cluster node:

1. Under the General tab of the Local Area Connection Properties dialog box, click Internet Protocol (TCP/IP) and then click the Properties button. Make sure you don't accidentally check the TCP/IP box and thus disable it.
2. In the Internet Protocol (TCP/IP) Properties dialog box, enter the IP address and subnet mask used for the NLB cluster. These settings should be the same for all nodes in the cluster.
3. Click OK to close the Internet Protocol (TCP/IP) Properties dialog box.
4. Click OK to close the Local Area Connection Properties dialog box.

The key with unicast NLB clusters is to ensure that each interface on each cluster node is configured correctly. Figure 10-11 depicts a three-node NLB cluster that has a public IP address of 65.19.150.100. In this configuration, the NIC on each node with the NLB service enabled will have an IP address of 65.19.150.100. The second NIC on each node (used for intercluster communication) has an IP address on the 172.16.1.x/24 subnet.

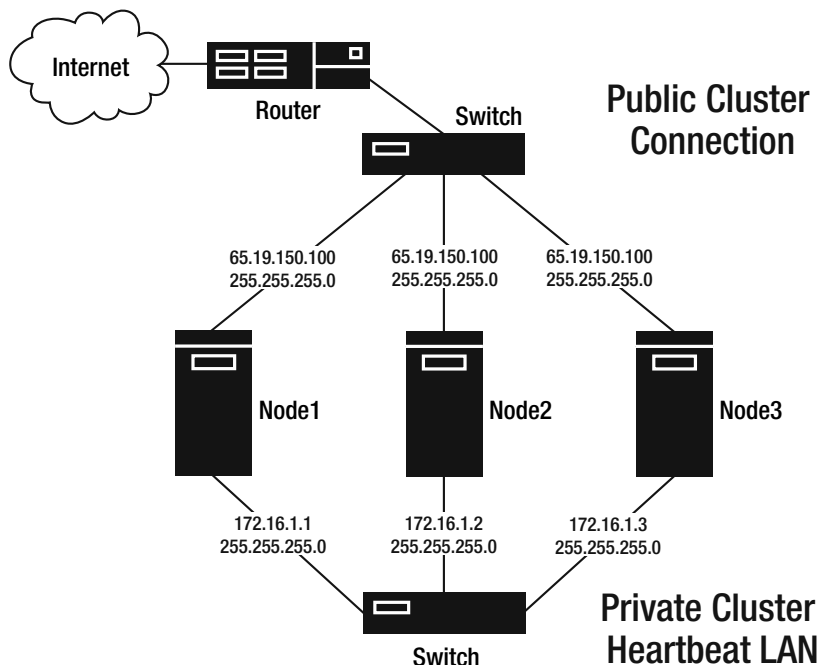


Figure 10-11. *Three-node unicast NLB cluster*

Testing the NLB Cluster

Once you've finished configuring the NLB service on each node, it's time to validate that the cluster nodes have converged as one. You can do this by opening the Computer Management administrative tool and then opening the System Event log and looking for an event with an event ID of 29. If you open the event (refer to Figure 10-6 earlier), you should see that each of the nodes is listed as having been converged. Another great way to test the cluster is to run a `ping -t` command to the cluster's public IP address and then unplug the public interface network cable from each cluster node one at a time. For the cluster shown in Figure 10-11, from a client you'd run the command `ping -t 65.19.150.100`. As each node is unplugged, you may see that the ping response takes a few milliseconds longer to return, but you shouldn't see any ping echo requests time out. Another approach to testing the cluster is to use your Web browser to view a Web page hosted by the cluster. After unplugging a node in the cluster, close and then reopen the browser and reconnect to the cluster's name or IP address. You should still be able to view the page. Closing and then reopening the Web browser is necessary because many Web browsers cache Web page information. If you leave your Web browser open and just hit the Refresh button to reconnect to a page, you may be viewing cached page content and not even know it. This is why testing HTTP reconnects with the browser remain-open is unreliable.

Configuring Port Rules

Configuring port rules is the final frontier for NLB cluster configuration. To this point, you've seen everything else you have to do. As you can see, it's easy to get an NLB cluster running, and that's why their use in the industry is continuing to grow.

Configuring port rules provides greater control of your NLB cluster by giving you the following abilities:

- You can limit the ports to which the cluster responds.
- You can weight different nodes to carry more or less of a load for certain ports.

For example, if you have a Web server farm that handles only HTTP traffic, you can configure a port rule so the NLB cluster responds only to requests made on port 80. You can apply this same concept to any service you plan on running on the NLB cluster. Simply create a rule for the port number used by the service.

When configuring port rules, you'll need to be aware of several settings:

- **Cluster IP Address:** When multiple virtual servers are hosted on the same NLB cluster, this setting allows you to configure a filter that applies to only one of the virtual servers in the cluster.
- **Port Range:** This allows you to select the range of ports to which the rule will apply.
- **Protocols:** This allows you to select whether to apply the rule to TCP, UDP, or both.

- **Filtering Mode:** This determines how the port rule will be applied. You can configure the Filtering Mode as one of the following:
 - **Multiple Host:** The rule will be applied to multiple hosts. You'll have to manually apply the rule to each host's NLB settings, unless you configure the settings using the Network Load Balancing Manager. When you choose this setting, you have the ability to configure affinity and load weight on each node.
 - **Single Host:** This specifies that network traffic for the port rule will be handled by a single host in the cluster.
- **Affinity:** When multiple host filtering is employed, the Affinity setting determines how the NLB service should direct client requests to the multiple cluster nodes. You can configure the Affinity setting as follows:
 - **None:** This setting means the NLB service won't maintain client connections with a particular node.
 - **Single:** This setting, which is the default, allows clients to perform all the necessary tasks with the node in which they originally connected. This setting should always be used if the node maintains session state data for the client, such as server cookies between connections.
 - **Class C:** This setting is used primarily when it's possible that clients will connect to the cluster through one of several proxy servers. This option ensures that all requests from a client will be handled by the same node during the client's session.
- **Load Weight:** This allows you to configure the rule so its associated traffic is equally distributed among all cluster nodes, or you can set an individual weight to each node as a percentage. This means if you had a two-node cluster (let's keep it simple!) and set the load weight on one node to 80 and set the weight on the other to 20, the first node would respond to 80 percent of client requests.
- **Handling Priority:** When single-host filtering is used, the handling priority determines how each single host will respond to client requests. You can assign each node a unique priority value for the port rule. Handling priority allows some nodes to have precedence for HTTP traffic, for example, and others to have precedence for FTP traffic.

With an understanding of the options associated with port rules, you'll now look at the steps to configure them. Remember, you must configure the rules on every node in the cluster. A rule configured on one node doesn't replicate to all other nodes. With this in mind, although in this section you'll see how to configure port rules node by node, you can configure a port rule and apply it to all nodes in the cluster using the Network Load Balancing Manager.

Caution The NLB service doesn't allow rules to overlap. Before adding any new rules, you'll first need to remove or edit the default rule, which applies to all available ports.

To configure port rules, follow these steps:

1. Click Start ► Control Panel ► Network Connections, and then click the NIC to be used in the NLB cluster.
2. In the Status dialog box, click the Properties button.
3. Under the General tab of the Local Area Connection dialog box, click Network Load Balancing and then click the Properties button.
4. In the Network Load Balancing Properties dialog box, click the Port Rules tab.
5. To edit the existing default filter, click the Edit button. Remember that the default filter covers all ports, so you'll need to either edit the default filter or delete it in order to add filters that will perform as they're configured. If you wanted to add a new filter, you'd click the Add button.
6. In the Add/Edit Port Rule dialog box, enter the parameters for the filter. In the example shown in Figure 10-12, a port rule is configured so that one node in the cluster handles 70 percent of all HTTP (port 80) traffic. Assuming that there were three other nodes in the cluster, each additional node could be configured to handle 10 percent of the port 80 traffic ($70\% + 10\% + 10\% + 10\% = 100\%$).

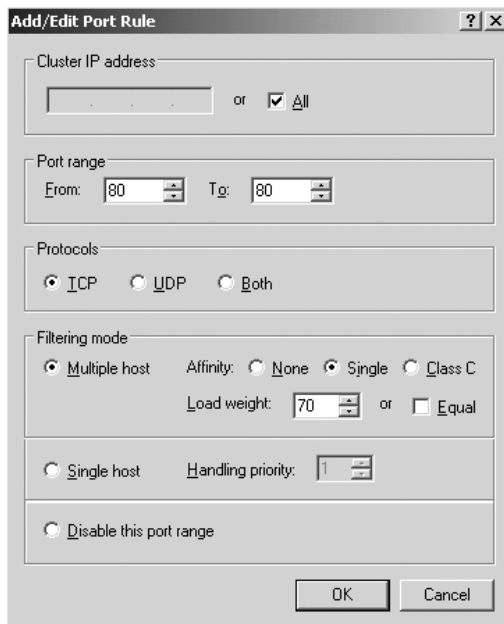


Figure 10-12. *Configuring HTTP port rule*

Tip You can also block specific ports or port ranges by creating a filter and selecting the Disable This Port Range option.

7. Once the port rule is configured, click OK in the Add/Edit Port Rule dialog box.
8. Click OK to close the Network Load Balancing Properties dialog box.
9. Click OK to close the network connection Properties dialog box.

Managing NLB Clusters with the Network Load Balancing Manager

One of the best features included with the Windows Server 2003 OS is the Network Load Balancing Manager administrative tool. The Network Load Balancing Manager greatly eases your administrative efforts for managing NLB cluster nodes. Say goodbye to having to repeat the same process 30 times, for example, to make one change on an NLB cluster consisting of 30 nodes. With Windows 2000, you had to be careful to make sure all configuration parameters were consistent from node to node with an NLB cluster. Now with the Network Load Balancing Manager, you can make one change to the cluster as a whole and automatically apply the change to all the cluster nodes.

To open the Network Load Balancing Manager, click Start ► Administrative Tools ► Network Load Balancing Manager. With the management tool open, you'll see some common administrative tasks that you can perform.

Monitoring Status of an Existing Cluster

To check the status of an existing cluster, follow these steps:

1. Open the Network Load Balancing Manager.
2. Click the Cluster menu, and select Connect to Existing.
3. In the Connect dialog box, enter the IP address or FQDN of the cluster node to connect to and then click the Connect button.
4. Now click the cluster shown in the lower portion of the window, and click the Finish button.
5. The cluster and all its nodes should now be displayed. You should see all active nodes listed, along with their priority and initial state (see Figure 10-13).

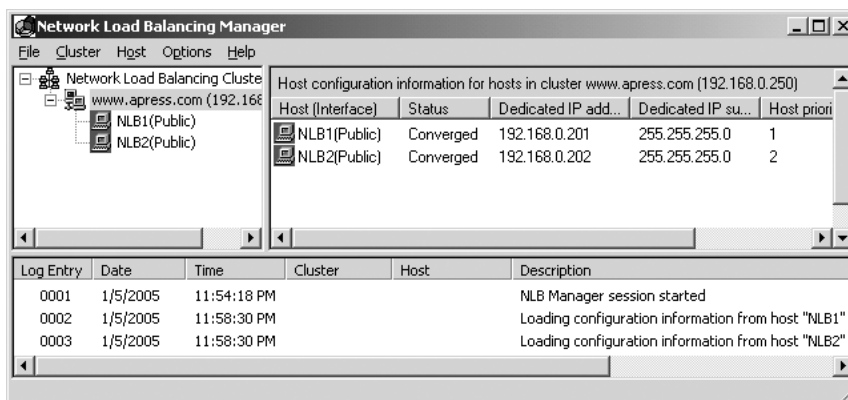


Figure 10-13. *The Network Load Balancing Manager*

Changing Configuration Parameters of an Existing Cluster

One of the best features of the Network Load Balancing Manager is that it allows you to simultaneously make changes to all cluster nodes. With this tool, you no longer have to manually duplicate configuration changes one node at a time.

To modify cluster configuration parameters, follow these steps:

1. Open the Network Load Balancing Manager, and connect to the cluster you want to manage.
2. Right-click the cluster object in the Network Load Balancing Manager GUI, and select Cluster Properties.
3. Under the cluster object, you can create and modify clusterwide port rules. Any changes will be applied to all active cluster nodes.

Caution Unlike with server clusters, NLB clusters don't maintain a quorum log holding configuration data. Therefore, any changes you make to the cluster won't be applied to any offline nodes. You'll have to manually duplicate the changes on offline nodes.

4. Once you've modified the cluster properties, click OK.

Note You can also modify the configuration of any cluster node by right-clicking the node's icon in the Network Load Balancing Manager and selecting Properties.

Adding a Node to an Existing Cluster

To add nodes to the cluster, follow these steps:

1. Open the Network Load Balancing Manager, and connect to the cluster you want to manage.
2. Right-click the cluster object in the Network Load Balancing Manager GUI, and select Add Host to Cluster.
3. Enter the IP address of the host to add, and click OK.
4. Enter or verify that the host's configuration parameters are correct, and click OK.

Removing a Node from the Cluster

The Network Load Balancing Manager also allows you to quickly and easily remove hosts from the cluster. Removing a host automatically disables its NLB network service.

To remove a node from the cluster, follow these steps:

1. Open the Network Load Balancing Manager, and connect to the cluster you want to manage.
2. Right-click the node object for the node you want to remove, and select Delete Host.
3. When prompted, click Yes to remove the host.

Creating a New Cluster

The Network Load Balancing Manager also lets you quickly create a cluster and then add hosts to the cluster. To create a new cluster, follow these steps:

1. Open the Network Load Balancing Manager.
2. Click the Cluster menu, and select New.
3. Enter an IP address, subnet mask, and FQDN for the cluster.
4. Select either unicast or multicast operation, and click Next.
5. You can now add hosts using the steps listed earlier.

Managing NLB Clusters with the `wlbs.exe` Command

The `wlbs.exe` command allows you to locally or remotely administer an NLB cluster or individual cluster nodes from the command line. The command allows you to start and stop the NLB service on the entire cluster or on individual nodes. You can also query the status of the cluster and display the cluster's configuration parameters.

The syntax for the `wlbs.exe` command is as follows:

```
WLBS <command> [<cluster>[:<host>]] [/PASSWD <password>] [/PORT <port>] [/LOCAL]
```

Table 10-1 describes the commands and options for the `wlbs.exe` command.

Table 10-1. *wlbs.exe Command Options*

Command	Description
disable <port> all	Disables all traffic for port rule for port specified. If no port is entered, disables all traffic for all ports in the NLB cluster.
display	Shows current cluster status, configuration parameters, and recent System Event log messages.
drain <port> all	Disables new traffic for port rule for port specified. If no port is entered, the command disables new traffic for all NLB cluster ports.
drainstop	Causes cluster to stop operations once all existing connections are terminated. Newly attempted connections will be rejected.
enable <port> all	Enables all traffic for port rule for port specified. If no port is entered, enables all traffic for all ports in the NLB cluster.
help	Displays online help.
ip2mac <cluster>	Displays the MAC address for the specified cluster.
query	Displays which hosts are converged with the cluster and the current cluster state. Hosts are listed by their unique priority number. Cluster states may be as follows: Converged: The cluster has converged, and the responding host isn't the default host (highest priority). Converged As Default: The cluster has converged, and the responding host is the default host. Converging: The cluster is in the process of converging. Draining: The cluster is in the process of draining all active connections and will then stop. Unknown: The host responding to the request hasn't started the NLB cluster network service and has no knowledge of the cluster's state.
reload	Reloads cluster network parameters from the registry.
resume	Resumes the use of remote cluster control commands.
start	Starts cluster operations on the specified host or cluster.
stop	Stops cluster operations on the specified host or cluster.
suspend	Suspends the use of remote cluster control commands.

Table 10-2 describes the remote options for the `wlbs.exe` command. You should use them when remotely administrating an NLB cluster. These commands allow you to specify the NLB cluster or host on which to execute the `wlbs.exe` command.

Table 10-2. *wlbs.exe Remote Options*

Remote Option	Description
<cluster>	Specifies the cluster on which to run the <code>wlbs.exe</code> command operation.
<host>	Specifies the target host for the command. By default, the command affects all NLB cluster hosts.
/local	Causes <code>wlbs.exe</code> command to run only on the local machine.
/passw <password>	Specifies remote-control password for the NLB cluster.
/port <port>	Specifies cluster's remote-control UDP port number.

Still a little unsure of how to use the `wlbs.exe` command? We'll show some examples.

The command and output from querying the NLB cluster 65.19.150.100 would be as follows:

```
C:\>wlbs query 65.19.150.100
WLBS Cluster Control Utility V2.4 (c) 1997-2003 Microsoft Corporation.
Cluster 65.19.150.100
Host 1 has entered a converging state 1 time(s) since joining the cluster
And the last convergence completed at approximately: 1/5/2005 11:38:57 PM
Host 1 converged as DEFAULT with the following host(s) as part of the cluster:
1, 2, 3, 4, 5
C:\>
```

Another common operation with NLB clusters is to prevent additional HTTP traffic (port 80) from connecting to the NLB cluster if you need to take it down for maintenance. You can use the command to stop all HTTP traffic on the entire cluster by specifying the cluster name in the syntax or can use the command to stop port traffic on a single node. The command and resultant output to stop additional HTTP traffic on the local NLB cluster node is as follows:

```
C:\>wlbs drain 80
WLBS Cluster Control Utility V2.4 (c) 1997-2003 Microsoft Corporation.
Cluster 65.19.150.100
NEW traffic handling for specified port rule(s) disabled.
C:\>
```

Using the Registry to Modify NLB Cluster Parameters

You can edit the registry to modify several NLB cluster parameters. You may find it necessary to modify the registry in order to improve cluster performance or to troubleshoot clusters.

Caution Always exercise caution when modifying the registry. For cluster consistency, you must ensure that any registry value changed on one node is also changed on all other nodes.

Table 10-3 describes the most commonly edited NLB-related registry values, with their purpose and configurable parameters.

Table 10-3. *Common NLB Registry Values*

Registry Value	Default Setting	Range	Description
AliveMsgPeriod	1000	100–10000	Time (in milliseconds) between node heartbeat broadcast messages.
AliveMsgTolerance before declaring a	5	5–100	Number of AliveMsgPeriod periods to wait nonresponsive NLB host offline and restarting convergence.
NetmonAliveMsgs	0	0–1	When set to 1, the Network Monitor will be enabled to capture NLB heartbeat messages.
RemoteControlUDPPort	2504	0–65535	Port used for wlbs.exe command remote-control commands.

Most NLB registry parameters are modified to enhance NLB cluster performance. For example, you may want to increase the `AliveMsgPeriod` value to decrease NLB cluster node broadcast traffic. If you suspect trouble with an NLB cluster node, you may deem it necessary to verify that the node is sending heartbeat messages to the cluster.

Now that you've examined load-balancing clusters on Windows in great detail, in the next section you'll look at the most predominant form of load balancing on Linux: LVS clustering.

Building Linux Virtual Server (LVS) Clusters

Although third-party tools exist, most Linux load-balanced clusters are designed around the LVS Project. Compared to the Microsoft NLB architecture, you'll see that LVS uses a fundamentally different approach. With LVS, one or two servers must exist outside the cluster to distribute client traffic amongst cluster members. This means that to build a four-node LVS cluster, you'll need at least five servers. Figure 10-14 shows this configuration.

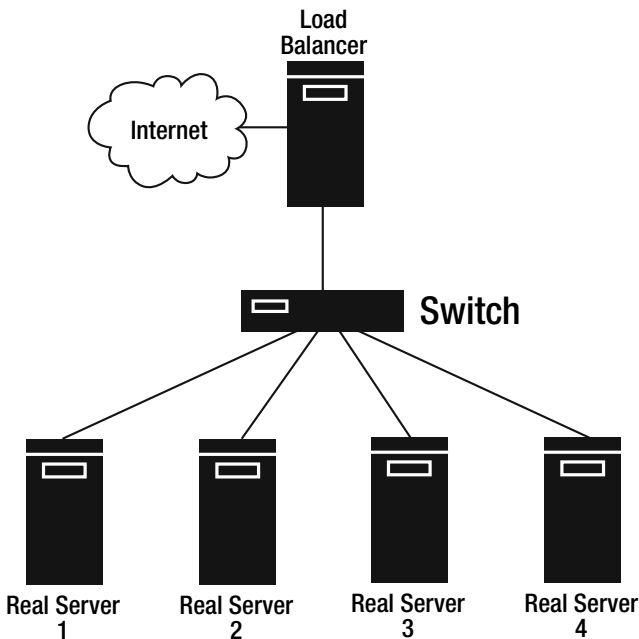


Figure 10-14. *Four-node LVS cluster*

In Figure 10-14, the server labeled as Load Balancer accepts incoming client requests and directs them to an internal RS. Each RS is a cluster node. With the load balancer directing client traffic, the RS nodes in the cluster can be placed on the same LAN or distributed across a wide-area network (WAN). This architecture has one fundamental flaw: fault tolerance. If the load balancer fails, then the entire cluster goes down. To overcome this problem, most LVS cluster implementations use two systems as load balancers. One system serves as the active load balancer, and the second system is passive, coming online only in the event that the active system fails. Figure 10-15 shows this configuration.

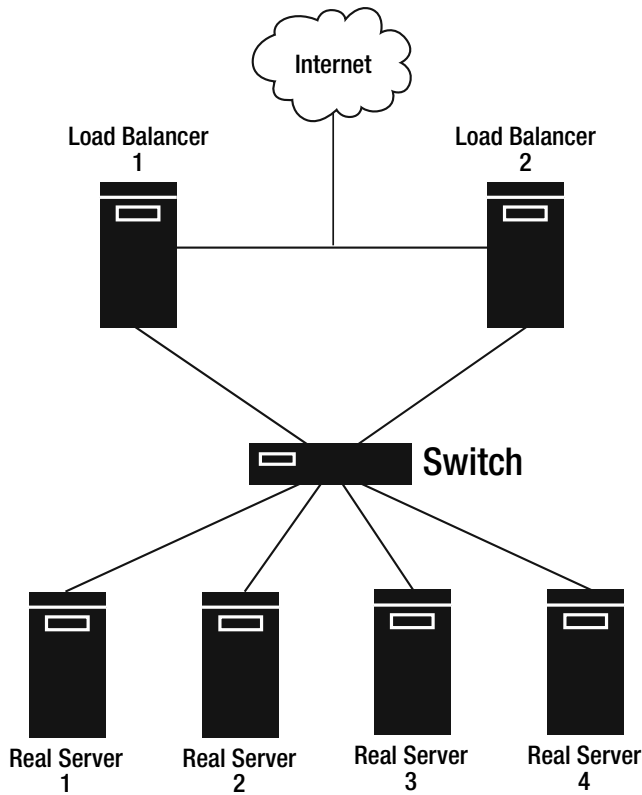


Figure 10-15. *Fault-tolerant four-node LVS cluster*

For small fault-tolerant LVS clusters, the two additional servers could be a financial burden; for larger cluster implementations, the additional servers often get lost in the overall cost of 20 other servers.

Understanding LVS Architecture

LVS is generally configured in one of three different ways:

- Virtual server via NAT
- Virtual server via IP tunneling
- Virtual server via direct routing

In the next three sections, we'll cover each of these unique configurations.

Virtual Server via NAT

With the virtual server via NAT architecture, the load-balancer server is dual-homed and NATs all traffic to the real servers on an internal LAN. (Figures 10-14 and 10-15 showed this configuration.) With NAT, each load-balancer server directs client traffic into the internal LAN and to a real server. When the RS replies, the reply goes back through the load-balancer system before returning to the requesting client.

With the NAT server, all traffic into and out of the cluster must travel through a single server. The greatest problem with this approach is scalability, with the LVS cluster typically not being able to scale beyond 10–20 nodes.

Virtual Server via IP Tunneling

Several advantages exist with virtual server via IP tunneling, most notably of which is scalability. Unlike configuring LVS via NAT, the IP tunneling approach causes the load-balancer server to simply direct client requests to the real servers via a VPN tunnel. Replies from the real servers will use a different network. This will significantly decrease the workload of the LVS load balancer. Figure 10-16 shows this approach.

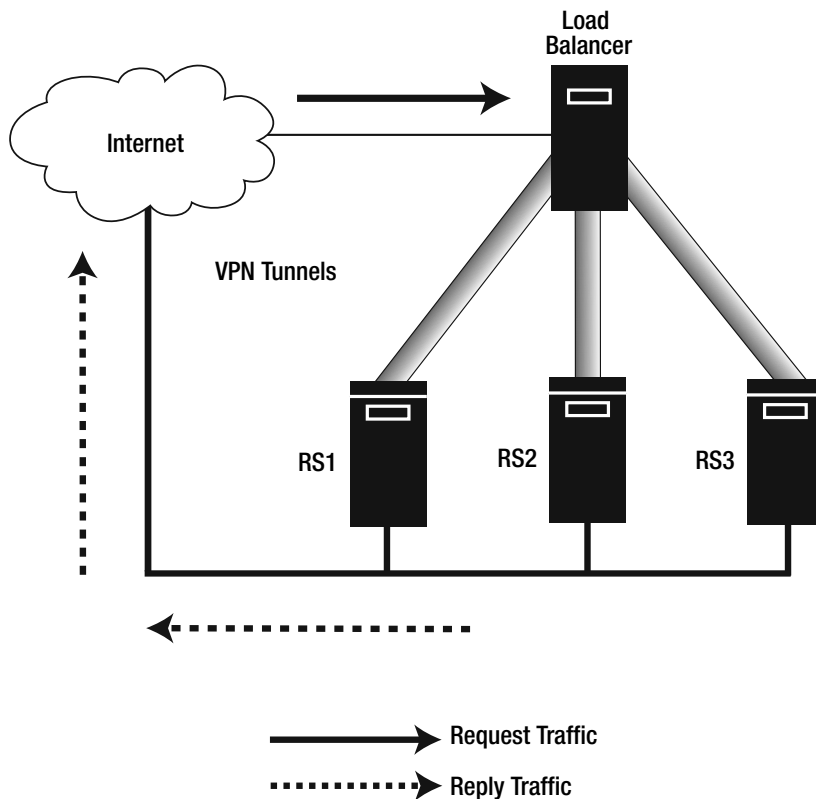


Figure 10-16. *Virtual server via IP tunneling cluster*

With the use of VPN tunneling, this cluster can easily be distributed amongst multiple sites and connected via the Internet. Using a VPN to logically connect all real servers to the load balancer ensures privacy and security.

Virtual Server via Direct Routing

The virtual server via direct routing approach is similar in configuration to the virtual server via NAT, with the exception that reply traffic won't flow back through the load balancer; instead, replies will be sent directly from the real servers to the requesting client. Like with a virtual server via NAT, real servers connect to the load balancer via a LAN. Replies from the real servers would return to the client over a different LAN segment, which is ultimately routed back to the Internet.

Implementing LVS Implementation

Unfortunately, implementing LVS isn't exactly a walk in the park. Configuring an LVS cluster will require a kernel recompile and will require that you download the LVS cluster software from <http://www.linuxvirtualserver.org>. Setup steps will vary by Linux distribution and by kernel version, so to make your life easier, we've included all the GPL LVS installation documentation on this book's companion CD. To view the setup documents, navigate to the `Book_Files/Ch10/LVS` folder on the CD. In the folder, you'll see an `LVS_mini_how_to.pdf` file and an `LVS_how_to.pdf` file. Combined, these guides, written by Joseph Mack, offer more than 200 pages of in-depth LVS cluster configuration information.

If you're most interested in the latest and greatest available documentation, it's always best to check for updated information at <http://www.linuxvirtualserver.org/documents.html>. Here you'll find Joseph Mack's latest contributions to the LVS project, as well as other useful documents, tips, and tricks.

Of course, to set up LVS, you'll also need the LVS software. This is also included on the companion CD in the `Book_Files/Ch10/LVS` folder. For the most recent LVS software, point your Web browser to <http://www.linuxvirtualserver.org/software.index.html>.

Summary

In this chapter, you examined load-balanced clusters. One of the most difficult aspects of learning clustering, for most people, is that you usually need two to four servers and lots of hardware in order to practice cluster configuration. Thanks to virtualization, this is no longer the case. In the next chapter, you'll see ways to configure both failover and load-balanced clusters using VM technology. This gives you the ability to learn, test, and demonstrate clustering technologies on a single system. Turn the page to let the fun begin!



Building Virtual Machine Clusters

So far, you've seen the value of clustering technologies, but you probably also have an appreciation for their price. Running clusters inside VMs running on a single host isn't normally a task for production environments, but it does have several other uses.

We've talked to hundreds of cluster administrators across the United States in search of just one administrator who has no problem training others on clustering using production cluster servers. As you might have guessed, the search isn't going too well. With cluster hardware being so expensive and with cluster implementations being mission-critical, most organizations don't want to put their production clusters at risk for the sake of training. Ultimately, the result is that many administrators find themselves having to learn clustering for the first time on the job. To overcome the problems with training new users on clustering, VM clusters are the ideal solution.

Another common use for VM clusters is as a means to test the recovery of your production cluster servers. If you manage clusters in a production environment, you probably run at least one backup of the cluster each day. However, how often do you get to test your backups and practice disaster recovery? For many organizations that don't have a test cluster configuration available in their labs, the answer to this question is, never!

In this chapter, we'll walk you through several cluster configuration recipes for Windows and Linux operating systems. As you'll soon see, the hard part to clustering with VMs is in getting the VMs to see and work with virtual shared storage disks while not corrupting them in the process. Once you have the proper shared storage in place and have configured each VM OS, the rest of the process is no different from setting up clusters on actual physical servers. We'll start by covering how to set up Microsoft Windows VM clusters.

Building Microsoft VM Clusters

With Windows Server operating systems, you can configure both server clusters and network load-balanced clusters. In the following sections, we'll show how to configure VMs to support SCSI and iSCSI server clusters, and then we'll show the configuration procedures for setting up network load-balanced clusters.

Note Chapter 12 discusses iSCSI concepts in detail.

Setting Up Windows Server Clusters

When running Windows Server clusters in a VM environment, your primary obstacle is the configuration of the shared storage. With VMware and Virtual Server, this can be a tricky endeavor. You have several ways to configure shared storage if you want your cluster *not* to work. However, if you want the cluster to work, then you have to follow specific guidelines. Probably the most frustrating aspect of configuring a Microsoft cluster inside a VM is that it will initially appear to work. The real test will come when you initiate the first failover from one node to another. After the first failover, you may find that the shared virtual disks are corrupt. If they aren't corrupted during the first failover, you may find that the disks become corrupt the first time you shut down both nodes and then attempt to boot them again.

Rather than continuing to scare you with what can happen if you build the VM cluster incorrectly, we'll show you how to build a working VM Windows Server cluster. In the sections that follow, you'll see the configuration steps for all major VM applications, including VMware Workstation, VMware GSX Server, and Microsoft Virtual Server. Also, if you're looking to build a cluster with more than two nodes or would like to build a cluster using Microsoft Virtual PC, jump ahead to the "Setting Up iSCSI Windows Server Clusters" section.

Note Once you set up the VMs according to the procedures outlined in the section that pertains to your VM application, you can then advance to the "Installing the Windows Server 2003 Cluster Service" section in this chapter to complete the setup.

Configuring Clusters on VMware Workstation

We'll start by showing you how to set up a server cluster on VMware Workstation. Like with an actual cluster implementation on physical systems, the most challenging aspect of configuring a server cluster using VMware is getting the shared storage to function properly. Unfortunately, if you make a mistake in configuring the shared storage, you usually won't realize it until the first time you attempt a failover with the cluster. You'll know you have a problem if a disk resource or other cluster resources are unable to come online following the failover.

Caution VMware officially supports cluster VM configurations only on its GSX Server and ESX Server product lines. While we've had success clustering VMware Workstation VMs using the clustering procedure documented in this section, if you run into problems with this configuration, VMware won't lend support.

Your first task in configuring the cluster is to prepare two Windows Server 2003 Enterprise Edition VMs.

Each VM should have the following general settings:

- A single 4GB IDE virtual hard disk for the OS and applications
- At least 128MB of RAM
- Two virtual host-only NICs

To meet the requirements for clustering, one of the VMs needs to be configured as a domain controller (unless the VMs have access to another domain controller). Both VMs should be members of the same domain. Once the VMs have the Windows Server 2003 Enterprise Edition OSs installed and configured, you'll then need to power down the VMs so you can configure their shared storage. At this point, you should have folders on the VM host for your two cluster-node VMs. This is also a good time to configure a separate folder for the cluster's shared disk resources. Figure 11-1 shows this folder structure. Notice that there's a folder for each VM, as well as a folder named Shared Disks.



Figure 11-1. VM cluster folder structure

With the VMs ready to go, you now need to set up the shared disk resources. Since VMware Workstation doesn't natively support cluster configuration, using the standard .vmdk virtual disk files won't work. However, using plain (.pln) disk files will allow you to build a working Windows VM cluster. You'll find a plain virtual disk file that you can use to build the cluster on the companion CD. Look in the Book_Files\Ch11 folder, and you'll see a file named PlainDisk500MB.zip.

Caution Plain disks don't dynamically expand, so when you unzip the PlainDisk500MB.zip file, its two unzipped files will consume 500MB of disk space.

To configure the shared storage resources, follow these steps:

1. Unzip the PlainDisk500MB.zip file to the Shared Disks folder created earlier.
2. At this point, you should see two files, PlainDisk500MB.pln and PlainDisk500MB.dat, in the folder. Rename each file to match its settings for the shared SCSI bus. For example, if the first unzipped plain disk will act as virtual SCSI disk 0:0, you could rename the PlainDisk500MB.pln file to SCSI0-0.pln. Next, you'd rename the PlainDisk500MB.dat file to SCSI0-0.dat.
3. For each additional disk you want to share in the cluster, repeat steps 1 and 2.
4. Once you've created the disk files, you need to configure each .pln file to map to its associated .dat file. To do this, open the first .pln file (for example, SCSI0-0.pln) in Notepad. Next, edit the path listed after the ACCESS parameter so it points to the plain disk's associated .dat file. For example, the following .pln file configuration maps to the G:\VMs\W2K3 Cluster\Shared Disks\SCSI0-0.dat file:

```
DRIVETYPE      scsi
CYLINDERS      500
HEADS          64
SECTORS        32
ACCESS "G:\VMs\W2K3 Cluster\Shared Disks\SCSI0-0.dat" 0 1024000
```

5. Assuming you decided to configure three shared disks in the cluster, you'd wind up with the following files in the Shared Disks folder:

```
SCSI0-0.dat
SCSI0-0.pln
SCSI0-1.dat
SCSI0-1.pln
SCSI0-2.dat
SCSI0-2.pln
```

6. Open the VMware Workstation UI, and open the first VM to participate in the cluster. Then click the Edit Virtual Machine Settings link.
7. In the Virtual Machine Settings dialog box, click the Add button.
8. When the Add Hardware Wizard opens, click Next.
9. In the Hardware Types field, select Hard Disk and click Next.
10. Select Use an Existing Virtual Disk, and click Next.
11. In the Select an Existing Disk dialog box, click the Browse button.
12. Navigate to the Shared Disks folder on the host system.

13. Now click the Files of Type drop-down menu, and select the VMware Plain Disks (*.pln) option.
14. Click the first SCSI plain disk file (for example, SCSI0-0.pln), and click Open.
15. Now click the Advanced button.
16. In the Virtual Device Node drop-down menu, select the SCSI value that pertains to the plain disk you're adding. For example, for the SCSI0-0 plain disk, you'd select SCSI 0:0 from the menu. Once you've selected the SCSI address, click Finish.
17. Repeat steps 7–16 for each plain disk to be added to the shared storage configuration.
18. Click OK to close the Virtual Machine Settings dialog box.
19. Repeat steps 6–18 on the second VM in the cluster.

When you've finished adding the virtual hard disk files, each VM's configuration should look like Figure 11-2.

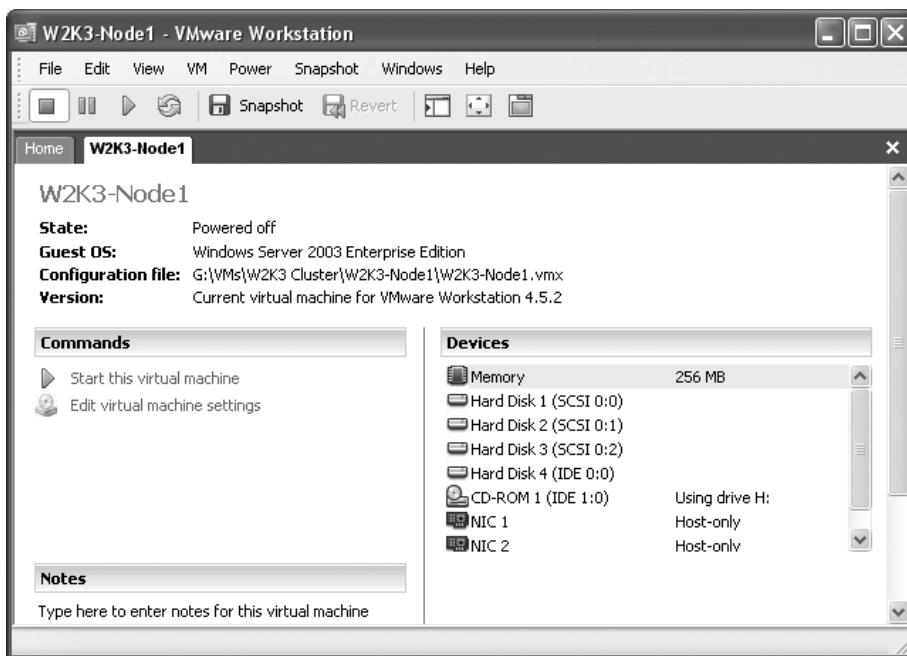


Figure 11-2. VM hardware configuration

Once you've assigned the virtual disk files to each VM, you'll need to close the VMware application. This is necessary because each VM's configuration file now requires some manual editing. Once you've closed the VMware application, follow these steps:

1. Locate the VM's .vmx file in Windows Explorer. By default, this file will be in the VM's associated folder.
2. Right-click the .vmx file, select Open With, and then select Notepad.
3. In Notepad, add the following line to the beginning of the configuration file:

```
disk.locking = "False"
```

4. Locate the lines that identify each shared virtual SCSI disk. Before the first SCSI disk definition, add the following line (see Figure 11-3):

```
scsi0.sharedBus = "virtual"
```

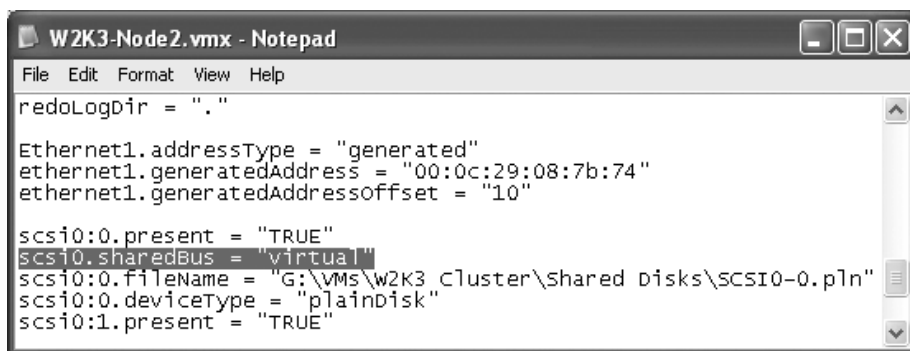


Figure 11-3. Edited VM configuration file

5. Save the .vmx file in Notepad.
6. Repeat steps 1–5 for the second VM in the cluster.

At this point, you have all the hardware in place and configured to set up a Microsoft server cluster using the two VMs. For detailed procedures on installing the Cluster Service, turn to the “Installing the Cluster Service” section later in this chapter.

Configuring Clusters on VMware GSX Server

Your first task in configuring the cluster is to prepare two Windows Server 2003 Enterprise Edition VMs. Each VM should have the following general settings:

- A single 4GB IDE virtual hard disk for the OS and applications
- At least 128MB of RAM
- Two virtual host-only NICs

To meet the requirements for clustering, you'll need to configure one of the VMs as a domain controller (unless the VMs have access to another domain controller). Both VMs should be members of the same domain. Once the VMs have the Windows Server 2003 Enterprise Edition OSs installed and configured, you'll then need to power down the VMs so you can configure their shared storage. At this point, you should have folders on the VM host for your two cluster-node VMs. This is also a good time to configure a separate folder for the cluster's shared disk resources. This means you should have three folders: one for each VM and one for the shared storage. On our test Linux GSX Server, we have the three folder names—SharedDisks, W2K3-Node1, and W2K3-Node2—stored inside a folder named Cluster. You could use similar names for GSX Server running on a Windows host.

With the VMs ready to go, you now need to set up the shared disk resources.

To configure the shared storage resources, follow these steps:

1. Open the VMware Virtual Machine Console, and connect to the GSX Server host. Open the first VM node in the cluster, and then click the Edit Virtual Machine Settings link.
2. In the Virtual Machine Settings dialog box, click the Add button.
3. When the Add Hardware Wizard opens, click Next.
4. In the Hardware Types field, select Hard Disk and then click Next.
5. Select Create a New Virtual Disk, and click Next.
6. In the Select Disk Type dialog box, click the SCSI radio button and then click Next.
7. Enter a size for the virtual disk (at least 0.5GB), leave the Allocate Disk Space Now and Split Disk into 2GB Files boxes selected (for the best performance), and click Next.
8. You're now prompted to specify a disk file. In the Disk File field, click the Browse button and navigate to the SharedDisks folder. Now enter a name for the disk file in the File Name field (for example, enter **SCSI0-0**), and click Open.
9. Now click the Advanced button in the Specify Disk File dialog box.

10. In the Virtual Device Node drop-down menu, select SCSI0:0 Hard Disk 1. For each disk you add, you select the next available SCSI target (in other words, SCSI 0:1).
11. Once you've selected the SCSI target for the virtual disk, click Finish.
12. Repeat steps 2–11 to add more shared SCSI disks to the cluster.
13. Click OK to close the Virtual Machine Settings dialog box.

When you've finished adding the virtual hard disk files, each VM's configuration should look like Figure 11-4.

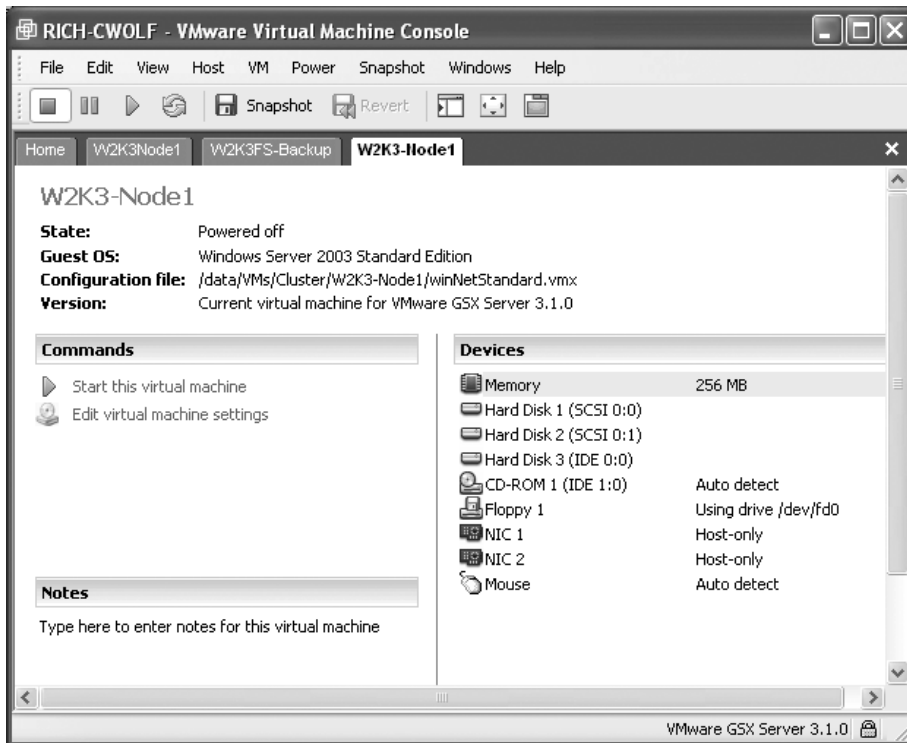


Figure 11-4. Clustered GSX Server VM hardware configuration

After creating the disks on the first node, you now need to add them to the second node's configuration.

Caution The virtual SCSI target settings used by the shared disks must be identical on each VM.

To do this, follow these steps:

1. Open the VMware Virtual Machine Console, and connect to the GSX Server host. Open the second VM node in the cluster, and then click the Edit Virtual Machine Settings link.
2. In the Virtual Machine Settings dialog box, click the Add button.
3. When the Add Hardware Wizard opens, click Next.
4. In the Hardware Types field, select Hard Disk and then click Next.
5. Select Use an Existing Virtual Disk, and click Next.
6. In the Select an Existing Disk dialog box, click the Browse button.
7. Navigate to the SharedDisks folder on the host system, and select the first available SCSI disk (for example, SCSI0-0.vmdk). Then click the Open button.
8. Now click the Advanced button.
9. In the Virtual Device Node drop-down menu, select the SCSI value that pertains to the virtual disk you're adding. For example, for the SCSI0-0 virtual disk, you'd select SCSI 0:0 from the menu. Once you've selected the SCSI address, click Finish.
10. Repeat steps 2–9 to add more shared SCSI disks to the cluster node.
11. Click OK to close the Virtual Machine Settings dialog box.

Once you've assigned the virtual disk files to each VM, you'll need to close the VMware application. This is necessary because each VM's configuration file will now require some manual editing. Once you've closed the VMware application, follow these steps:

1. Navigate to the first cluster VM's folder, and open its associated .vmx file in a text editor, such as Vi, gedit, or Notepad.
2. In the text editor, add the following line to the beginning of the configuration file:
`disk.locking = "False"`
3. Locate the lines that identify each shared virtual SCSI disk. Before the first SCSI disk definition, add the following line (refer to Figure 11-3 earlier):
`scsi0.sharedBus = "virtual"`
4. Save the .vmx file.
5. Repeat steps 1–4 for the second VM in the cluster.

At this point, you have all the hardware in place and configured to set up a Microsoft server cluster using the two VMs on GSX Server. For detailed procedures on installing the cluster service, turn to the “Installing the Cluster Service” section later in this chapter.

Configuring Clusters on Microsoft Virtual Server

Like VMware GSX Server, Microsoft Virtual Server 2005 also natively supports clustering. In this section, you'll see how to prepare two Windows Server 2003 VMs to run as a server cluster. Like with previous examples, you'll need to configure one node as a domain controller (or ensure the VMs can connect to another domain controller), join both VMs to the domain, and also configure a user account that's a member of the Domain Admins group for use by the Cluster Service.

At this point, we're assuming you have the two VMs configured with Windows Server 2003 Enterprise Edition operating systems installed. The VMs should have a single IDE virtual hard disk for the OS and installed applications, along with two virtual NICs. Figure 11-5 shows this starting configuration.

"W2K3-Node1" Configuration	
 <u>General properties</u>	"W2K3-Node1"
When Virtual Server starts:	Never automatically turn on virtual machine
When Virtual Server stops:	Save state
 <u>Virtual Machine Additions</u>	Virtual Machine Additions information not available
 <u>Memory</u>	128 MB
 <u>Hard disks</u>	1 virtual hard disk installed; Undo disks are disabled
Virtual hard disk 1	Attached to primary channel (0) Virtual hard disk file "W2K3-Node1.vhd" Maximum size is 4 GB; Currently expanded to 10.5 KB
 <u>CD / DVD</u>	1 virtual CD / DVD drive installed
Virtual CD / DVD drive 1	Attached to secondary channel (0) Host drive "F"
 <u>SCSI adapters</u>	No virtual SCSI adapters installed
 <u>Network adapters</u>	2 virtual network adapters installed
Virtual network adapter 1	Connected to "Internal Network" Current Ethernet (MAC) address: 00-03-FF-07-CC-AD
Virtual network adapter 2	Connected to "Internal Network" Current Ethernet (MAC) address: 00-03-FF-04-CC-AD
 <u>Scripts</u>	Scripts disabled
 <u>Floppy drive</u>	No media captured
 <u>COM ports</u>	2 COM ports installed
COM port 1	Attached to none
COM port 2	Attached to none
 <u>LPT ports</u>	1 LPT port installed
LPT port 1	Attached to none

Figure 11-5. Virtual Server VM initial configuration

Note At the time of publication, Virtual Server 2005 supported only one virtual hard disk on a shared SCSI bus. This means to have more than one shared SCSI disk in the cluster, you'll need to configure at least two virtual SCSI adapters per node.

To complete the hardware configuration for the cluster VMs, both VMs should be powered down. With both machines off, your first task will be to add a virtual SCSI adapter to them.

Adding a Virtual SCSI Adapter

To add a virtual SCSI adapter, perform these steps on each VM:

1. From the Virtual Server Master Status Web page, under the Virtual Machines section, mouse over the Configure link and select the first cluster-node VM.
2. From the VM Configuration page, scroll down and click the SCSI Adapters link.
3. At this point, you should be on the SCSI Properties page. Now click the Add SCSI Adapter button.
4. As shown in Figure 11-6, check the Share SCSI Bus for Clustering box, leave the SCSI adapter ID set to 7, and click the OK button.
5. If you plan on having more than one shared disk in the cluster, then repeat steps 1–4 to add a second virtual SCSI adapter to the VM.
6. The SCSI adapter should now appear on the VM's configuration page. Repeat steps 1–5 on the second cluster VM. When adding the virtual SCSI adapter to the second VM, set the adapter's SCSI ID to 6.

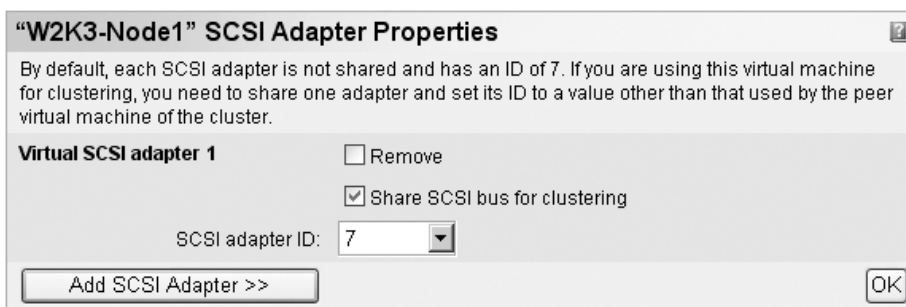


Figure 11-6. *Creating a SCSI adapter with a shared SCSI bus*

With the virtual SCSI adapter in place, you can now create a virtual hard disk to attach to the adapter.

Creating and Configuring the Shared Virtual Hard Disks

A good practice with shared disks is to store them in their own folder. With this in mind, we used the following folders for the test Windows 2003 cluster: W2K3-Node1, W2K3-Node2, and SharedDisks. If you don't have a SharedDisks folder created, open Windows Explorer and create one before proceeding.

Here are the steps to create a shared virtual SCSI disk:

1. From the Virtual Server 2005 Master Status page, under the Virtual Disks section, mouse over the Create link and select Fixed Size Virtual Hard Disk. Note that for the trade-off of saving disk space over performance, you could also select Dynamically Expanding Virtual Hard Disk for this step.
2. You're now asked to specify the name and location for the hard disk. As shown in Figure 11-7, enter the complete path to the new disk file in the Virtual Hard Disk File Name field. In our example, we assigned the new disk the name SCSI0-0.vhd.

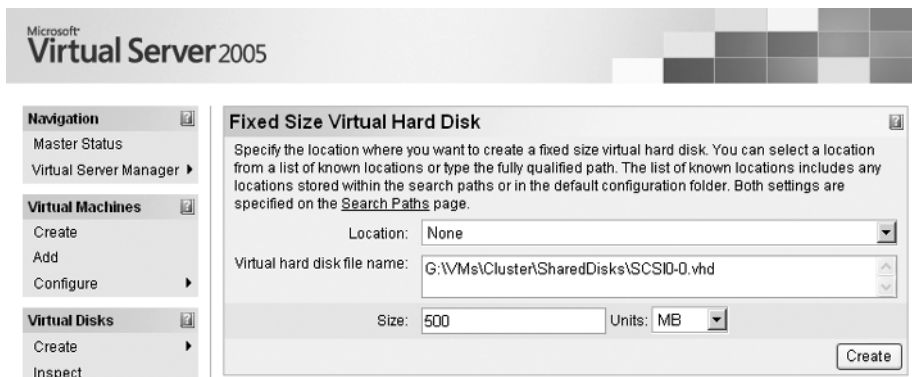


Figure 11-7. *Creating a virtual hard disk file*

3. Now set the disk size to the size you want, which should be at least 500MB, and then click the Create button.
4. You'll be returned to the Master Status page, and you should see an event stating the virtual disk was created in the Recent Events section of the page.

Once you've created a virtual SCSI hard disk, you can now add it to the configuration of each cluster-node VM. To do this, follow these steps:

1. From the Virtual Server Master Status Web page, under the Virtual Machines section, mouse over the Configure link and select the first cluster-node VM.
2. From the VM Configuration page, scroll down and click the Hard Disks link.
3. Now from the Virtual Hard Disk Properties page, click the Add Disk button.

4. You should now see fields appear for an additional virtual hard disk. With the default single IDE virtual hard disk configured initially, you'll see a section for Virtual Hard Disk 2. This is where you'll assign the first shared virtual disk to the VM. To do so, select the SCSI 0 ID 0 selection from the Attachment drop-down menu.
5. Enter the full path to the shared disk file in the Fully Qualified Path to File field. For example, on our test setup, the path to the shared disk file is `G:\VMs\Cluster\SharedDisks\SCSI0-0.vhd`.
6. Once finished entering the shared disk information, click the OK button. Figure 11-8 shows our test configuration, with a single IDE virtual disk for the OS, along with shared SCSI virtual disks for the cluster.

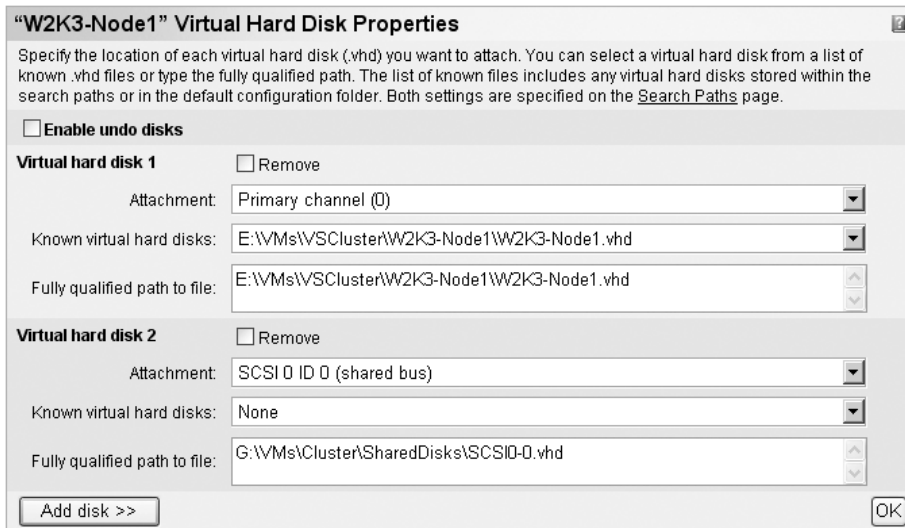


Figure 11-8. Assigning a shared virtual hard disk to a VM

7. Repeat steps 1–6 on the second cluster node.

With the shared virtual disks configured, you can now install the Cluster Service. You can find the steps for doing this later in this chapter in the “Installing the Windows Server 2003 Cluster Service” section.

Setting Up iSCSI Windows Server Clusters

iSCSI provides additional possibilities when configuring Windows VM clusters. With traditional emulated shared SCSI buses on VM applications, server clusters have always been limited to two nodes. With iSCSI, this is no longer the case. If the host system can support it, you can scale a cluster up to eight nodes. Also, with iSCSI, you can build a server cluster with any VM application that supports virtual SCSI disks.

In a VM environment, the easiest method to build an iSCSI cluster is to dedicate a Windows Server VM as the iSCSI target. This will allow all nodes in the cluster to access the shared storage over a dedicated storage LAN by connecting to the designated iSCSI target VM using the iSCSI protocol. In the following sections, we'll cover the steps you can follow on any VM application to build the iSCSI server cluster shown in Figure 11-9.

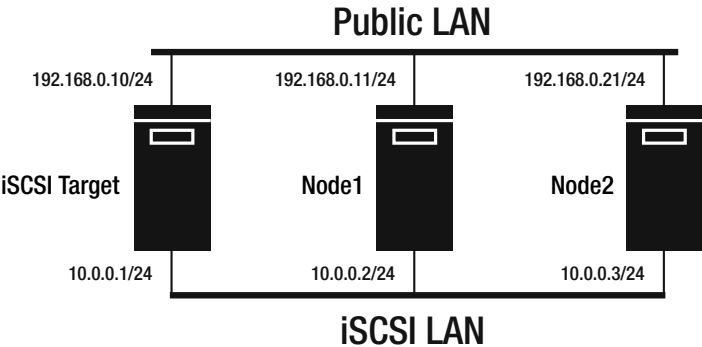


Figure 11-9. iSCSI VM cluster

With the ultimate goal in mind, you'll now start setting up the initial VM configuration.

Preparing for the VM Configuration

To configure the iSCSI cluster, you'll need at least three VMs (for a two-node cluster). One VM will act as the iSCSI target and will host the cluster's shared storage, and the other two VMs will serve as nodes in the cluster. You should create all VMs with two virtual NICs. The cluster-node VMs will need a single virtual hard disk for their OS and application installations, and the iSCSI target VM will need additional SCSI hard disks to share as iSCSI disk resources.

Once you've configured the VMs, you'll then need to power them up and install the Windows Server 2003 Enterprise Edition operating system. Remember that as a shortcut, you could configure a single VM, Sysprep it, and then copy the Sysprepped VM to additional folders to use as the remaining VMs in the cluster. This will prevent you from having to perform multiple OS installations.

With the operating systems installed, your next step is to configure the network settings for each of the two virtual NICs on each VM. Each NIC should appear as Local Area Connection 1 and Local Area Connection 2. For ease of configuration and administration, a best practice is to rename each connection after its intended purpose. In our lab, we renamed the Local Area Connection 1 NIC to Public and the Local Area Connection 2 NIC to iSCSI&Heartbeat. With the network connections renamed, you can then move onto configuring their TCP/IP settings. Table 11-1 lists the settings we used in our test example.

Table 11-1. *iSCSI Cluster Network Settings*

Host/NIC	IP Address	Subnet Mask	Gateway	DNS
iSCSI target: Public	192.168.0.10	255.255.255.0	192.168.0.1	192.168.0.10
iSCSI target: iSCSI&Heartbeat	10.0.0.1	255.255.255.0		
Node1: Public	192.168.0.11	255.255.255.0	192.168.0.1	192.168.0.10
Node1: iSCSI&Heartbeat	10.0.0.2	255.255.255.0		
Node2: Public	192.168.0.12	255.255.255.0	192.168.0.1	192.168.0.10
Node2: iSCSI&Heartbeat	10.0.0.3	255.255.255.0		

With the network interfaces configured, it's now a good time to verify that each VM cluster node can ping both IP addresses of the iSCSI target. Listing 11-1 shows an example of executing ping to test connectivity from one cluster node.

Listing 11-1. *Pinging the iSCSI Target*

```
C:\>ping 192.168.0.10
```

```
Pinging 192.168.0.10 with 32 bytes of data:
```

```
Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 192.168.0.10:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0 % loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ping 10.0.0.1
```

```
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 10.0.0.1:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0 % loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>
```

With the general VM settings complete and the network settings configured and tested, it's now time to perform specific configurations on each VM. Let's start with the iSCSI target.

Configuring the iSCSI Target

The first order of business with the iSCSI target is to configure its additional virtual hard disks to be used by the iSCSI service. Using Disk Management to format and partition each drive can do this. Since Windows clustering requires Active Directory, it's good practice to configure the iSCSI target to also act as a domain controller.

Installing Active Directory on the iSCSI Target

You can configure the target as a domain controller by running the `dcpromo` command. When you run `dcpromo`, select the following options for the Active Directory configuration:

- Set the domain controller to host a new domain.
- Select for the domain to exist in a new forest.
- Enter a unique domain name (for example, enter **apress.com**).
- Accept the default NetBIOS name.
- Accept the default database and log file locations.
- Accept the default shared system volume location.
- Select the option to install and configure DNS on this computer.
- Select all the remaining setup defaults.

Once the installation completes and the domain controller reboots, it's a good time to go ahead and create the Cluster Service user account. Open the Active Directory Users and Computers administrative tool, and create a user account (we prefer to name this account *cluster*). Add the cluster user account to the Domain Admins user group, and then close Active Directory Users and Computers.

Installing the iSCSI Target Software

With Active Directory configured and the Cluster Service account created, the last step in setting up the iSCSI target is to install the iSCSI target software. Rocket Division Software (<http://www.rocketdivision.com>) offers an excellent iSCSI target application; you can find a full version of it on the book's companion CD in the `RocketDivision\StarWind` folder.

To install the StartWind software, follow these steps:

1. Copy `StarWind.exe` to the iSCSI target VM.
2. Double-click the `StarWind.exe` file.
3. When the setup wizard opens, click Next.
4. Accept the license agreement, and click Next.
5. Use the default installation location, and click Next to continue.

6. Select the Full Installation option as shown in Figure 11-10, and click Next.

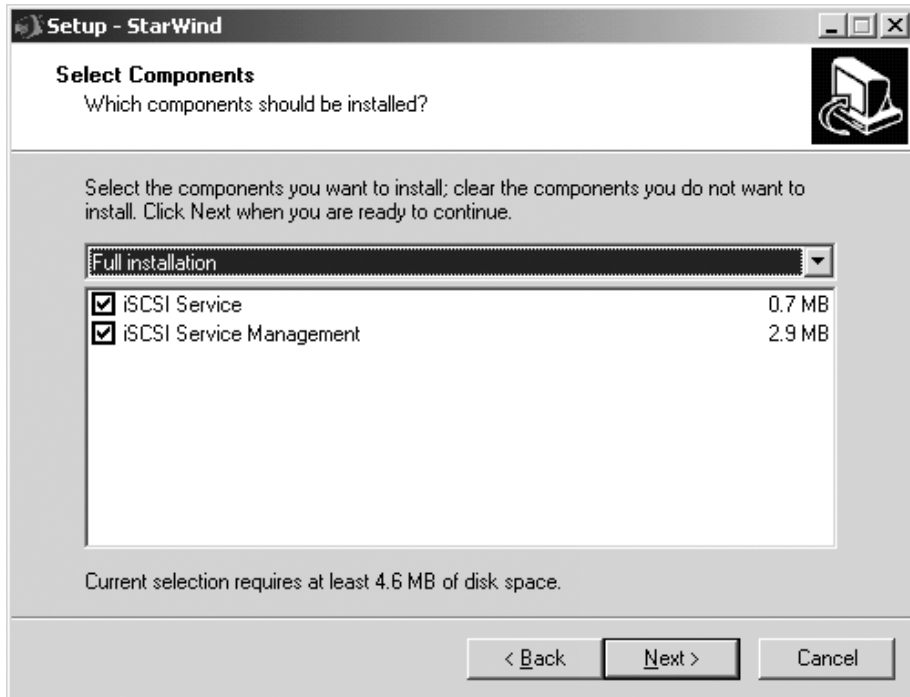


Figure 11-10. *Selecting StarWind's full installation option*

7. Use the default Start Menu folder location, and click Next.
8. In the Additional Tasks dialog box, check the Create a Desktop Icon box, and if desired, clear the German and Spanish Additional Languages boxes. Then click Next.
9. Click Install to install the StarWind iSCSI software.
10. When the installation completes, click Finish.

Configuring StarWind

Once you've installed the StarWind software, you need to configure its default password so you can access it through its management UI. To do this, follow these steps:

1. In Windows Explorer, browse to the C:\Program Files\Rocket Division Software\StarWind folder.
2. Double-click the starwind.conf file.
3. When prompted by the OS to select how you want to open the file, click the Select the Program from a List radio button, and click OK.

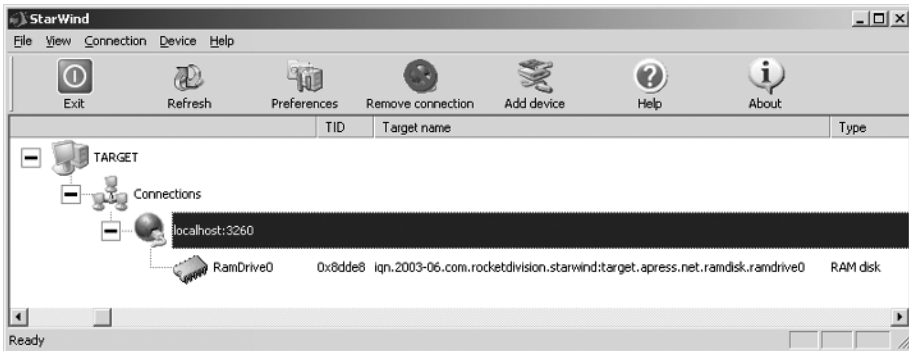


Figure 11-12. Initial StarWind iSCSI service connection

5. Notice that by default, a shared RamDrive is configured. This won't be needed for clustering, so right-click the RamDrive0 object and select Remove. When prompted to confirm the device removal, click Yes.
6. Now with the localhost object selected in the StarWind UI, click the Add Device icon on the toolbar.
7. When the Add Device Wizard opens, click Next.
8. Now in the Device Type Selection dialog box, click the SPTI Device radio button and click Next.
9. As shown in Figure 11-13, select the PhysicalDrive1 device and click Next.

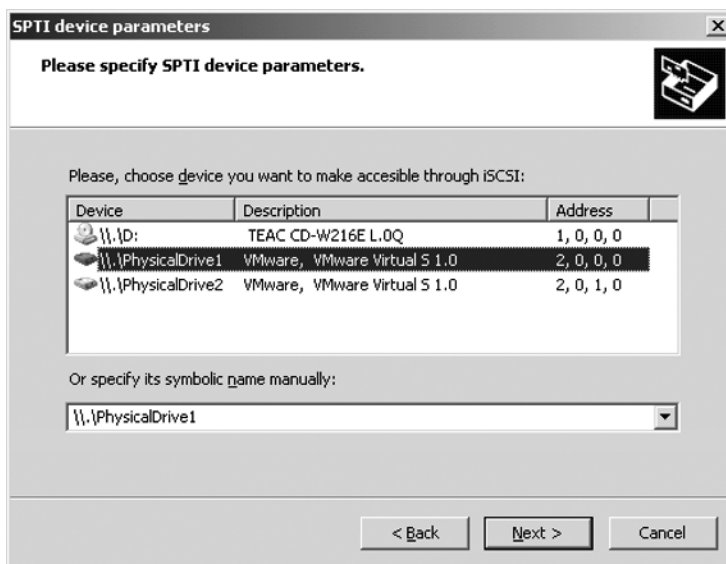


Figure 11-13. Selecting the first physical drive to share

10. Now enter **Quorum** for the name of the device, and click Next.
11. In the Completing the Add Device Wizard window, verify that PhysicalDisk1 is listed and has the network name Quorum and then click Next.
12. Click Finish to close the Add Device Wizard.
13. Repeat steps 6–12 to share the PhysicalDisk2 resource. This time, enter **Data** as the share name in step 10.

When finished, your configuration should appear similar to what's shown in Figure 11-14.

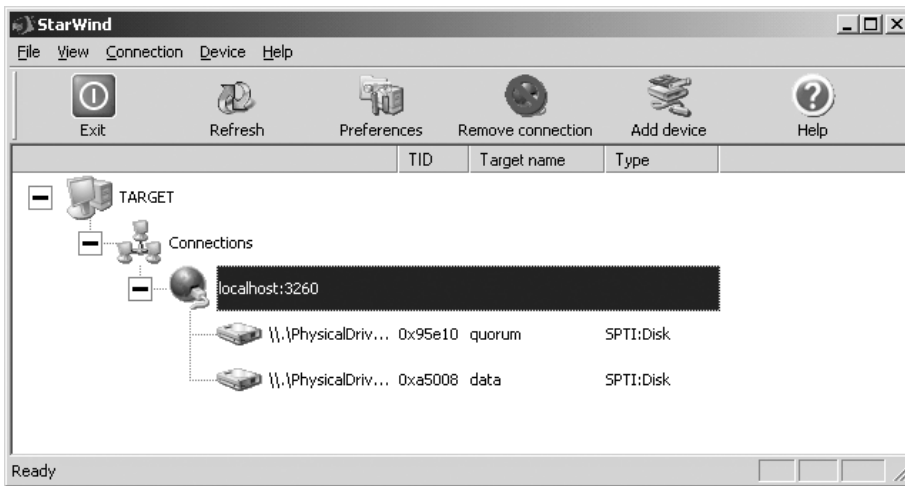


Figure 11-14. iSCSI target with two configured disks

With the disks configured, you now need to configure the connection security settings for the iSCSI clients. To do this, follow these steps:

1. In the StarWind UI, right-click the localhost object and select Permissions.
2. In the Permissions dialog box, click the Add button.
3. You now need to enter at a minimum a local name and local secret for the connection. As shown in Figure 11-15, we entered the following settings for Node1:
 - **Local Name:** Node1
 - **Local Secret:** ClusteringIsFun
 - **Peer Name:** node1.apress.net

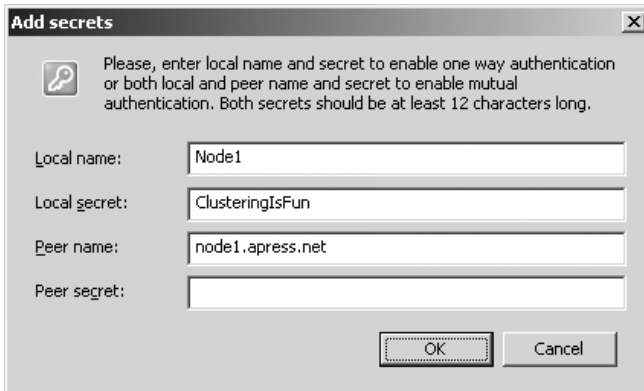


Figure 11-15. *Setting Node1 permissions*

4. When finished entering the name and secret settings, click OK.
5. Repeat steps 3–4 for Node2. When finished, the Permissions dialog box should appear similar to what's shown in Figure 11-16.

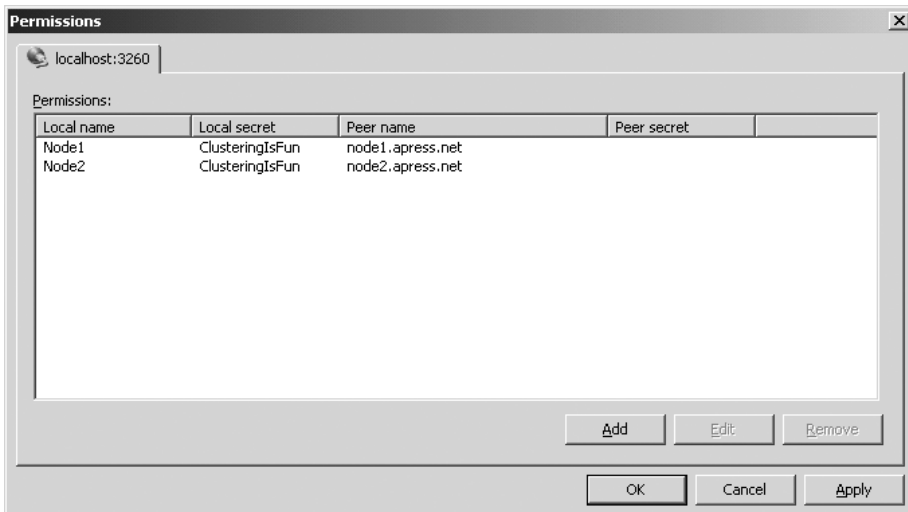


Figure 11-16. *Final permissions settings for both nodes*

6. Click OK to close the Permissions dialog box.

With the iSCSI target now set up, you need to make one additional change to its configuration, which will allow multiple concurrent connections to access shared resources. Unfortunately, you can't make this change through the GUI tool, so you'll again need to open the C: \Program Files\Rocket Division Software\StarWind\starwind.conf file in Notepad. With the file open, scroll down toward the bottom of the file until you see the line add "\\.\PhysicalDrive1" "Quorum". Add the following statement to this line: -share:"rw" -sessions:8. This will allow up to eight clients to access the volume, and all will have read and write access. You'll need to add the same statement to the next line of the starwind.conf file so that the line reads "\\.\PhysicalDrive2" "Data" -share:"rw" -sessions:8. Figure 11-17 shows these changes.

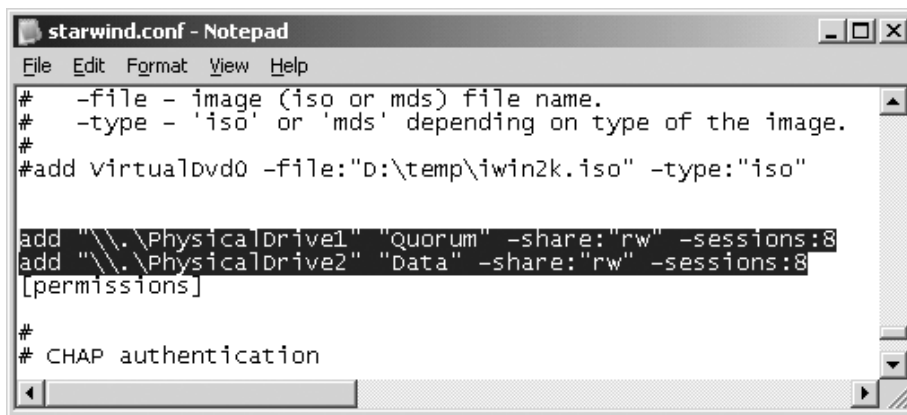


Figure 11-17. *Configuring physical drives to be shared*

At this point, the iSCSI target is ready to go. Now you can configure the cluster nodes. Before proceeding, be sure to restart the StarWind iSCSI service to apply the manual changes made to the shared disks.

Preparing the Cluster Nodes

With the iSCSI target ready to go, you can now configure the two cluster nodes as clients. Prior to doing so, if you have yet to join each cluster-node VM to the domain configured on the iSCSI target, do so now. Once you join them to the domain, you can then set up each node to access the iSCSI storage.

To set up the VM cluster nodes for iSCSI access, you need to install the iSCSI initiator software. You can download this software for free from Microsoft at <http://www.microsoft.com/windowsserversystem/storage/technologies/iscsi/default.mspx>. At the time of publication, the latest x86 version of the Microsoft SCSI initiator software is 1.06-initiator-x86fre.msi. Once you've downloaded the software, you'll then need to copy it to each cluster-node VM. With the initiator software on each cluster node, follow these steps to install the software:

1. On the Node1 VM, locate the copied SCSI initiator software, and double-click the initiator MSI installation file.
2. When the iSCSI Initiator Setup Wizard opens, click Next.
3. Use the default software installation location provided, select to install the software for Everyone, and click Next.
4. You should now be prompted to confirm the installation. Click Next.
5. In the License Agreement dialog box, click the I Agree radio button and click Next.
6. In the Microsoft iSCSI Installation Program dialog box, click the Install Complete iSCSI Initiator radio button and click OK.
7. In the EULA dialog box, click Agree.
8. When prompted that the initiator was installed successfully, click OK.
9. When the installation information window opens, click Next.
10. Click Close to close the window.
11. Repeat steps 1–10 on Node2 to install the initiator service.

At this point, you're now ready to configure the iSCSI initiator on each node. Let's start with Node1. To configure Node1 to connect to the shared iSCSI storage, follow these steps:

1. On Node1, double-click the Microsoft iSCSI initiator shortcut on the desktop.
2. In the iSCSI Initiator Properties window, click the Target Portals tab.
3. Now click the Add button.
4. In the Add Target Portal dialog box, enter **10.0.0.1** in the IP Address or DNS Name field and then click the Advanced button (see Figure 11-18). Remember that 10.0.0.1 is the IP address of the iSCSI target.

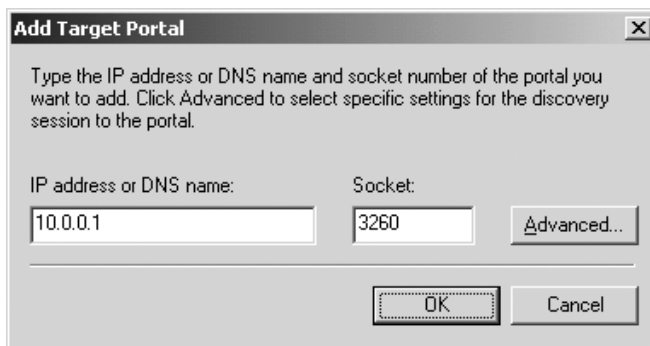


Figure 11-18. *Adding an iSCSI target portal*

5. In the Advanced Settings dialog box, check the CHAP Logon Information box. Then enter the node name as the username, and enter the secret that was used in step 3 of the StarWind iSCSI target configuration in the Target Secret field. Figure 11-19 shows these settings.

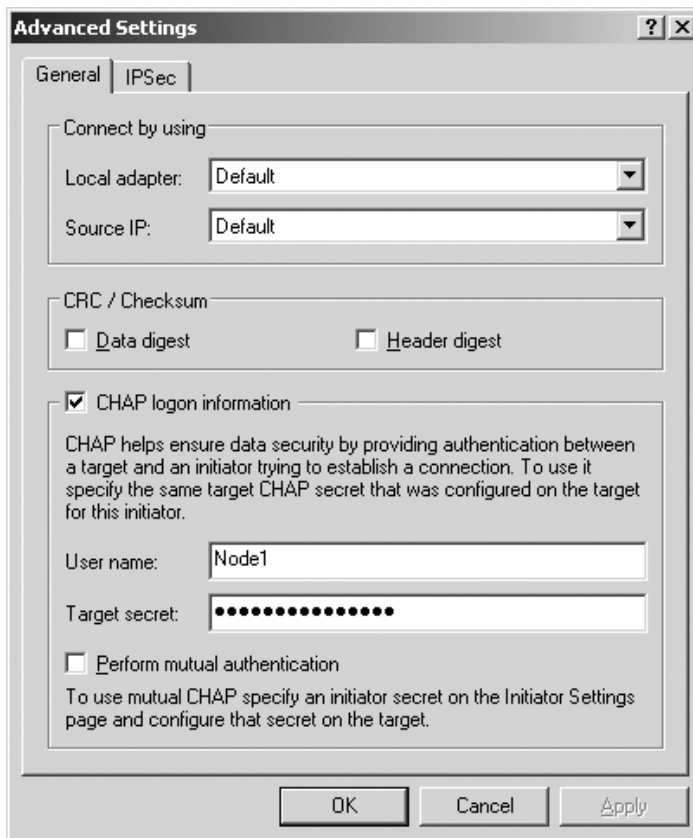


Figure 11-19. *Configuring CHAP authentication*

6. When finished entering the CHAP settings, click OK.
7. Click OK in the Add Target Portal dialog box.
8. You should now see the 10.0.0.1 target listed as an available portal. Click the Available Targets tab.
9. Under the Available Targets tab, click the Data target and then click the Log On button.
10. In the Log On to Target dialog box, check the Automatically Restore This Connection When the System Boots box and then click the Advanced button.

11. In the Advanced Settings dialog box, you need to perform the same steps you did in step 5 earlier. Check the CHAP Logon Information box. Then enter the node name as the username, and enter the secret that was used in step 3 of the StarWind iSCSI target configuration in the Target Secret field.
12. Repeat steps 9–11, this time selecting the Quorum target.
13. The Available Targets tab in the iSCSI Initiator Properties window should now look like Figure 11-20.

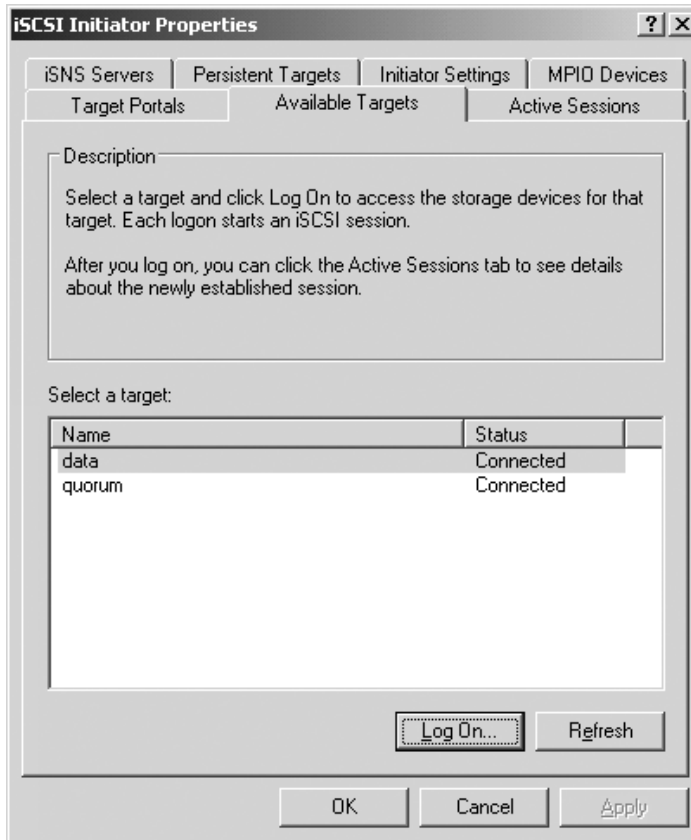


Figure 11-20. Properly configured iSCSI target disks

14. Click OK to close the iSCSI Initiator Properties window.

At this point, the cluster node should now be able to see the iSCSI shared storage. For Node2, you now need to repeat the previous 14 steps to connect it to the shared storage.

With the shared storage now configured, all that's left is to configure drive letters for the shared storage on each cluster node. To do this, follow these steps:

1. On Node1, click Start ► Administrative Tools ► Computer Management.
2. In the Computer Management MMC, click Disk Management.
3. In the upper-right pane of the window, right-click the Data volume, and select Change Drive Letter and Paths.
4. In the Change Drive Letter and Paths for Data dialog box, click the Add button.
5. In the Add Drive Letter or Path dialog box, select R in the Assign the Following Drive Letter drop-down menu and click OK.
6. Right-click the Quorum volume, and select Change Drive Letter and Paths.
7. In the Change Drive Letter and Paths for Data dialog box, click the Add button.
8. In the Add Drive Letter or Path dialog box, select Q in the Assign the Following Drive Letter drop-down menu and click OK.
9. Repeat steps 1–8 on Node2 to assign the same drive letters to the shared iSCSI volumes.

With the shared volumes and cluster nodes configured, you're now ready to install the Cluster Service.

Caution The iSCSI target VM must be fully booted up before starting the cluster-node VMs. Otherwise, you may need to restart the Microsoft iSCSI initiator service on each cluster node in order to connect to the shared iSCSI storage.

Installing the Windows Server 2003 Cluster Service

In this section, we'll cover the general procedure for configuring the Windows Server 2003 Cluster Service in a two-node VM cluster configuration. These procedures are generic in format, so they will apply to any VM application.

Although these procedures cover the general installation steps, they assume the following clustering prerequisites have been met:

- Domain controller configured and online
- Cluster user account that's a member of the Domain Admins global group or is at a minimum set as a local administrator on each cluster node
- Shared storage configured and online
- Cluster public and heartbeat networks configured

With the prerequisites met, you're now ready to install the Cluster Service.

Caution Microsoft strongly recommends only one node be online during the installation and initial configuration of the first cluster node. Once you've configured the first node in the server cluster, then power up each subsequent node and join it to the cluster.

To install the cluster service, follow these steps:

1. On Node1, click Start ► Administrator Tools ► Cluster Administrator.
2. When Cluster Administrator opens, you should see the Open Connection to Cluster dialog box appear. In this dialog box, select Create New Cluster in the Action menu and click OK.
3. When the New Server Cluster Wizard opens, click Next.
4. Enter a name for the cluster in the Cluster Name field, and click Next.
5. In the Computer Name field, leave the default name selected and click Next.
6. Cluster Administrator will now analyze the cluster configuration. If you completed all the previous steps successfully, you should see nothing but checks in this window. When the analysis completes, click Next.
7. In the IP Address field, enter an IP address for the cluster. In our example, we used 192.168.0.20. Once you've entered an address, click Next.
8. Now enter the username and password of the Cluster Service user account (created earlier), and click Next.
9. View the settings in the Proposed Cluster Configuration dialog box, and click Next to create the cluster. If you want to specify the disk to serve as the quorum disk, click the Quorum button at this time and then select the appropriate drive letter.
10. This wizard will now create the cluster. When it finishes, click Next.
11. Click Finish to close the New Server Cluster Wizard window.

You should now see the cluster shown in Cluster Administrator, with Node1 listed as the only cluster node. Perform these steps to add Node2 to the cluster:

1. Power up the Node2 VM. Once the VM finishes booting, in the Cluster Administrator MMC on Node1, right-click the cluster object, select New, and then click Node.
2. When the Add Nodes Wizard opens, click Next.

3. You should now see the Select Computers window. In the Computer Name field, enter **Node2** (or whatever you set as the host name of Node2), and then click the Add button. Then click Next.
4. Once the wizard finishes analyzing the cluster configuration, click Next.
5. Enter the password for the cluster service user account, and click Next.
6. Verify that the cluster configuration settings are correct, and click Next.
7. The wizard will now add Node2 to the cluster. When it finishes, click Next.
8. Click Finish to close the Add Nodes Wizard.

You should now see both Node1 and Node2 displayed in the cluster's configuration. At this point, the cluster is now online and ready for anything you can throw at it. Remember that with a shared iSCSI bus, you can build a server cluster with up to eight VM nodes, while shared SCSI buses support only up to two nodes. The only reason you can't add more than eight nodes with iSCSI is that this is the maximum number of cluster nodes supported by Microsoft Windows Server 2003 in a server cluster. iSCSI has no such limitation.

Now that you've seen a few ways to build virtualized server clusters, you'll look at the steps for setting up a load-balanced cluster on virtual machines.

Setting Up Windows NLB Clusters

Windows NLB cluster configuration is a relatively simple process. Since each cluster node in an NLB cluster maintains its own local storage, you don't have to worry about configuring shared SCSI virtual storage. Instead, your only real concern is setting up the network so that each cluster node can communicate with all other nodes. This basically means you can quickly configure an NLB cluster using any VM application.

In this section, we'll show how to build the NLB cluster shown in Figure 11-21.

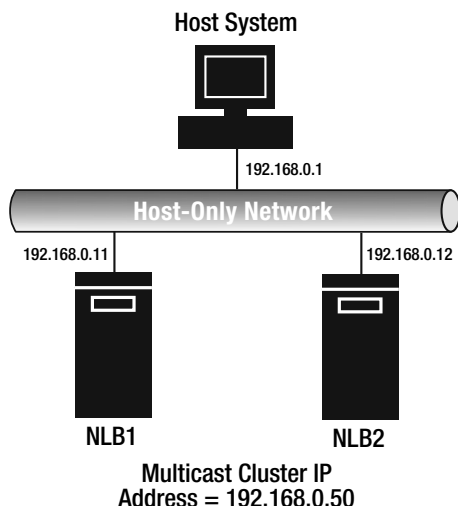


Figure 11-21. *Two-node NLB VM cluster*

Note that the cluster connects to the host system on the 192.168.0.x/24 host-only network. With a single virtual NIC in each VM, the NLB service will be configured as a multicast cluster.

Domain membership isn't required for NLB clustering, so each cluster node can simply have a base Windows Server 2003 installation and be configured in the same workgroup. With this as the starting point, you'll then need to configure the initial TCP/IP settings of each cluster node. In our example, the host system provided the DNS services and also acted as the default gateway. With this in mind, we configured the following TCP/IP settings on each VM:

- **Node1 settings:**
 - **IP address:** 192.168.0.11
 - **Subnet mask:** 255.255.255.0
 - **Default gateway:** 192.168.0.1
 - **DNS:** 192.168.0.1
- **Node2 settings:**
 - **IP address:** 192.168.0.12
 - **Subnet mask:** 255.255.255.0
 - **Default gateway:** 192.168.0.1
 - **DNS:** 192.168.0.1

After configuring the network settings of each node, verify connectivity with the host by pinging the IP address of each VM from the host system. Once you've established that the network settings are good to go, your final task in configuring the NLB cluster is to set up the cluster settings. The easiest way to do this is by using the Network Load Balancing Manager administrative tool.

Caution Unlike with server clusters, all NLB cluster nodes must be online when the service is configured using the Network Load Balancing Manager.

Follow these steps to use this tool and configure the cluster:

1. Log onto one of the VMs to be clustered.
2. Click Start ► Administrative Tools ► Network Load Balancing Manager.
3. When the Network Load Balancing Manager opens, click the Cluster menu and then click New.

4. In the Cluster Parameters dialog box, enter the IP address, subnet mask, and Internet name (FQDN) for the cluster (see Figure 11-22). Then click the Multicast radio button, and click Next.

Figure 11-22. *Configuring NLB cluster parameters*

5. Now you have the opportunity to add IP addresses to the cluster. Don't enter anything at this point, and click Next.
6. In the Port Rules dialog box, leave the default settings, and click Next.
7. In the Connect dialog box, enter the host name for the first VM in the cluster (for example, enter **NLB1**) and click the Connect button.
8. The node's network connection should now be displayed in the lower portion of the window. Click the connection, and then click Next.
9. In the Host Parameters dialog box, leave the default settings intact, and click Finish.
10. With the first node configured in the cluster, all that's left is to add the second node. To do this, right-click the newly created cluster object and select Add Host to Cluster.
11. In the Connect dialog box, enter the host name for the second VM in the cluster (for example, enter **NLB2**) and click the Connect button.
12. The node's network connection should now be displayed in the lower portion of the window. Click the connection, and then click Next.
13. In the Host Parameters dialog box, leave the default settings intact, and click Finish.

Both nodes should now appear as converged in the Network Load Balancing Manager GUI. With the nodes configured in the cluster, you can now configure additional services such as IIS so the cluster can balance a load of Web requests. To test, you should be able to ping the IP address of the cluster from the host system.

Building Linux VM Clusters

With the abundance of Linux distributions currently available, documenting a process to build clusters that will work on both Red Hat and SuSE, for example, is difficult at best. Both Red Hat and SuSE offer their own proprietary clustering solutions that are offshoots of LVS and Linux-HA. Each of these projects offers open-source code that can be installed and configured in virtual machines. For load balancing using LVS, you can find an abundance of information at <http://www.lvs.org>. Information on high-availability failover clusters is available at <http://www.linux-ha.org>.

Under most circumstances, Linux VM clusters serve as the perfect testing and training source for production Linux clustering solutions. This means that if you run a commercial product such as the Red Hat Cluster Suite in production, you can install the same software into a Red Hat VM, using the documentation and installation steps provided by Red Hat.

Since we don't know which Linux distribution you have, in this section we'll show you how to configure simple VM clusters using open-source software from the Linux-HA and LVS projects. We selected to perform our test cluster installations on Red Hat Enterprise Advanced Server 3.0 because of its majority of Linux market share.

In this section, we'll show how to build a failover cluster or LVS cluster using the Linux-HA software available at <http://www.ultramoney.org>. UltraMonkey provides the software and documentation of Linux-HA on Red Hat distributions. Since this particular cluster format doesn't incorporate the use of shared storage, you don't have to do any preparation to get the VMs in the cluster to share any virtual hard disks. This means you can use the process explained in this section on any VM application such as VMware or Microsoft Virtual Server.

Figure 11-23 shows the cluster you'll set up in this section.

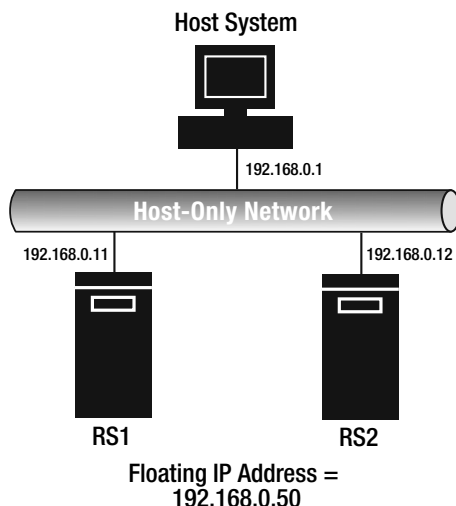


Figure 11-23. Two-node Linux high-availability cluster

Before installing the high-availability software, you must first configure each VM according to the settings in Figure 11-23. Before starting, you should have a VM named RS1 and another named RS2 (or whatever your heart desires). Next you'll need to make sure the proper TCP/IP settings are configured on each VM. With the VMs configured, you'll then need to connect them to the Internet to download the UltraMonkey software from <http://www.ultramoney.org>. You'll find a single folder available on the Web site to download all the Red Hat installation RPMs. Download each available RPM for your Red Hat distribution before continuing.

Once you have the available software, you're ready to set up the VM cluster. The first step in the installation process is to upgrade `mkinitrd`. To do this, follow these steps:

1. Assuming you're logged on and have downloaded the RPMs as root, change to the `/root` directory.
2. Next, use RPM to install the latest `mkinitrd` package. For example, use `rpm -hFv mkinitrd-3.5.13-1.um.1.i386.rpm`.

With `mkinitrd` upgraded, the next step is to upgrade the kernel. To do this, download the kernel RPM that's appropriate for the host hardware architecture and then use RPM to install the kernel upgrade. Here are examples for different host architectures:

- **AMD Athlon:** `rpm -Fhv kernel*2.4.21-20.EL.um.1.athlon.rpm`
- **i686:** `rpm -Fhv kernel*2.4.21-20.EL.um.1.i686.rpm`
- **i586:** `rpm -Fhv kernel*2.4.21-20.EL.um.1.i586.rpm`

With the kernel upgraded, you'll need to reboot the Linux VM for the changes to be applied, but prior to doing so you can use the RPM to install the remaining packages for running and configuring LVS and Linux-HA on the VMs.

Note The remaining packages to be installed depend on the following packages: `perl-Digest-HMAC`, `perl-Digest-SHA1`, and `perl-Parse-RecDescent`. You can download these packages from <http://www.rpmsfind.net> and then install them using the `rpm -Uvh` command.

You can now install the remaining configuration and management packages that were downloaded from UltraMonkey. These packages include the following:

- `heartbeat-1.0.4-2.rh.el.um.1.i386.rpm`
- `heartbeat-lldirectord-1.0.4-2.rh.el.um.1.i386.rpm`
- `heartbeat-pils-1.0.4-2.rh.el.um.1.i386.rpm`
- `heartbeat-stonith-1.0.4-2.rh.el.um.1.i386.rpm`
- `ipvsadm-1.21-1.rh.el.1.i386.rpm`
- `libnet-1.1.0-1.rh.el.1.i386.rpm`

- perl-Authen-SASL-2.03-1.rh.el.um.1.noarch.rpm
- perl-Convert-ASN1-0.16-2.rh.el.um.1.noarch.rpm
- perl-IO-Socket-SSL-0.92-1.rh.el.um.1.noarch.rpm
- perl-Mail-IMAPClient-2.2.7-1.rh.el.um.1.noarch.rpm
- perl-Net-SSLeay-1.23-1.rh.el.um.1.i386.rpm
- perl-Parse-RecDescent-1.80-1.rh.el.um.1.noarch.rpm
- perl-XML-Namespacesupport-1.08-1.rh.el.um.1.noarch.rpm
- perl-XML-SAX-0.12-1.rh.el.um.1.noarch.rpm
- perl-ldap-0.2701-1.rh.el.um.1.noarch.rpm

You can install all the previously listed RPMs collectively by using a single `rpm -Uvh` command and including each package as a parameter in the command syntax. Note that the package versions listed are the most recent at the time of book publication and may have different version numbers by the time you download them.

Once all the packages are installed, run the `reboot` command to reboot the VM. After the VM reboots, you'll need to copy the following files from `/usr/share/doc/heartbeat-1.0.4` to the `/etc/ha.d` folder: `ha.cf`, `haresources`, and `authkeys`.

With these files in place, your last option is to configure the Node parameters in the `ha.cf` and `haresources` files. You must also set the `authkeys` file to mode 600. To do this, run the command `chmod 600 /etc/ha.d/authkeys`. Finally, you can enable the heartbeat by running `/etc/init.d/heartbeat start`.

At this point, you have a working Linux failover cluster!

Note Remember, Chapter 9 covers failover clustering in detail. For more information on specific Linux-HA cluster configuration, turn to Chapter 9.

Summary

In this chapter, you looked at an abundance of cluster configurations inside VMs. While we covered configurations that are either available with their respective operating systems or free on the Internet, remember that many organizations use third-party clustering applications. With this in mind, to learn and practice clustering, a good practice is to install your organization's specific products inside the cluster VMs. Of course, if you're just looking to learn and play, we've provided plenty to keep you busy in this chapter.

Clusters and storage resources have much more potential when they're attached to storage networks. You'll look at how you can achieve this potential in Chapter 12.



Introducing Storage Networking

It's tough to understand the concepts that drive storage virtualization without a handle on storage networking. With this in mind, we decided to leave nothing to chance; in this chapter, we'll provide an overview of the many technologies driving today's storage networks.

Storage networks are designed and operate similarly to Ethernet networks, with the exception that instead of interconnecting hosts to hosts, hosts are connected to a shared network of storage devices. With storage networking, a high-speed data network is dedicated to storage devices. Why do this, you ask? We think that the real question is, why didn't we think of storage networking sooner?

Several factors have contributed to the growth of today's storage networks:

- Sheer volume of data, coupled with exponential data growth
- High cost of storage resources
- Organizational focus on data protection and recovery

The bottom line with storage is that today the typical organization has data in the high gigabyte, terabyte, or even petabyte range. To put these values in perspective, Table 12-1 shows the number of megabytes in a gigabyte, terabyte, and petabyte.

Table 12-1. *Byte Conversion Comparison*

Data Size	Equal To...
1GB	1,024MB
1TB	1,024GB or 1,048,576MB
1PB	1,024TB or 1,048,576GB, or 1,073,741,824MB

With the typical IT department likely managing anywhere from 500GB to 5PB, you can see how managing this sheer volume of data can be difficult. Not only do you have an enormous amount of data to back up every night but you also have to keep track of where everything is. This is where virtualization comes into the picture. With storage virtualization, some form of virtualization software will sit between the applications and the storage itself. To access a file or to find a file that was backed up, you don't need to know exactly where the file is. The virtualization software will do that for you.

However, to truly understand how you can virtualize storage, you need to know a little about how storage is configured in midsize to enterprise environments. With that in mind, this chapter will lay the foundation for Chapter 13 by explaining each of the core concepts behind storage networking. The concepts covered in this chapter include the following:

- SCSI
- Fibre Channel SANs
- iSCSI SANs

We'll start by covering the foundation behind nearly all enterprise-class storage systems: SCSI.

Introducing SCSI

SCSI has long been a popular choice for connecting both external and internal storage to servers. When using SCSI to interconnect storage devices, you need to be aware of several pitfalls. If you use the wrong cables, adapters, or terminators, for example, you can find yourself in a lot of trouble. One of the arts with SCSI is to make sure all the parts are compatible with each other. Compatibility isn't too difficult if you're fortunate enough to fully understand the world of SCSI, so the next section begins with terminology.

Speaking SCSI

Understanding SCSI terminology is key to choosing the proper SCSI hardware for your storage infrastructure. Here are some of the primary terms associated with SCSI systems:

- **Chain:** Several SCSI devices connected together on the same SCSI bus.
- **HBA:** Card that acts as an interface between the SCSI bus and a host computer. The HBA is the SCSI card.
- **SCSI ID:** Unique number assigned to each SCSI device on a SCSI bus. Each SCSI device on the bus must have a unique ID, including the SCSI host adapters.
- **LUN:** Provides a way to differentiate between SCSI devices sharing a single SCSI ID.
- **SCSI bus:** Physical path for SCSI data flow. The following are the two general types of SCSI buses:
 - **Narrow:** 8-bit bus, uses 50-pin connector
 - **Wide:** 16-bit bus, uses 68-pin connector
- **Terminator:** Group of resistors placed on the end of a SCSI bus that prevents deflection of SCSI signals.
- **Target:** Another name for a SCSI device.

To get to the heart of SCSI, some terms may need a little more clarification, so we'll cover SCSI IDs and LUNs next and then move onto the other components of the SCSI architecture.

ID vs. LUN

The easiest way to differentiate between these two terms is to consider how houses are arranged on a street. Each house has a unique address, or number, assigned to it. The same can be said about SCSI devices on a SCSI bus. Now picture an apartment building containing several apartments. With one building and a single street address, another method is needed to differentiate each apartment. Therefore, each apartment gets its own apartment number. Think of LUNs as the equivalent to apartment numbers. Sometimes several SCSI devices will share a SCSI ID. Many disk arrays have a single configurable SCSI ID. LUNs identify drives in the array.

As far as IDs are concerned, the highest number equates to the highest priority. That's why most SCSI adapters use an ID of 7. For 8-bit SCSI, possible SCSI IDs range from 0 to 7.

Note SCSI IDs are usually set by using jumpers for internal SCSI devices and by using either a dial or a push-button numeric switch for external SCSI devices.

Using SCSI Buses

To keep it simple, a SCSI bus is the data path between SCSI devices. The bus consists of the HBA, cables, and terminators that make up the connections between all SCSI devices. When several devices are connected on a single SCSI bus, they're typically referred to as being *daisy chained*. Narrow SCSI uses an 8-bit bus, with 0 to 7 being the valid range of IDs. Wide SCSI typically uses a 16-bit bus, with 0 to 15 being the range of possible SCSI IDs.

To help differentiate between the capabilities of each modern SCSI bus, Table 12-2 provides information on each SCSI bus type.

Table 12-2. *SCSI Bus Types*

Bus Type	Bus Width (Bits)	Bandwidth (MB/sec)
SCSI-1	8	5
SCSI-2	8	5
Wide SCSI	16	10
Fast SCSI	8	10
Fast Wide SCSI	16	20
Ultra SCSI	8	20
Ultra SCSI-2	16	40
Ultra2 SCSI	16	80
Ultra160 SCSI	16	160
Ultra320 SCSI	16	320

The predominate SCSI interface today is the parallel interface, which is divided into the following categories:

- Single-ended (SE)
- Differential, or high voltage differential (HVD)
- Low voltage differential (LVD)

Tip When working with any SCSI component, you must know its interface type, as not all SCSI interfaces are interchangeable.

Many single-ended and LVD devices are compatible with each other, and you'll often see the distinction "LVD/SE" on many SCSI HBAs, alerting you that the adapter supports both interfaces. You can't mix LVD and HVD or single-ended and HVD on the same SCSI bus, unless your intention is to set your SCSI devices on fire!

Tip When a single SE SCSI device is connected to an LVD/SE bus, the bus automatically becomes and assumes the characteristics of an SE bus. Keep this in mind when planning your SCSI storage implementation.

Single-ended SCSI devices are generally the least expensive and consequently are the most common. With this interface, all electronic signals travel over a single wire and ground connection; thus, they have the name *single-ended*. A problem with single-ended devices is electrical noise. In areas of high electromagnetic radiation, SE SCSI can have problems.

Differential SCSI buses send electronic signals over pairs of wires, making transmissions over the differential medium less susceptible to noise than SE devices. At this point, we could put you to sleep discussing differential amplifier theory (a topic we happen to really love), but we'll spare you the misery. If you aren't willing to accept that differential SCSI is less susceptible to noise and want more detailed information on SCSI in general, point your Web browser to <http://www.scsifaq.org>. There you'll find more than you probably ever could have imagined about SCSI.

To tie up the differences between SE, HVD, and LVD cables, Table 12-3 lists their maximum distances.

Table 12-3. *SCSI Maximum Cable Lengths*

SCSI Interface	Maximum Cable Length
SE	6 meters
HVD	25 meters
LVD	12 meters

Now that you realize how important it is to know what type of SCSI device you're working with, you may be asking the question, how do I tell them apart? Standard symbols are inscribed on many SCSI devices, which allows you to determine their bus type. Figure 12-1 shows each symbol and its meaning.



Figure 12-1. *SCSI device symbols*

If you've worked with SCSI for a while, you've probably seen plenty of SCSI devices that have no symbol. These are generally older devices and interfaces and are most likely single-ended devices. If you want to be certain about a particular SCSI device, your best bet is to write down its model number and locate its data sheet at the manufacturer's Web site. Most manufacturers have their data sheets available, so you probably won't have any problems taking this approach. If all else fails, and you have an ohmmeter or multimeter handy, you can perform an ohm check of the adapter. To do this, take the following steps:

1. Power off the device.
2. Measure resistance between pins 2 and 24 on 50-pin connectors and between pins 2 and 33 on 68-pin connectors.

If you measure between 0 and 0.3 ohms, you have a single-ended device. If the measurement is higher and closer to 1 ohm, then you have differential.

Tip Before measuring resistance with the meter, touch both meter leads together to make sure it reads 0 ohms. Sometimes natural resistance in old meter leads will throw off your measurements.

Understanding Termination

Another common pitfall when working with SCSI is proper termination. Not only do you need to know when and where to terminate a SCSI bus but you also have to make sure to use the right terminators. Don't think that just because it fits it's good. This isn't true with SCSI. Terminators prevent a SCSI signal from being reflected up and down the SCSI bus repeatedly. Terminators act as a load that a SCSI signal can be dissipated across, thus absorbing the signal and clearing the SCSI bus. Without the right terminators, your SCSI bus won't work properly.

Where to Terminate

You must place your terminators in the right locations on the SCSI bus. The simple rule of thumb regarding termination is to place a terminator at the end of the bus. Be careful not to use a terminator in the middle of a bus. Figure 12-2 shows this concept.

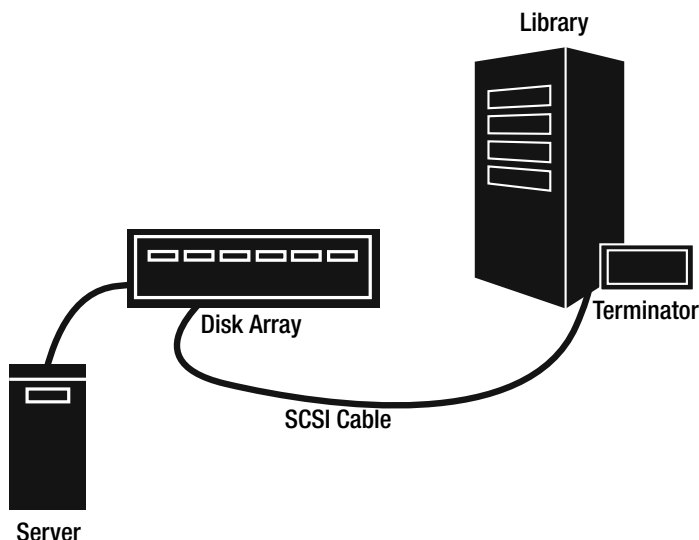


Figure 12-2. *Proper SCSI termination*

Figure 12-3 shows the actual connections to an external SCSI disk. Note that this illustration doesn't reflect the typical setup for a shared cluster disk because only one external cable is attached, meaning that the disk is attached to only one system. If configured as a shared cluster disk, the terminator would be replaced with an external cable that connects to a SCSI HBA on the second cluster node.

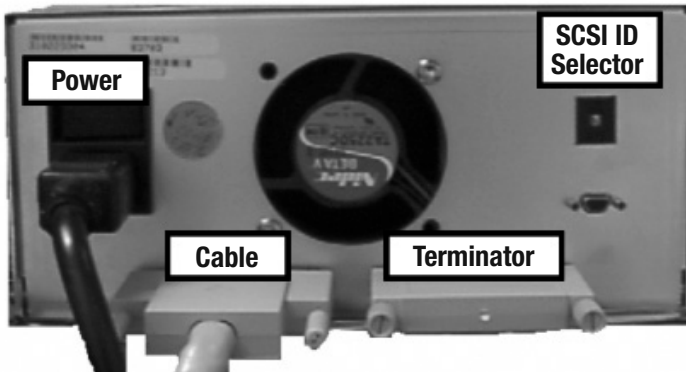


Figure 12-3. *External SCSI hard disk with terminator attached*

Tip Many SCSI cards have an autoterminate setting that can be configured in their BIOS. Enabling autotermination causes the cards to enable their internal termination when they sense they're at the end of the bus. Many cards that support autotermination act as terminators when they're powered off. This allows you to connect two SCSI HBAs to a single shared storage device without the need to purchase any other additional terminators.

If you're still a little confused about where to terminate a bus, just remember these rules:

- When connecting only internal SCSI devices:
 - Make sure the SCSI HBA is terminated.
 - Terminate the last device connected to the internal bus.
- When connecting only external SCSI devices:
 - Make sure the SCSI HBA is terminated.
 - Terminate the last device connected to the external bus.
- When connecting both internal and external SCSI devices to one HBA:
 - Disable termination on the SCSI HBA.
 - Terminate the last device connected to the internal bus.
 - Terminate the last device connected to the external bus.

Knowing where to terminate is only half the story. You must also know how to terminate, which requires knowledge of terminator types. Different SCSI interfaces have unique power requirements and thus warrant a terminator that meets this requirement.

Caution Make sure you use the proper terminators on your SCSI buses. Using a 50-pin HVD terminator on a 50-pin SE bus won't work. Since both terminators look the same, this mistake is easy to make.

Terminator Types

SCSI HBAs are typically terminated either by placing a jumper in a certain position or by enabling termination in the SCSI BIOS. For most SCSI devices, you'll have to place a physical termination device on their last SCSI connection in order to terminate the bus. If you're unsure of whether your SCSI device has onboard termination, check with the manufacturer. Most manufacturer Web sites contain information on the configuration options available for their products.

When selecting terminators, your first step should be to identify the SCSI bus you're using. If you're using single-ended SCSI, then you need a single-ended terminator. While matching the SCSI type to the terminator is the obvious consideration, deciding on terminators involves a few more considerations, depending on the SCSI flavor you're using, so the next section provides a quick look at the termination alternatives for each SCSI type.

SE Termination

Single-ended SCSI devices can be terminated by using passive, active, or forced perfect termination (FPT). The differences between each termination type are as follows:

- **Passive:** Termination provided by a simple resistor network; this isn't recommended for SCSI-2 or Ultra SCSI devices.
- **Active:** Termination network aided by voltage regulator. Regulated voltage results in a cleaner signal and is recommended for all SE SCSI buses.
- **FPT:** Termination provided using clamping diode network, which results in additional current on the SCSI bus, which in turn provides for longer possible cable lengths. This isn't recommended because FPT terminators can cause compatibility problems with the SCSI devices on the bus.

Caution Make sure all devices on the SCSI bus support FPT prior to using FPT terminators; otherwise, it's possible that some devices may be damaged as a result of using the FPT terminators.

It's usually easy to differentiate between active and passive termination on SCSI HBAs. Active termination is usually software-driven via the SCSI BIOS or provided via a jumper setting. Passive termination on HBAs is achieved through a connected resistor network.

LVD Termination

LVD termination is achieved using a 1.25 voltage source and a resistor network. While some exclusive LVD terminators exist, you can also find LVD/SE terminators that have the ability to autoswitch between LVD and SE termination. Because of the versatility of LVD/SE terminators, they've become popular.

HVD Termination

HVD SCSI buses are terminated using a passive resistor network. Although HVD uses a form of passive termination, you must remember to never use HVD terminators on an SE or LVD bus. At a minimum, you'll find that your bus operates intermittently, if it works at all.

Cable Selection

If you've already spent thousands of dollars on your SCSI storage devices, don't try to save a few bucks by buying cheap cables. When most problems arise on a SCSI bus, they're usually attributed either to the cables or to the terminators. This means proper cables should also be a concern of yours.

When deciding on cables, you'll find some cables that are configured using single wires, but the better cables will use twisted pairs of conductors. For narrow SCSI, you'll see 25 twisted pairs, and the cables will have 34 twisted pairs for wide SCSI. Also when selecting cables, don't forget about cable length. Table 12-3 listed the maximum cable length for each SCSI bus. The safest bet is to buy the best quality cables at the length you require. Chaining several SE SCSI devices using external SCSI cables is an easy way to build an unreliable SCSI bus!

Now that we've spent a good deal of time on SCSI storage, we'll now cover storage that uses the FCP.

Introducing Fibre Channel

Although SCSI has for years had a strong allegiance from countless devoted followers, the many advantages of Fibre Channel are starting a new trend for connecting to external storage. Fibre Channel architecture is versatile and scalable, offering the following advantages:

- Faster than SCSI, offering up to 4Gb/sec (500MB/sec) data transfer rates
- Allows cables distances to be as long as 10 kilometers
- Supports up to 16 million interconnected storage devices
- Allows for real-time hotswapping, permitting you to add and remove devices from a live bus, without any service interruption
- Gives you the ability to configure a SAN, which allows you to consolidate all your essential storage, including backup libraries, on a backend network

Figure 12-4 shows a simple SAN. The majority of today's storage area networks use Fibre Channel as their transport medium. SANs are entire physical networks that are dedicated to storage. With a SAN, you can interconnect storage devices on their own high-speed network, allowing for faster data transfer between hosts and also the sharing of storage resources. For example, many backup products allow servers to directly back up to a shared library in a SAN. This gives you much better backup and restore performance as compared to having to back up data over a LAN to a central media server with a direct-attached library.

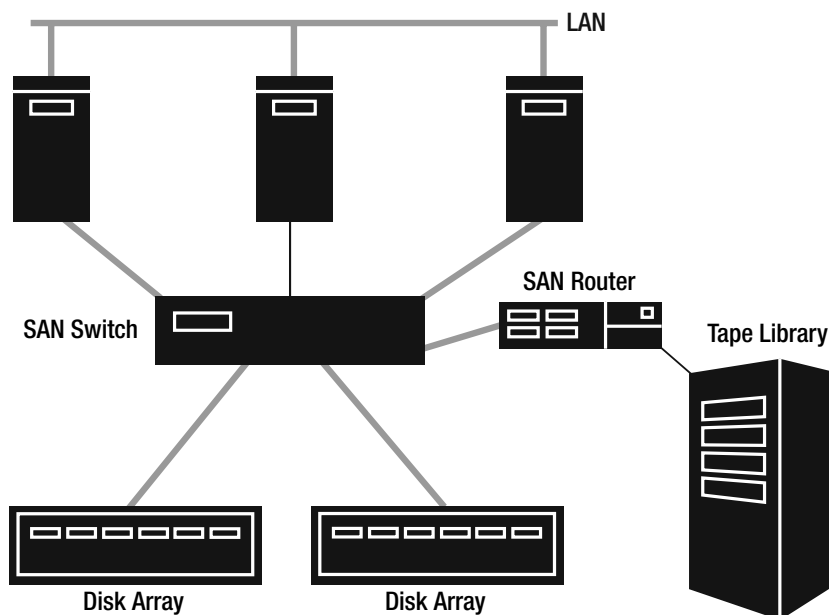


Figure 12-4. *Switched-fabric SAN*

Also, if you plan on configuring failover clusters with more than two nodes, your choice must be Fibre Channel. At this point, you may be thinking, why would I every use SCSI if Fibre Channel is so great? If so, here are some reasons to *not* use Fibre Channel:

- It's more expensive than SCSI.
- Implementations are more complex.
- It requires additional maintenance considerations.

If you're configuring storage in a small office and price is a consideration, then SCSI, EIDE, or ATA is still your best bet. Although Fibre Channel provides you with an abundance of alternatives for future expansion, it does so at a high price. In spite of the price, your mouth might be drooling at this point by the mere thought of the endless possibilities offered by Fibre Channel. With that in mind, you'll now look a little deeper at Fibre Channel.

Introducing Fibre Channel Cables

Many people may love to overwhelm you with their knowledge of Fibre Channel details. Although this may help them properly answer a Jeopardy question someday, it won't help them configure Fibre Channel any better than you. Rather than bore you with every aspect of fiber-optic technology, we'll spare you the details and focus on the essentials.

You can implement Fibre Channel using either copper cable or fiber-optic cable. Copper is much less expensive to implement and maintain than fiber-optic cable, but it doesn't offer the same potential bandwidth (currently 2Gb/sec maximum compared to 4Gb/sec with fiber optic).

Note If you're confused by *fiber* and *fibre*, remember that *fibre* is used in conjunction with Fibre Channel, and *fiber* is used to describe fiber-optic cables.

You can connect copper cable in a Fibre Channel network using either an intracabinet configuration or an intercabinet configuration. The differences between the two are as follows:

- **Intracabinet:** The maximum cable length is 13 meters, and all connections are typically made within a single enclosure.
- **Intercabinet:** The maximum cable length is 30 meters, and all connections are typically made between several enclosures.

As far as connectors are concerned, the initial Fibre Channel copper connectors were of the DB-9 variety, but now you'll mainly see high-speed serial data connectors (HSSDCs), which are similar to USB connectors in appearance.

Fiber-optic cables are generally referenced by their mode, or frequency, that the cables support. The following are the two fiber-optic modes:

- **Single-mode (9 micron):** Fiber-optic cable that allows for a maximum distance of 10 kilometers and offers a bandwidth of 4Gbps with plans for expansion to 10Gbps
- **Multimode:** Multimode cables come in two forms:
 - **62.5 micron:** Offers a maximum distance of 175 meters
 - **50 micron:** Offers a maximum distance of 500 meters

As you can see, the smaller the micron, the greater the distance the cable can support. The term *micron* refers to the size of the core of the optical cable. While fiber-optic cable is more expensive, it's ideal in areas that are susceptible to electromagnetic interference, since optical cables carry signals on a beam of light, as opposed to using electrical current.

Note We're only scratching the surface of fiber-optical cable architecture. For more information on this and other Fibre Channel–related topics, visit the Fibre Channel Industry Association's Web site at <http://www.fibrechannel.org>.

We're not yet completely finished covering architecture, but first you must understand the new hardware involved in a Fibre Channel network. After an overview of hardware, we'll cover Fibre Channel network topologies.

Introducing Fibre Channel Hardware Devices

Several new hardware pieces are involved in a Fibre Channel network. You'll see some familiar names with Fibre Channel, but devices will have different purposes. An example of this is the Fibre Channel bridge, which doesn't perform as a network bridge normally does. Among the hardware devices that you'll use in a Fibre Channel SAN are the following:

- HBA
- Gigabit Interface Converter (GBIC)
- Switch
- Hub
- Bridge/router

In the next five sections, we'll cover each of these hardware devices in greater detail.

Host Bus Adapters

The idea of HBAs is really nothing new. Today, you connect servers to external storage devices by using SCSI adapters. Fibre Channel HBAs fill the same role, providing an interface between a server's internal PCI bus and a Fibre Channel network.

To connect a server to a Fibre Channel network, you need to connect the HBA to a Fibre Channel cable. This is similar to using SCSI via an HBA on any PC or server. However, as you may well know, you can't just grab any SCSI adapter and use it to connect a server to a SCSI bus. Instead, you have to be careful to have everything matched up right. For example, connecting an SE SCSI hard disk to an HVD SCSI adapter is a great way to fry a perfectly good SCSI drive.

Most Fibre Channel HBAs have built-in adapters (known as GBICs) for a particular cable type, such as multimode optical or copper. With this in mind, make sure when purchasing a Fibre Channel HBA that the HBA is compatible with your planned Fibre Channel cable type.

Note For more information on HBAs, point your Web browser to QLogic (<http://www.qlogic.com>) or Emulex (<http://www.emulex.com>). QLogic and Emulex are the most popular Fibre Channel HBA vendors in the industry.

Gigabit Interface Converter (GBIC)

GBICs are modular transceivers that allow you to interconnect Fibre Channel cable to Fibre Channel devices such as switches and routers. The job of the GBIC is to convert the Fibre Channel signal from the transmission medium to data that can be interpreted by the HBA or switch. You can get GBICs for both copper and fiber-optic cable, and you can identify them by their cable connection type, which can be DB-9, HSSDC, HSSDC-2, multimode optical, or single-mode optical. Figure 12-5 shows a multimode optical GBIC.



Figure 12-5. *Multimode optical GBIC*

The advantage to GBICs is that they give you the flexibility to mix and match physical mediums such as fiber optic and copper in your SAN topology. Furthermore, GBICs allow hardware devices such as switches to be manufactured without relying on a particular medium type. Most switches have ports for GBICs. So, you'll connect the Fibre Channel cable to its appropriate GBIC type and then plug the GBIC into the switch. This means even if you decide to change your Fibre Channel transmission medium, you won't need to replace your Fibre Channel switch and will instead only have to upgrade the GBICs. With many Fibre Channel switches costing more than \$10,000, having this flexibility is significant.

Switch

Fibre Channel switches usually connect devices on the Fibre Channel network. Fibre Channel switches are available for both switched-fabric and arbitrated-loop topologies. In terms of connecting to the SAN, some switches have ports designed to connect to GBICs, while other switches allow Fibre Channel cable to be plugged directly into them.

The types of ports that a switch supports will tell you the type of networks to which the switch can connect. These are the most popular switch port types:

- **F_Port:** Used for switched-fabric topologies
- **FL_Port:** Used for Fibre Channel–arbitrated loop (FC-AL) topologies
- **U_Port:** Used for universal port (fabric or FC-AL)
- **E_Port:** Used to interconnect Fibre Channel switches

In addition to these standard ports, many newer Fibre Channel switches also provide ports to connect the switch to IP networks in order to support more advanced protocols such as FCIP, iFCP, and iSCSI.

Like with Ethernet switches, Fibre Channel switches also provide for multiple point-to-point connections when configured in a switched-fabric topology. Fibre Channel hubs, on the other hand, play a familiar role in SANs.

Note For more information on Fibre Channel switches, take a look at the product offerings of these popular switch vendors: Brocade (<http://www.brocade.com>), McDATA (<http://www.mcdata.com>), and Cisco Systems (<http://www.cisco.com>).

Hubs

Like with Ethernet hubs, bandwidth in a Fibre Channel hub is shared, meaning that only one host connected to the hub can transmit data at a time. You'll most likely see Fibre Channel hubs in FC-AL topologies, which are discussed in a moment. Although less expensive than switches, their inefficient bandwidth utilization make them a less desirable choice.

Bridge/Router

The *bridge*, or *router*, is a device that's used to connect a Fibre Channel SAN to a SCSI device. The job of the device is to bridge Fibre Channel communications to SCSI bus communications; hence, it has the name *bridge*. However, most vendors list this product as a router, since it "routes" between two storage mediums (Fibre Channel and SCSI). For whatever reason, however, no vendor seems to want to go out on a limb and call this device what it'd be called in the world of Ethernet—a *brouter* (bridging router)!

The router is an important consideration when you plan to implement a SAN, since it allows you to connect your existing SCSI storage devices (disk arrays and libraries) to the SAN. This way you don't wind up having to lose your initial SCSI storage investment.

Now that you have an idea of the different components in a Fibre Channel network, you may be wondering what they look like or how they're used, so that's what we'll cover next.

Note The two giants in the Fibre Channel router market are Advanced Digital Information Corporation (<http://www.adic.com>) and Crossroads Systems (<http://www.crossroads.com>).

Fibre Channel Network Topologies

Fibre Channel networks are connected using one of three different topologies:

- Point-to-point topology
- Switched-fabric topology
- Arbitrated-loop topology

Point-to-Point Topology

The point-to-point topology operates just as you'd think, by placing dedicated connections typically between a server and a storage device. This is equivalent to using a crossover cable to interconnect two computers on a network without needing a hub or switch. To configure a system in this topology, you'd connect its HBA directly to a hardware device. Figure 12-6 shows the point-to-point topology.

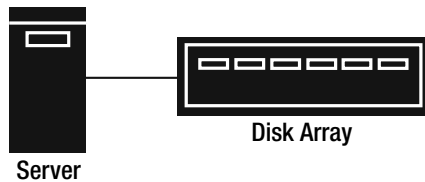


Figure 12-6. *Point-to-point SAN topology*

Switched-Fabric Topology

Switched fabric has emerged as the most popular Fibre Channel topology. The simplest way to think of switched fabric is as an equivalent to Ethernet networking. Each device connected to the SAN will connect to a port on a switch. For scalability, you can add purposed switches to the SAN fabric. Figure 12-4 (shown earlier) displays a switched-fabric topology.

Arbitrated-Loop Topology

Arbitrated loop is the equivalent to a Token Ring network topology. Arbitrated-loop topologies support up to 127 nodes interconnected through a hub. Like with a Token Ring topology, only one device attached to the loop may transmit at a time. Figure 12-7 shows the logical configuration of an arbitrated-loop topology. Although this topology is less expensive to implement because of the lesser cost of Fibre Channel hubs over switches, it's also not as scalable or efficient.

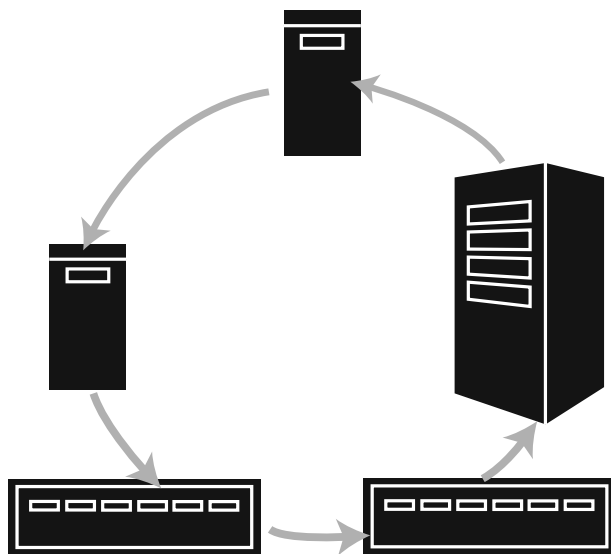


Figure 12-7. *Arbitrated-loop SAN topology*

Understanding Zoning

If all your servers are connected to the SAN, and nearly all server storage resides on the SAN, then you have a potential security nightmare. Imagine the server in the engineering department being able to see all the storage on the finance server. We're sure you can imagine what could go wrong. Many organizations don't want every server on the SAN to be able to use every storage resource on the SAN. Not only is it possible that unauthorized users can view data but also someone may accidentally delete data needed by another server or application. Also, if two systems attempt to write to the same physical disk at the same time, the disk can become corrupt. You can avoid these problems with zoning.

The easiest way to think of zoning is of the SAN equivalent to VLANs. With LANs, you can set up VLANs on a switch to segment the single physical switch into multiple logical switches. This causes connections to some switch ports not to be able to see connections to other switch ports. With zoning, you can apply the same concept to SAN switches. This way, a particular server can see only what its switch allows it to see.

You have two different methods for setting up zoning. One way to zone is by port. For example, you can allow devices on switch 1, port 2, to communicate with devices connected to switch 1, port 9. You can also configure zoning using what are called *world-wide names* (WWNs). WWNs are unique 64-bit identifiers for devices or ports. Some devices with multiple ports have WWNs for each port, allowing for more granular management. Because of their length, WWNs are expressed in hexadecimal, so a typical WWN would look like 4D:01:6B:94:59:D9:12:74 (a format similar to that used with Ethernet network adapter MAC addresses).

With WWN zoning, devices are zoned by device or device ports. This allows you to move the device on the SAN and change its associated switch port without impacting the zoning configuration. However, if the device fails and has to be replaced, you'd have to reconfigure the zoning so that the WWN of the replacement device is associated with the correct zone.

If you need to fully secure data on the SAN, zoning is the best option, especially when multiple departmental servers or applications are accessing storage on the same SAN. Consider zoning to be your SAN's best insurance against data loss, unauthorized access, or data corruption.

Configuring Fibre Channel Hardware

The best information on configuring your Fibre Channel network will come from the hardware vendors you decide to use. In writing this book, we went with an ADIC FCR 250 router to connect our Fibre Channel network to SCSI storage devices and had each node and the router connected using a Brocade Silkworm 2400 switch. This isn't the latest and greatest technology available today, but this stuff is expensive!

Most of your switches and routers allow you to connect and configure them using up to three methods:

- Telnet
- Hyperterminal
- HTTP (Web browser)

You'll most likely have to hyperterminal to the bridge, router, or switch using a null modem serial connection connected directly to one of your servers or workstations. From there you can configure the TCP/IP settings of the device so it can be accessed with either Telnet or HTTP. You'll probably find the HTTP interface to be the easiest to use. To give you an idea of these configurations, Figure 12-8 shows the Brocade Silkworm 2400 Web GUI that you can use to configure zoning.

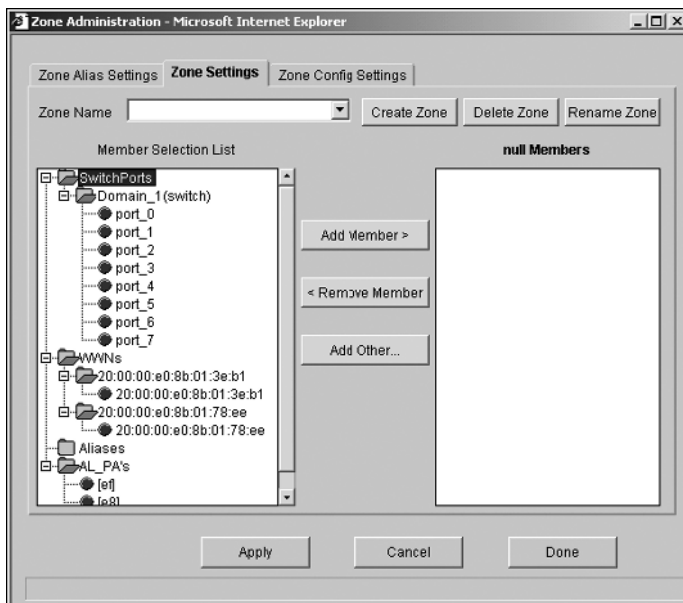


Figure 12-8. Configuring zoning on a Brocade Silkworm 2400

Extending the SAN with FCIP and iFCP

For many organizations that require high data availability, disbursing storage across two or more remote sites offers the security of being able to maintain data availability even after a disaster has occurred at one site. You can achieve this availability through technologies such as clustering, replication, snapshots, and traditional backup and recovery. For organizations to be able to configure a storage infrastructure that traverses geographical boundaries, they need a protocol that can take advantage of WAN economic limitations.

The cheapest transmission medium is the Internet, which requires IP. With this in mind, wouldn't it be cool to be able to bridge SANs in two sites together through the Internet? That's what Fibre Channel over IP (FCIP) is all about. For this to happen, you'd need a device capable of doing the translation from Fibre Channel to FCIP. Some Fibre Channel switches have integrated FCIP ports that allow you to do this. Remember, however, that FCIP doesn't provide any means to directly interface with a Fibre Channel device; instead, it's a method of bridging two Fibre Channel SANs over an IP network.

Internet Fibre Channel Protocol (iFCP) is much more robust than FCIP. Like FCIP, iFCP can also be used to bridge Fibre Channel switches over an IP network. However, this protocol also gives you the ability to network native IP storage devices and Fibre Channel devices together on the same IP-based storage network. With the rise of Gigabit Ethernet networks, consider iFCP to be a way to provide full integration between your Fibre Channel and IP network. Another rising protocol that provides the same level of hardware integration over Gigabit Ethernet is iSCSI. We'll cover iSCSI next.

Introducing iSCSI

Internet SCSI works similarly to iFCP, except that instead of encapsulating FCP data in IP packets, it encapsulates SCSI data. In being designed to run over Ethernet, iSCSI allows you to leverage existing Ethernet devices on your storage network. Also, let's assume you purchase a few new Gigabit Ethernet switches for an iSCSI SAN. As technology improves and you decide to upgrade to faster gigabit switches, you can use the older switches to connect hosts on the LAN. With Fibre Channel switches, you don't have this level of flexibility.

Understanding iSCSI Architecture

iSCSI architecture involves a host configured as an iSCSI target. The iSCSI target can be a server with locally connected storage (IDE, SCSI, or even Fibre Channel) or can be a storage device that natively supports iSCSI. Clients that access the storage over the network using the iSCSI protocol are known as *initiators*. Initiators need to have iSCSI client software installed in order to access the iSCSI target. To set up your own iSCSI storage network, this book includes both iSCSI target and initiator software from Rocket Division Software. Look for this software in the RocketDivision folder on the companion CD. Figure 12-9 shows a typical iSCSI environment, with two initiator hosts and one iSCSI target.

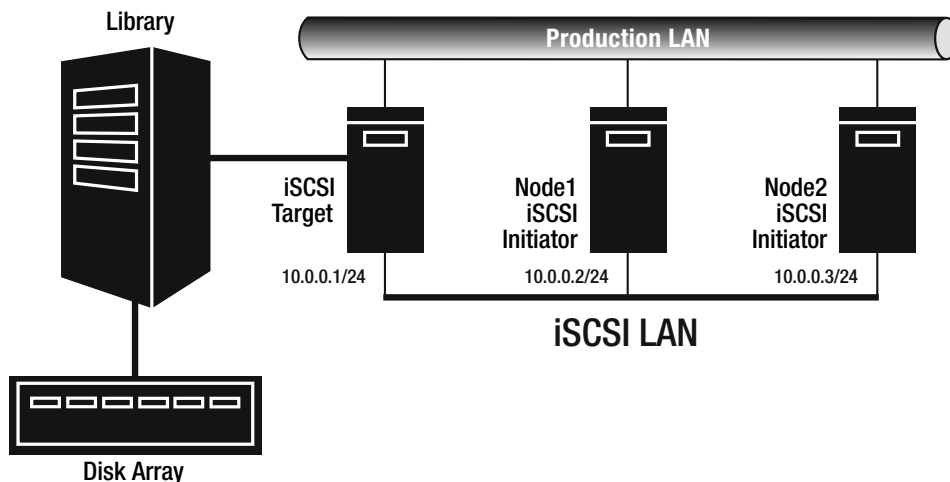


Figure 12-9. iSCSI architecture

Securing iSCSI

Sending data over Ethernet, especially if the data traverses several LANs or WANs, doesn't make every administrator comfortable. With that in mind, iSCSI supports both encrypting authentication and encrypting the data payload. iSCSI supports the following methods for the iSCSI target to authenticate iSCSI initiators:

- Challenge Handshake Authentication Protocol (CHAP)
- Kerberos version 5
- Simple Public Key Generic Security Service (SPKM1)
- Simple Public Key Generic Security Service version 2 (SPKM2)
- Secure Remote Password (SRP)

To encrypt the data payload, iSCSI uses IP Security (IPSec). IPSec is written into the iSCSI standard, so for any hardware or software to use iSCSI, it must support IPSec encryption using Encapsulating Security Payload (ESP), 3DES encryption, and Internet Key Exchange (IKE) for session negotiation. To configure the iSCSI security settings, consult the documentation provided by the iSCSI product vendor. Chapter 11 has an example of iSCSI configuration.

Note For more information on iSCSI, take a look at <http://www.iscsistorage.com> or examine the Linux iSCSI Project at <http://linux-iscsi.sourceforge.net>.

Using SAN Backup and Recovery Techniques

Although you can build SANs with Fibre Channel, iSCSI, or a mixture of both, the ways you can use SANs to back up data are similar for all SANs, regardless of protocol used. In the following sections, we'll cover the backup options available for SANs. Since servers attached to the SAN can see storage as local, you can still perform your normal backup types (full, incremental, differential, and copy) to storage devices on the SAN. So, you shouldn't need to alter your current backup configurations, unless you plan to take advantage of the other backup types now offered by SANs. The next three sections describe those backup types.

Performing LAN-Free Backups

LAN-free backups prevent a server's backup data from having to traverse the LAN, thus affecting the network performance of both clients and servers. If servers are directly connected to the SAN and are backing up to storage devices located in the SAN, then by definition those servers are performing LAN-free backups. When backup data doesn't traverse the LAN, it's considered LAN-free. Figure 12-10 shows this type of backup.

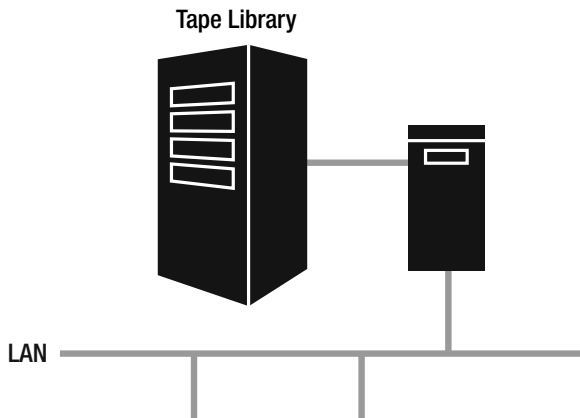


Figure 12-10. *LAN-free backup*

Performing Server-Free Backups

With server-free backups, no CPU cycles are used by the server to be backed up, thus freeing it for other activities. With traditional backups, the servers involved in the backup (the server doing the backup and all of the servers being backed up) had to utilize their own CPU cycles to collect and move data. This meant that anytime a backup was run, a significant burden was often placed on these servers. With such a heavy load on the servers, most organizations were forced to run backups only during nonwork hours, such as at night and on weekends.

SANs, however, change the entire storage landscape. Since any device connected to the SAN can theoretically access any storage on the SAN, you don't need to burden every server for backups. Instead, you can designate a data mover to move data on the SAN to backup devices. Data movers can be other servers or can be libraries or switches with data mover software integrated in their firmware. With this approach, you can back up SAN resources with no impact on the primary servers attached to the SAN. The only major requirement, other than a data mover host, is to ensure that your backup and recovery software supports server-free backups for your particular SAN configuration. Figure 12-11 shows this type of backup. Note that the dotted line represents the flow of backup data.

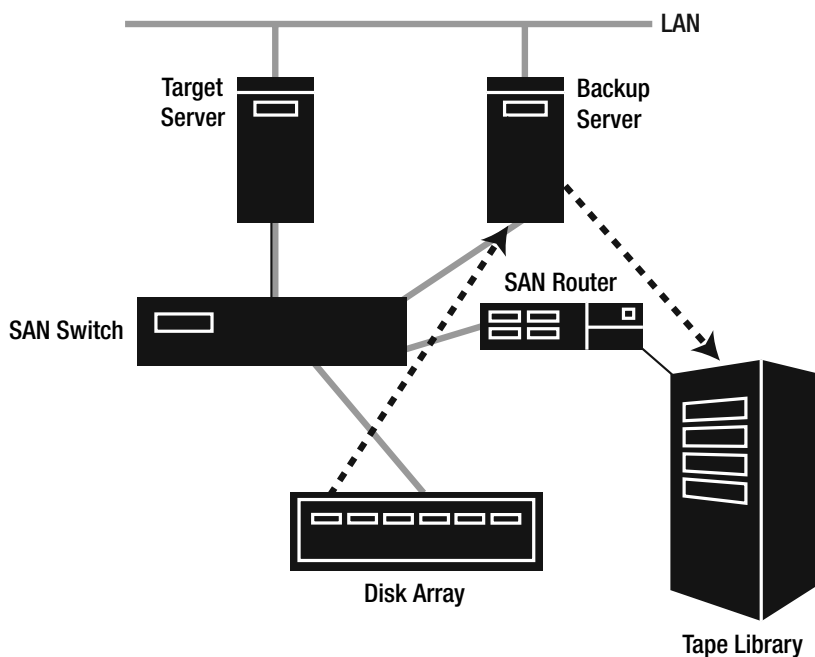


Figure 12-11. *Server-free backup*

Performing Serverless Backups

With serverless backups, no server is involved in the backup process. This means you don't even need a server designated as the data mover. With this approach, SAN switches or routers that support SCSI-3 Extended Copy commands can move the data to be backed up. To use this type of backup, you'll need to ensure that SCSI-3 Extended Copy is supported by your SAN hardware as well as your backup software. Figure 12-12 shows this type of configuration. Notice that the backup data flows from a disk array to a router and then to a library.

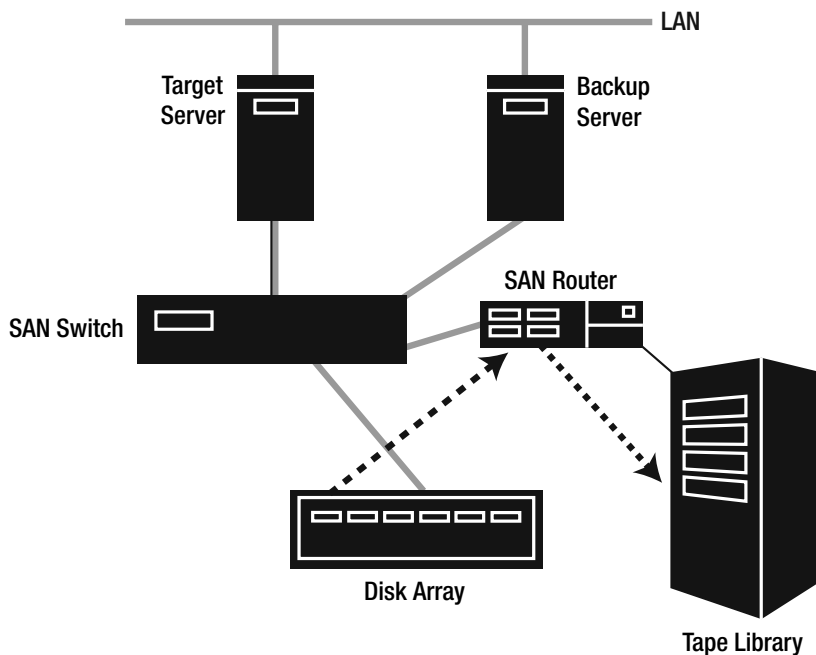


Figure 12-12. *Serverless backup*

Tip For more information on storage networking technologies and concepts, take a look at the Storage Network Industry Association home page at <http://www.snia.org>.

Summary

In this chapter, we provided you with an overview of SAN technologies so you can further explore storage virtualization in Chapter 13. With 500-page books dedicated to storage networking alone, we couldn't cover everything in just one chapter. However, with a good understanding of how storage networks work, along with an appreciation of the hardware devices involved, you're ready to see how storage virtualization is further changing how we look at storage. Turn to Chapter 13 to see how you can manage and view entire storage systems that traverse several physical locations as single logical entities.



Virtualizing Storage

Storage *virtualization* is the concept of logically representing multiple physical storage devices. RAID opened the door to storage virtualization, but managing anywhere from hundreds of gigabytes up to terabytes (or even petabytes!) of storage has resulted in a growing problem. When you have so much storage to manage, how do you locate files? How do you manage backups? When you need to restore something, where is it? How do you simplify access to storage for the users? All these questions have resulted in a growing storage management headache. In this chapter, we'll explore the driving theory behind storage virtualization, as well as the technologies that make it happen. We'll start by covering storage virtualization's first technology: RAID.

RAID: The Root of Storage Virtualization

When Nostradamus first predicted storage virtualization, he said it would start with RAID and within 20 years would really take off. He couldn't have been more accurate!

Well, maybe Nostradamus didn't explicitly mention RAID in his many predictions, but he did predict that California would split from the United States in 1988. Oh wait—that didn't happen either! Rather than give any more credit to Nostradamus, we'll cover early storage virtualization: RAID.

In short, RAID allows you to configure two or more physical disks to collectively act as a single logical disk. Presenting physical disk resources as a single logical element to the operating system provides for a simple form of storage virtualization. In the following sections, we'll cover the most common types of RAID arrays and show how to configure RAID.

Introducing Common RAID Levels

To implement fault-tolerant disks, you simply need to know the fundamentals of each RAID level and also each level's common uses. The following are the most common RAID implementations today:

- RAID 0
- RAID 1
- RAID 5
- RAID 0+1
- RAID 1+0
- RAID 5+0

Over the next couple of pages, we'll take you on a brief stroll through the world of RAID, stopping along the way to examine common uses for each RAID level.

RAID 0

Many people consider RAID 0 to be the adopted son of the RAID family. Although level 0 is welcome and accepted, it doesn't have one trait common to all other RAID levels—fault tolerance. Remember that the first word in the RAID acronym is *redundant*, and RAID 0 provides no redundancy. Because of this, RAID 0 is often combined with other RAID levels in order to achieve fault tolerance.

Although not fault tolerant, RAID 0 offers the fastest performance of all RAID levels. RAID 0 achieves this level of performance by *striping* its data across two or more physical disks. Striping means that data is being written to multiple disks simultaneously. The operating system sees all the disks in what's known as the *stripe set* as a single physical disk. Figure 13-1 depicts RAID 0 disk striping.

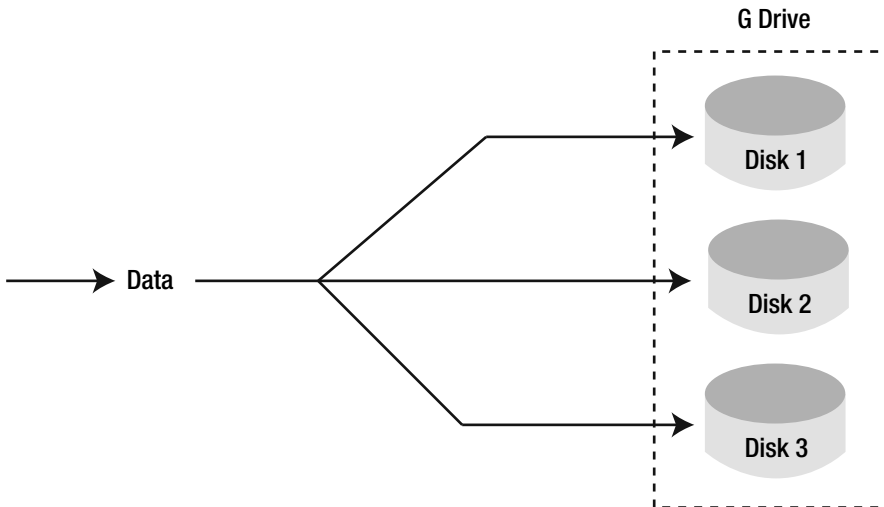


Figure 13-1. RAID 0 array consisting of three physical disks

To understand how RAID works, suppose you want to store the acronym *RAID* in a RAID 0 array containing four disks. Now picture each disk as a cup. Since RAID has four letters, each cup would have a single letter stored in it. With the letters evenly spaced, you could theoretically drop all the letters into the four cups simultaneously. This is the advantage of RAID 0—it's fast. A problem, however, happens when one of the cups (disks) is lost or damaged. You wind up losing a portion of the stored data, and recovery isn't possible. The end result is that all data on all disks is lost.

Here are some reasons to use RAID 0:

- You have a need for speed—and nothing is more important.
- You don't need data redundancy or fault tolerance.
- You need fast storage available for temporary or dynamic files.

Because of these reasons, RAID 0 is ideal for high-performance databases or logging, especially when combined with other RAID levels (discussed next).

Here are a couple of reasons to avoid RAID 0:

- You require fault tolerance.
- You're willing to sacrifice a slight degradation in performance for fault tolerance.

The remaining RAID levels we'll cover in the next sections offer fault tolerance at the expense of some of the performance found in RAID 0.

RAID 1

RAID 1 represents the first fault-tolerant RAID level and is available in two forms:

- Disk mirroring
- Disk duplexing

With both disk mirroring and disk duplexing, two or more physical disks provide redundancy by having one or more disks mirror each other. Data written or deleted from one disk in the mirror set is automatically written or deleted on all other disks in the set. With this approach, you ensure fault tolerance by having redundant copies of the same data on several disks. The failure of a single disk won't cause any data loss.

The disk mirroring and disk duplexing implementations of RAID 1 differ in how the physical drives are connected. With disk mirroring, the physical disks in the mirror set are connected to the same disk controller. With disk duplexing, the physical disks in the mirror set are connected using at least two disk controllers. Disk duplexing is the more fault-tolerant RAID 1 implementation because it removes a disk controller as a single point of failure.

Figure 13-2 illustrates the operation of a RAID 1 array.

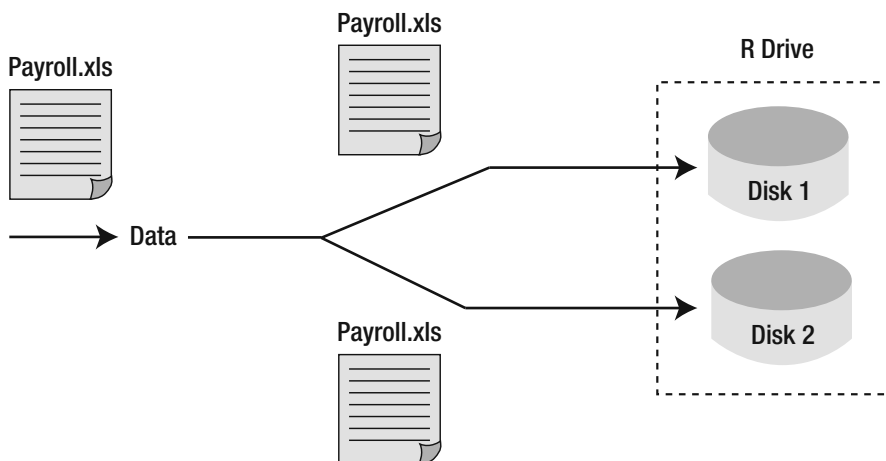


Figure 13-2. RAID 1 array consisting of two physical disks

Notice in Figure 13-2 that the file `Payroll.xls` is being saved to the RAID 1 array. When the file is saved, the entire file is written to both disks in the array. The actual write operation doesn't complete until the file is finished being written to the slowest disk. This means the actual performance of the RAID 1 array will be equal to the speed of the slowest disk.

RAID 1 is ideal when you're looking for an easy means to ensure data redundancy. Because RAID 1 automatically creates a mirror of disks, you're getting a continuous online backup of data. This allows for little to no data loss in the event of a disk failure. Keep in mind, however, that with both physical disks in the same physical location, RAID 1 should still not be considered a substitute for backup. Although RAID 1 gives you online data protection, it doesn't protect against disasters.

The one disadvantage to RAID 1 is that you have to purchase at least twice the amount of disk space for the data you want to store, depending on the number of disks in the RAID 1 mirror. If you're planning on configuring two disks to mirror each other, remember that one disk will work exclusively as a backup. Adding disks to the mirror set will obviously raise the price of storage even more. For example, to build a 200GB RAID 1 volume consisting of two physical disks, you'd need a total of 400GB of storage (2 disks × 200GB).

RAID 5

RAID 5 operates similarly to RAID 0 by striping data across multiple disks. However, two primary differences exist between RAID 5 and RAID 0:

- RAID 5 requires three or more physical disks, whereas RAID 0 can be implemented with just two disks.
- RAID 5 incorporates parity for data protection.

By using parity, a RAID 5 disk array can withstand the loss of a single physical disk and still operate. This is where the fault tolerance stops, though. If more than one disk in the array fails, all data is lost. Like with all the other RAID levels, the operating system sees the physical disks in the array as a single disk. In terms of performance, RAID 5 is slower than RAID 0, but it outperforms RAID 1.

Since RAID 5 uses parity to provide fault tolerance, you must consider the storage of the parity data when sizing a RAID 5 array. The basic cost for protection in a RAID 5 array amounts to the cost of one physical disk. For example, if you configured four 200GB disks as a RAID 5 array, you'd be left with 600GB of total storage, which equals the sum of the capacity of all but one disk. This amounts to 600GB/800GB, or 75 percent availability of your purchased 800GB of storage. As you add disks to the array, the cost decreases. If you used five 200GB disks instead, you'd still need only one disk for the parity bit, giving you 800GB of total storage, or 80 percent availability of the purchased storage space.

RAID 5 is the ideal storage solution in the following instances:

- You're looking for both speed and fault tolerance for a shared storage array.
- Data must be available, even after the loss of a single disk.

Here are some reasons for thinking twice about RAID 5:

- You can't afford to sacrifice any storage space, even if it allows for fault tolerance. (As inexpensive as magnetic disk storage is becoming, this isn't a very good excuse.)
- Speed is more important than protection.
- Data stored on the array is so volatile that fault tolerance isn't necessary.

RAID 0+1

RAID 0+1 arrays are commonly known as *mirrored stripes*. This is because data is first striped to a RAID 0 array and then mirrored to a redundant RAID 0 array. Figure 13-3 shows this concept.

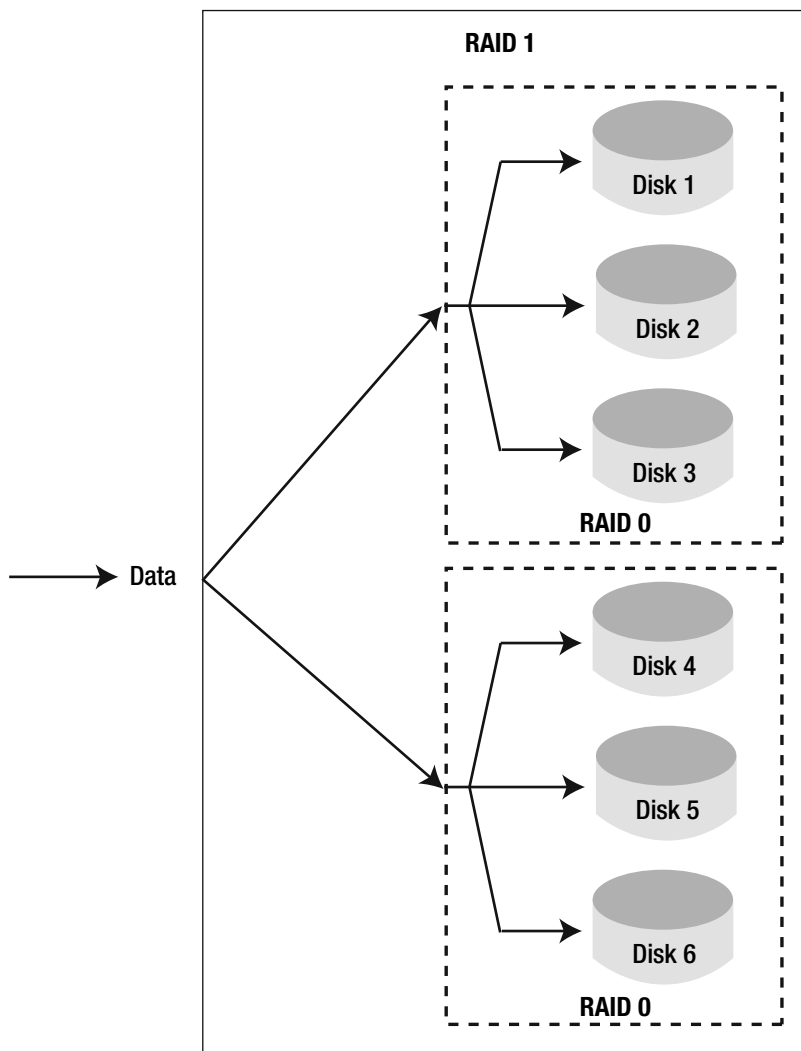


Figure 13-3. RAID 0+1 array

To configure RAID 0+1, you start by configuring two RAID 0 arrays. Next, you create a mirror from the two arrays. Note that by involving RAID 1 for redundancy, your storage investment will need to be double the amount of your storage requirement. Assuming that the array shown in Figure 13-3 uses 100GB disks, each RAID 0 array would be able to store 300GB of data ($100\text{GB} \times 3 \text{ disks}$). Since the second RAID 0 array is used for redundancy, it can't store new data (in addition to what's stored in the first RAID 0 array). This means that in purchasing 600GB worth of disk storage, you can use 300GB, or 50 percent.

The advantage of RAID 0+1 is that it offers the performance of RAID 0 but also provides fault tolerance. You can lose a single disk in the array and not lose any data. However, all you can lose is one disk. If you're looking for better fault tolerance, then RAID 1+0 is the better choice.

RAID 1+0

RAID 1+0 (also known as RAID 10) combines RAID 1 and RAID 0 to create a striped set of mirrored volumes. To configure this type of RAID array, you first create mirrored pairs of disks and then stripe them together. Figure 13-4 shows an example of how RAID 1+0 is implemented.

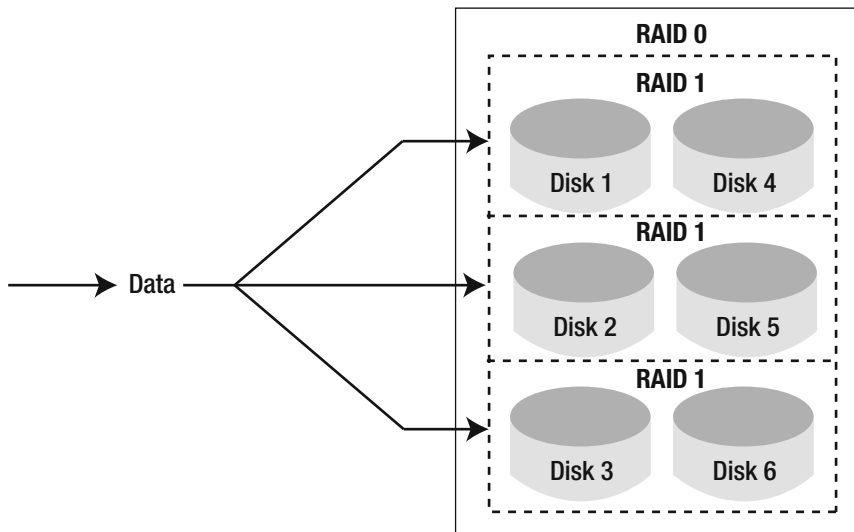


Figure 13-4. RAID 1+0 array

Notice that the configuration of RAID 1+0 is exactly the opposite of RAID 0+1. With 1+0, you start by mirroring two or more disks. This is why you see individual RAID 1 arrays inside the RAID 0 array. Then you stripe the mirrors together. The advantage to RAID 1+0 over RAID 0+1 is that it's more fault tolerant. If the array has six disks, you could lose up to three disks without losing any data. The number of disks that can fail is determined by where the failures occur. With RAID 1+0, as long as one physical disk in a mirror set in each stripe remains online, the array will remain online. In Figure 13-4, Disk 1, Disk 2, and Disk 3 could fail without interrupting data access. Also, if Disk 1, Disk 5, and Disk 6 failed, data would still be available. The bottom line in both of these scenarios is that at least one disk in each stripe is still online. If you

lose all the disks in a stripe (Disk 3 and Disk 6, for example), then all data is lost. Ultimately, RAID 1+0 gives you a high level of performance as well as fault tolerance. Like RAID 1 or RAID 0+1, your storage investment will still need to be double your storage requirement. If cost is a problem and you can sacrifice speed, then you may want to go with RAID 5+0.

RAID 5+0

Similar to RAID 1+0, RAID 5+0 is configured by combining RAID 5 and RAID 0. By striping data across RAID 5 volumes instead of RAID 1 volumes, you greatly reduce the cost per gigabyte. Figure 13-5 shows a RAID 5+0 array.

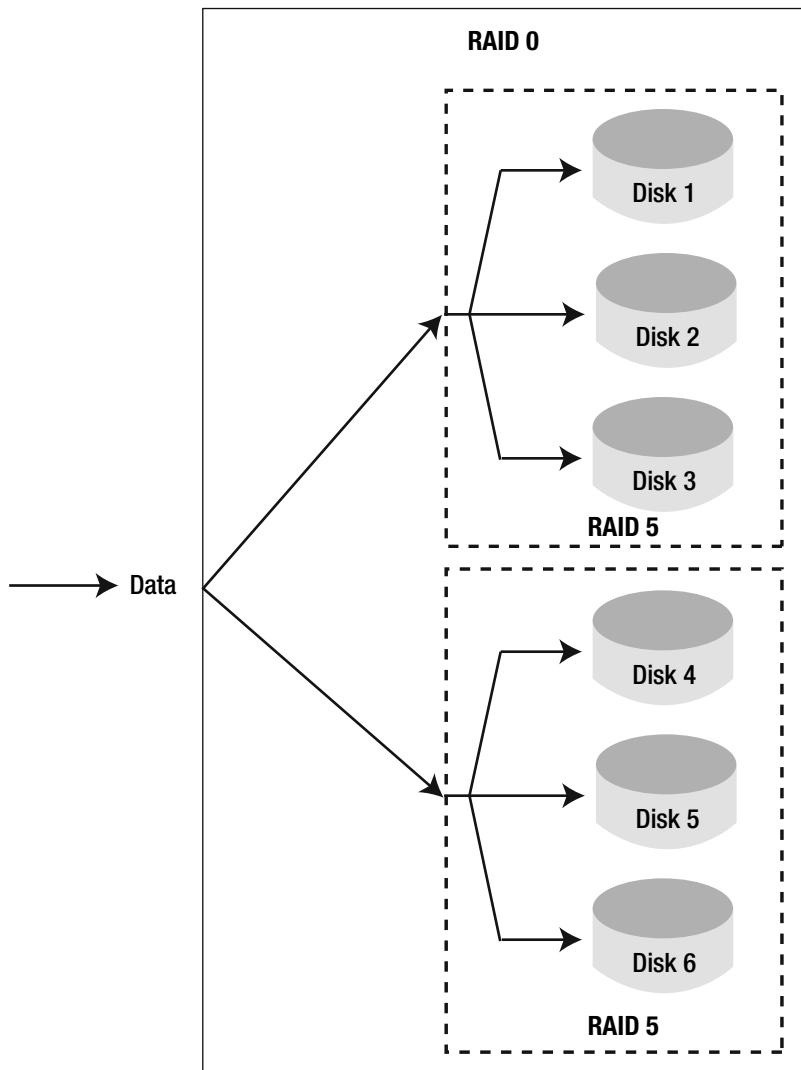


Figure 13-5. RAID 5+0 array

In comparison to RAID 5, you'll see faster write access to a RAID 5+0 array; however, when a drive fails, disk I/O to the array slows down significantly. Unlike RAID 5, RAID 5+0 is more fault tolerant in that it can withstand the loss of a disk in each of its subarrays. The array shown in Figure 13-5 can withstand the loss to a single disk in each group of three and still maintain operation. The sizing considerations of RAID 5 still apply with RAID 5+0, with the exception that the size of a disk in each RAID 5 subarray must be subtracted from the total available storage. For the array in Figure 13-5, if each disk were 400GB, you'd have a total of 1600GB of available storage. You can determine this number by subtracting the capacity of one disk in each subarray from the total capacity ($2400\text{GB} - 400\text{GB} - 400\text{GB} = 1600\text{GB}$). As with RAID 5, as you add disks to each subarray, the cost of total storage decreases.

Implementing RAID

Now that we've covered the different methods for logically configuring RAID, we're sure you're thinking it'd also be nice to know how to actually configure RAID. In general, you have two approaches for setting up RAID: hardware RAID and software RAID. Hardware RAID volumes are configured and managed by a hardware RAID controller, whereas software RAID is configured and managed through either the OS or a third-party application. In the next two sections, we'll cover the two fundamental RAID implementation methods in more detail.

Implementing Hardware RAID

Since hardware RAID is managed by a hardware RAID controller, which is typically a PCI card with either SCSI or Fibre Channel adapters (to connect to the storage devices), hardware RAID implementations are transparent to the OS. As far as the OS is concerned, it sees a single physical disk that's presented to it by the hardware RAID controller. Furthermore, since the array is managed by a separate hardware device, no CPU cycles are consumed on the host system for RAID calculations (in the event of RAID 5). When the operating system manages a RAID array, a tremendous burden is placed on the CPU.

If you're worrying about configuration with hardware RAID, don't. You can do the configuration during bootup by accessing the BIOS of the RAID controller. Typically you may have to navigate through a couple of menus to configure the disks in the array. This could involve adding or removing disks from an array or selecting the type of array, such as a RAID 1 or RAID 5, to configure. Many RAID controllers have a built-in scan feature that allows you to easily check the disks attached to the controller. This allows you to easily verify that the controller can find all physical disks.

Many of the prominent SCSI and RAID controller vendors post technical manuals for their controllers on their Web sites. The following are the sites for some of the prominent vendors in the RAID market:

- <http://www.adaptec.com>
- <http://www.qlogic.com>
- <http://www.lsilogic.com>
- <http://www.advansys.com>

- <http://www.symbios.com>
- <http://www.ami.com>
- <http://www.mcdata.com>

With a handle on the available hardware RAID solutions, you'll now take a quick look at software RAID.

Implementing Software RAID

With software RAID, you don't need a hardware RAID controller to configure the volumes in the RAID array; instead, the OS or an application handles everything. The advantage to this approach is that you can configure RAID with no additional hardware investment (RAID controller). Also, with software RAID not needing to rely on a hardware RAID controller, software RAID doesn't have the controller as a single point of failure for storage access.

Of course, these benefits to software RAID are at the expense of significant CPU overhead. The CPU loading of software RAID often makes it impractical on high-volume enterprise-class servers. However, if you're just looking for fault tolerance on a small office server without having to invest too heavily in hardware, then software RAID may be exactly what you need. As with hardware RAID, you still must use multiple physical disks to configure the RAID array, so breaking a disk into partitions to build a software RAID array isn't an option.

With Windows operating systems, you can configure software RAID using the Disk Management utility, which is part of the Computer Management MMC. Windows Server OSs support software RAID 0, RAID 1, and RAID 5 arrays, while Windows client operating systems support RAID 0 only.

With Linux operating systems, you can configure software RAID 0, RAID 1, and RAID 5 using the Disk Druid tool during a GUI installation of the OS. If the OS is already installed, you can use the Raidtools package to configure and manage software RAID.

Although RAID is one of the earliest forms of storage virtualization, today's storage networks have gone from virtualizing the resources of a single disk to virtualizing the resources of an entire organization. The Storage Network Industry Association (SNIA) Shared Storage Model has shaped much of the virtualization work.

Introducing the SNIA Shared Storage Model

The SNIA Shared Storage Model is the foundation for nearly all storage virtualization work today. With that in mind, we didn't consider it fair to talk about storage virtualization without getting to the heart of the technology. And there's no better way to do this than to examine SNIA's Shared Storage Model. The following sections come directly from the SNIA Technical Council's *Shared Storage Model: A Framework for Describing Storage Architectures*. After reading about the SNIA shared storage architecture, you'll then see how you can apply this model to modern storage networks.

■ **Note** The following sections come directly from the SNIA Technical Council's *Shared Storage Model: A Framework for Describing Storage Architectures* (SNIA, 2003), pages 10–21. As the preeminent storage networking think tank, SNIA has a perspective on the future of shared storage networking that's worth reading.

Why a Model for Shared Storage?

This document presents the *SNIA Shared Storage Model*, which describes a set of practical, possible *storage network architectures*.

Such an architecture describes a particular functional partitioning of services across the physical and logical resources in a shared storage environment. In use, an architecture is used to develop a *storage system design*, which includes things like the quantity and size of the resources, and possibly choices of suppliers, versions, etc. A completed design contains enough information to allow construction of a real system that conforms to it.

The SNIA Shared Storage Model is deliberately simple on the surface: the goal is to make it easy to use yet rich enough to cover a wide range of storage networks that are being—or could be—deployed. To make this easier, the model is primarily described in a graphical form, supported by a concise terminology that supports the key concepts.

A major intent of the model is to highlight the fundamental structures of a storage system that have the largest effect on the system's value proposition. These include but are not limited to

- The functions or services that a storage network architecture can support.
- The division of functional responsibility among the components of the system.
- Relationships between control and data flows.
- Boundary conditions where interoperability is likely to be an issue. This includes both places where interactions take place between architecture modules (and hence interoperability is required) and what kinds of interoperability must occur there.
- The implications of interface abstraction on technology evolution and marketplace competition.

Of course, the model doesn't explicitly cover all possible architectures. Instead, the intent is that the user of the model can use it to develop and describe his or her own particular mix of architectural elements and choices. Nonetheless, our experience with it to date has been that it is capable of covering a wide variety of different architectures and designs. It is likely, therefore, that the model does cover a new situation—although perhaps not in the most immediately obvious way.

Benefits of the Model

The benefits from a model of this form are that it can

- Provide a common vocabulary that people comparing and designing architectures can use to communicate with one another and make their design efforts and documentation more precise. (The analogy here is with the “definitions of terms” that are an important deliverable from the IETF and standards bodies.) This will also make it easier for vendors to explain their offerings to customers.
- Provide a common way of recording network storage architectures so that these architectures can be compared and contrasted with one another, making it easier for customers to compare different architectures and proposals.

Overall, the hope is that this common “vocabulary” will help to align the storage industry for the mutual benefit of all of its participants and customers.

Note that, although the model describes architectures, it is not itself an architecture. You cannot buy it or a system that it describes by specifying it in a bid or a request for a bid. You cannot “build it.” The model does not represent any value judgments between the architectures it describes. Instead, the model makes it possible to compare architectures and to communicate about them in a common vocabulary.

A Note on the Graphical Conventions Used in the Model

Throughout the model, we have tried to be consistent about the following graphical conventions:

- 3D objects represent physical entities (hosts, switches, etc).
- 2D objects with drop shadows represent functional entities.
- Orange/ochre represents storage devices.
- Dark blue represents host computer systems.
- Pale blue represents storage networks.
- Green represents file-level entities.
- Yellow is used for caches.
- Thick lines represent the major data transfer paths; thinner lines with arrowheads represent paths over which metadata flows.
- Diagonal stripes indicate metadata managers.
- The vertical placement is also important, particularly in the block subsystem layer.

The Classic Storage Model

All too often, the picture shown here (Figure 13-6) represents the current state of conversations about storage networking: vendors, system designers, and customers try to describe what they want—or what they have—using a set of inconsistent, ad hoc languages.

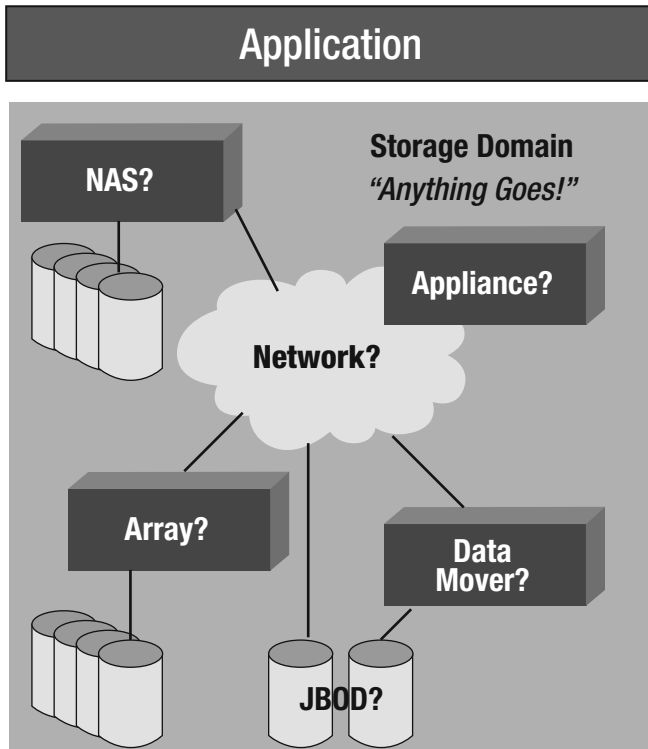


Figure 13-6. *The classic storage model*

Things are made worse by there being a great many network storage components, with relatively small differences between them. This causes designs that are actually the same to be described in different ways and different designs to be described sometimes using identical forms of words. This is clearly undesirable and results in many problems: it's often not obvious what is being proposed or described, trade-offs between alternatives are harder to identify than they could—or should—be, and it's harder for everybody to make high-quality decisions.

These confusions are not accidental: the wide variety of the range of system architectures that have been developed exhibit a great deal of complexity because they are trying to accommodate a great deal of information and cover many different elements and functions. Some of those elements are physical—boxes, wires, computers—and it is often the case that architectures are presented by describing the physical components in some detail, coupled with an explanation of what functions they perform. That is, the traditional approach focuses first on the physical partitioning that a particular vendor has selected, rather than on the range of options that may be possible. And because this is “box-centric” rather than “function-centric,” it is all too easy to misunderstand precisely what has been included.

The SNIA Shared Storage Model is an approach to removing these difficulties. It does so by taking a slightly different approach: it first identifies the functions that can be provided and then describes a range of different architectural choices for placing those on physical resources. As a result, the SNIA Shared Storage Model makes it easier to compare alternative architectures and designs, it lets architects think about functions independently of implementations, and it makes it simpler to anticipate new implementations or combinations of architectures, designs, and implementations.

The SNIA Shared Storage Model

The SNIA Shared Storage Model is shown in Figure 13-7.

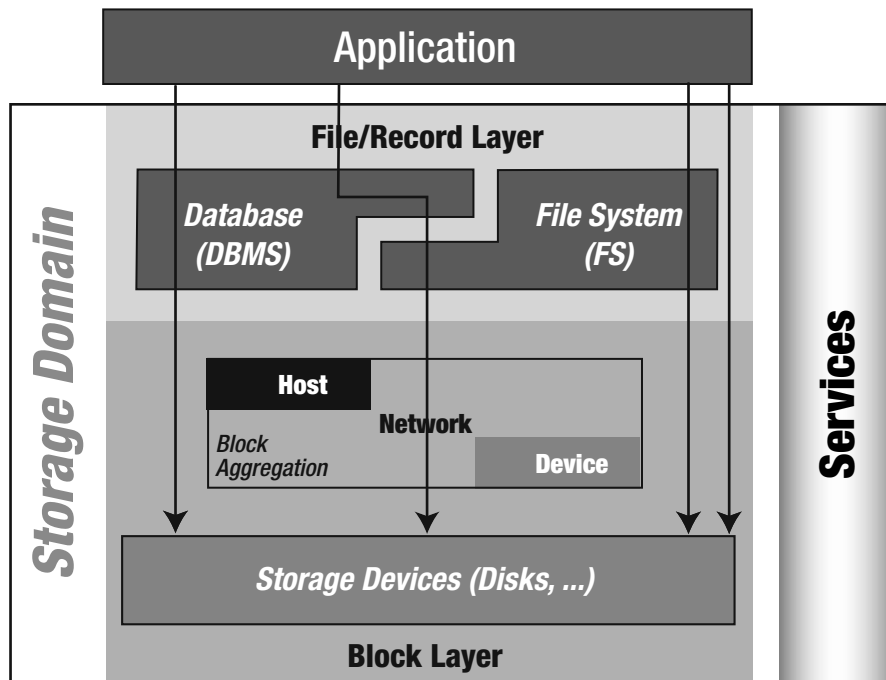


Figure 13-7. *The SNIA Shared Storage Model*

This is the highest-level picture of the SNIA Shared Storage Model. It has three main components within its scope:

- The **file/record layer**, which includes databases and file systems
- The **block layer**, which includes both low-level storage devices and block-based aggregation
- A **services subsystem**, which provides functions such as the management of the other components

Note that applications lie outside the scope of the model—they are viewed as “clients” of the storage domain in the broadest sense.

Storage System Components

The SNIA Shared Storage Model supports the following kinds of components:

Interconnection network: The network infrastructure that connects the elements of the shared storage environment. This network may be a network that is primarily used for storage access or one that is also shared with other uses. The important requirement is that it provides an appropriately rich, high-performance, scalable connectivity upon which a shared storage environment can be based.

The physical-layer network technologies that are used (or have been used) for this function include Fibre Channel, Fast- and Gigabit-Ethernet, Myrinet, the VAX CI network, and ServerNet. Network protocols that are used at higher layers of the protocol stack also cover a wide range, including SCSI FCP, TCP/IP, VI, CIFS, and NFS.

Redundancy in the storage network allows communication to continue despite the failure of various components; different forms of redundancy protect against different sorts of failures. Redundant connections within an interconnect may enable it to continue to provide service by directing traffic around a failed component. Redundant connections to hosts and/or storage enable the use of multipath I/O to tolerate interface and connection failures; multipath I/O implementations may also provide load balancing among alternate paths to storage. An important topology for multipath I/O uses two completely separate networks to ensure that any failure in one network cannot directly affect the other.

Host computer: A computer system that has some or all of its storage needs supplied by the shared storage environment. In the past, such hosts were often viewed as external to the shared storage environment, but we take the opposite view and will show examples of function mappings that place key components in such hosts.

A host typically attaches to a storage network with a host bus adapter (HBA) or network interface card (NIC). These are typically supported by associated drivers and related software; both hardware and software may be considered part of the shared storage environment.

The hosts attached to a shared storage environment may be largely unaware of each other, or they may explicitly cooperate in order to exploit shared storage environment resources. Most commonly this occurs across subsets of the hosts (*clusters*). One of the advantages of separating hosts from their storage devices in a shared storage world is that the hosts may be of arbitrary and differing hardware architecture and run different versions and types of operating system software.

Physical storage resource: A nonhost element that is part of the shared storage environment and attached to the storage network. Examples include disk drives, disk arrays, storage controllers, array controllers, tape drives and tape libraries, and a wide range of storage appliances. (Hosts are not physical storage resources.) Physical storage resources often have a high degree of redundancy, including multiple network connections, replicated functions, and data redundancy via RAID and other techniques—all to provide a highly available service.

Storage device: A special kind of physical storage resource that persistently retains data.

Logical storage resource: A service or abstraction made available to the shared storage environment by physical storage resources, storage management applications, or a combination thereof. Examples include volumes, files, and data movers.

Storage management: Functions that observe, control, report, or implement logical storage resources. Typically these functions are implemented by software that executes in a physical storage resource or host.

The Layering Scheme of the SNIA Shared Storage Model

The SNIA Shared Storage Model is a layered one (shown in Figure 13-8). The figure shows a picture of the stack with a numbering scheme for the layers. Roman numerals are used to avoid confusion with the ISO and IETF networking stack numbers.

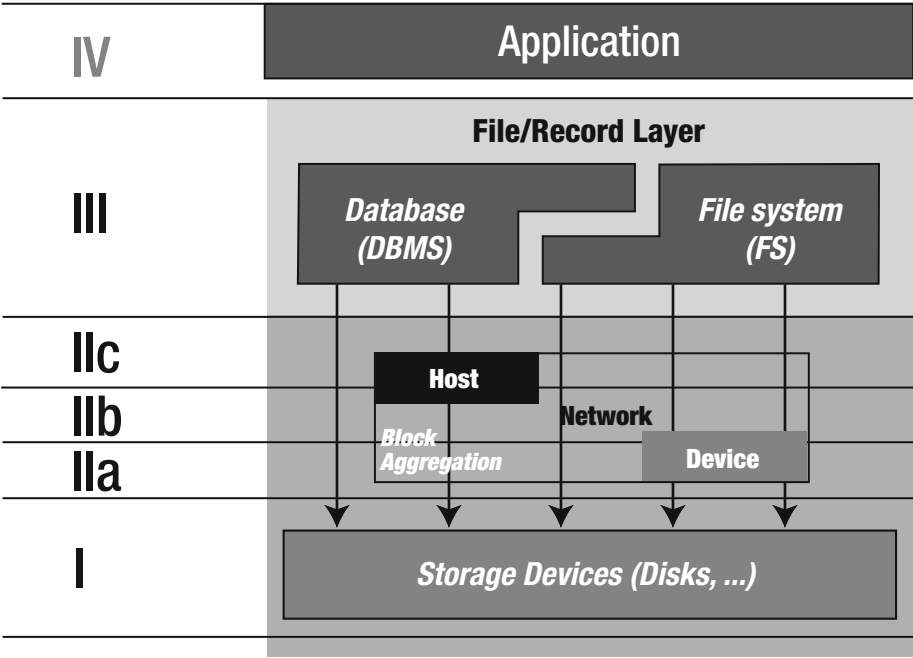


Figure 13-8. SNIA Shared Storage Model layers

The layers are as follows:

- IV. Application
- III. File/record layer
 - IIlb. Database
 - IIla. File system
- II. Block aggregation layer, with three function placements:
 - IIc. Host
 - IIb. Network
 - IIa. Device
- I. Storage devices

We will now describe each of these layers, from the topmost layer downwards.

The File/Record Layer

This layer (shown in Figure 13-9) packs small things such as files (byte vectors) and database tuples (records) into larger entities such as block-level volumes and storage device logical units.

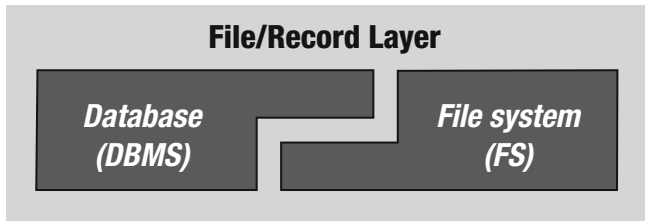


Figure 13-9. SNIA Shared Storage Model layers

The two most common implementations seen at this level are *database management systems* and *file systems*. Both provide mechanisms for naming or indexing files or records, enforcing access controls, performing space allocation and clustering, and caching data for rapid access.

In both cases, the file/record layer sits on top of one or more *volumes*: large block vector stores or byte vectors provided by an underlying block store or (sometimes) file system. That is, database management systems typically offer mappings (or packings) of

- Tuples or records → tables → volumes

Sometimes additional intermediate mapping layers are introduced, such as a grouping of tables together into a *table space* that sits atop one or more external volumes. This is usually done to make it easier to manipulate such mappings from inside the database management system in an environment-independent fashion.

File systems typically do mappings from

- Bytes → files → volumes

Because a byte vector can be used to emulate a block vector, the volumes that a database is mapped to can sometimes be files. This is most often done for small database systems where the performance penalties of the two levels of mapping it entails are outweighed by the simpler management that results from exploiting the naming and access control mechanisms offered by the file system.

Secondary functionality provided by this layer may include content indexing, aggressive prefetching, and write-behind techniques to improve performance, improve hierarchy management, and provide coherency across multiple copies in distributed systems.

In the future, we expect to see new implementations at this layer, such as file systems explicitly designed for replaying isochronous streams against multimedia objects (e.g., videos). Indeed, an HTTP Web cache might be considered a new kind of distributed file system.

Where Can It Be Done?

The functions provided by the file/record layer can be implemented in several different places (see Figure 13-10):

Solely in the host: These are the traditional host-based file systems and databases (the left-hand column in the figure shown here). In such systems, the implementation of the file system or database resides completely in the host, and the interface to the storage device is at the block-vector level.

In both client and server: These are the standard “network” file systems such as NFS, CIFS, etc. (The right-hand column of the figure.) Such implementations split their functions between the client (host) and the server system. (Note that this split happens *inside* the file system layer in the SNIA Shared Storage Model.) The client side always resides in a host computer. The server side, however, can reside in a

- **File server (sometimes database server):** Typically a host computer with local attached block storage devices that may be dedicated to this function
- **NAS head:** A dedicated-function computer acting as a file server and relying on external block storage devices connected through a storage network
- **Storage device:** Such as a disk array or “smart disk”

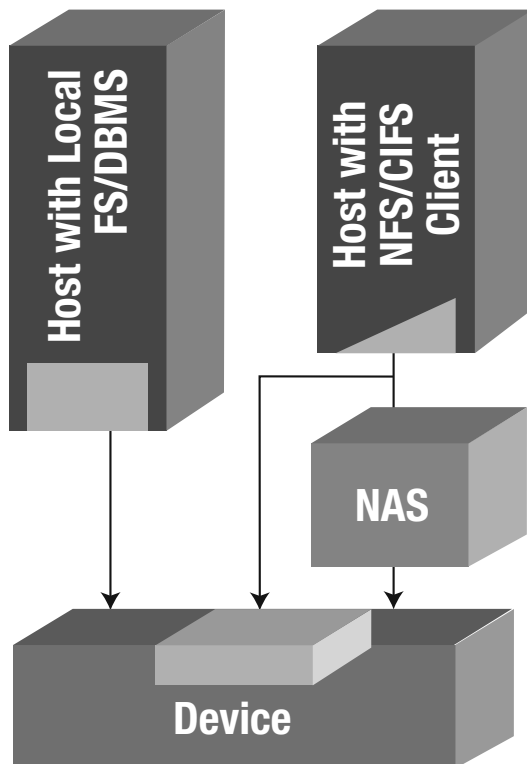


Figure 13-10. *File/record layer functions*

The Block Layer

The block layer (shown in Figure 13-11) provides low-level storage to higher layers, typically with an access interface that supports one or more linear vectors of fixed-size blocks. In SCSI, these logical address spaces are called *logical units* (LUs); a single SCSI storage device may support several such logical units.

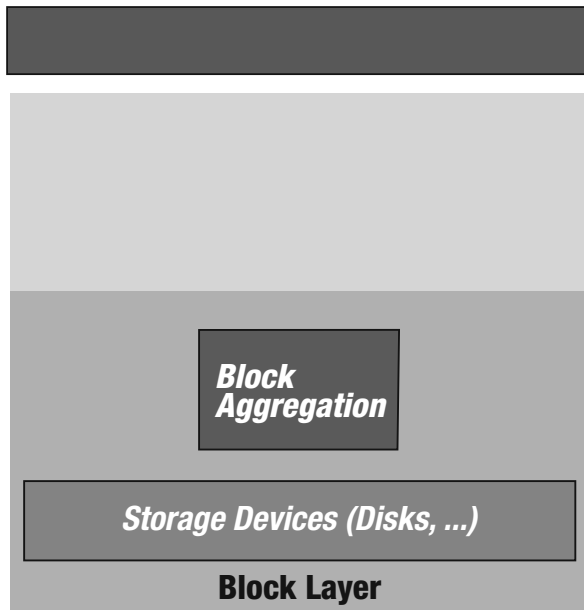


Figure 13-11. *The block layer*

Ultimately, data is stored on “native” storage devices such as disk drives, solid-state disks, and tape drives. These devices can be used directly, or the storage they provide can be *aggregated* into one or more block vectors to increase or decrease their size or provide redundancy. This aggregation can occur in many places—more on this below.

Secondary responsibilities of the block layer include a simple form of naming, such as SCSI logical unit names (LUNs), caching (and, in particular, nonvolatile caches for write-behind data), and (increasingly) simple access control.

Block Aggregation

Block aggregation comprises a powerful set of techniques that are used to serve many purposes. These include

Space management: Constructing a large block vector by assembling several smaller ones, packing many small block vectors into one large one, or doing both. (This “slicing and dicing” has historically been one of the most important functions of host-based logical volume managers.)

Striping: Apportioning load across several lower-level block vectors and the systems that provide them. The typical reason for doing this is to increase throughput by increasing

the amount of parallelism available; a valuable secondary benefit may be the reduction in average latency that can result as a side effect.

Redundancy: Providing redundancy for increasing availability in the face of storage device failures. This can be full redundancy (e.g., local and remote mirroring, RAID 1, RAID 10...) or partial redundancy (RAID 3, RAID 4, RAID 5, ...). Additional features like point-in-time copy (various versions of this are sometimes called *snapshot*) can be provided, which can be used to increase the effective redundancy level of a system and to help recover from other kinds of failures.

In practice, several of these functions are often combined together. For example, a system that can handle striping is often also capable of performing mirroring (a combination sometimes referred to as RAID 10).

Where Can It Be Done?

The block aggregation functions can be performed at several of the storage components described in the model. Indeed, it is common to find more than one being used.

- On the **host-side**, such as in logical volume managers, device drivers, and HBAs.
- In **components of the storage network** itself, such as specialized *SN appliances*. In addition, some HBAs are better thought of as part of the storage network.
- And, very commonly, **in the storage devices** themselves. Disk array controllers (e.g., RAID) are classic examples of this. Modern disk drive controllers provide some level of this functionality too, such as a logical-to-physical block mapping for supporting sparing.

How Is It Done?

Figure 13-12 shown here offers a simple, visual model of how block aggregation operates. It also illustrates the reason that block-based aggregation functions at different components can be composed together.

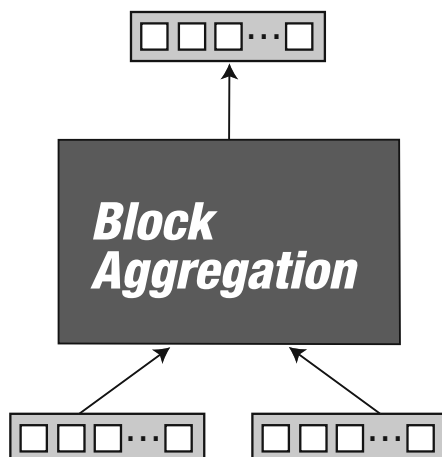


Figure 13-12. *Block aggregation operation*

You can think of each kind of aggregation function as a building block that imports (uses) one or more block vectors from “below” and exports to its clients one or more block-vectors at its upper interface that are constructed (i.e., *aggregated* or *virtualized*) from those imported ones. The construction can embody any or all of the functions described above.

Because the interfaces to both imported and exported block vectors are the same, these building blocks can often be stacked on top of one another; for example, mirroring across two disk arrays could be performed in a host logical volume manager, with RAID 5 redundancy applied within each of the disk arrays. Each layer could in theory also be internally constructed in this same way.

Sample Architectures

Figure 13-13 shows the first application of the model to a number of different block-based storage architectures.

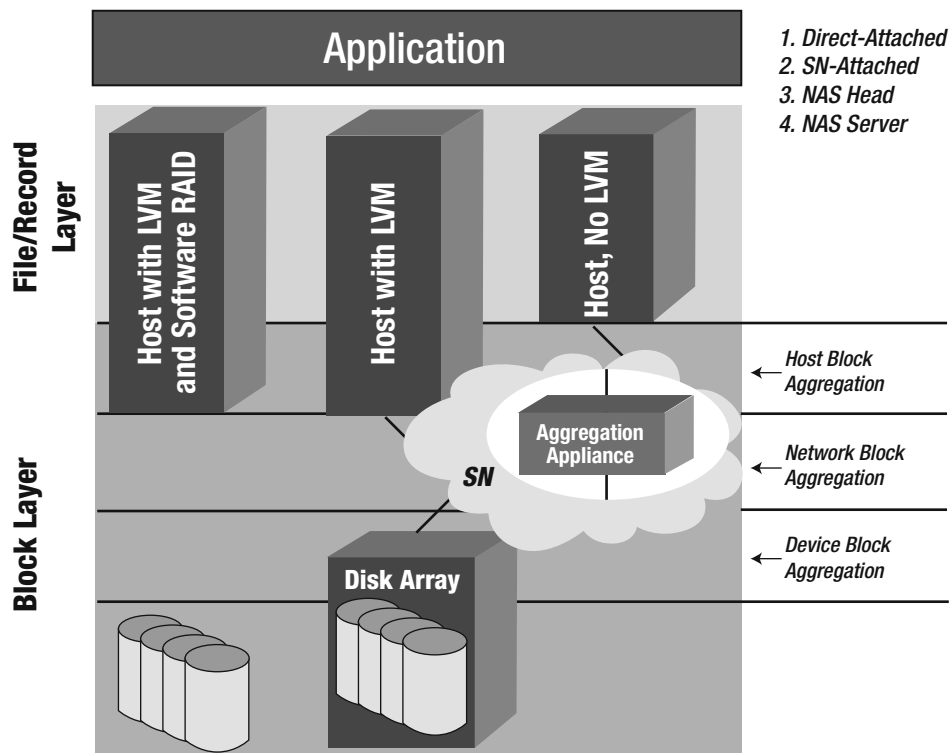


Figure 13-13. Application of SNIA Shared Storage Model

Direct-attach: The leftmost column illustrates this. A host is connected to some private, nonshared storage devices (e.g., disk drives attached by a local SCSI cable). In the figure, the host is shown running a logical volume manager (LVM)—perhaps capable of providing a software RAID function to provide protection against disk failure.

Storage-network attach: The second and third hosts, plus the storage network and disk array in the second column, illustrate this. This scheme introduces a storage network (the pale blue cloud) connecting one or more hosts—perhaps still running LVM software—to a disk array that is providing a further set of block aggregation functions. The disk array resources can now be shared between multiple hosts.

Storage-network aggregation: The final example embeds a block-aggregation function into the storage network in an aggregation appliance that might be providing access control and (say) striping aggregation functions.

Putting It All Together: Combining the Block and File/Record Layers

This picture shown in Figure 13-14 puts together both block-based and file-based storage architectures.

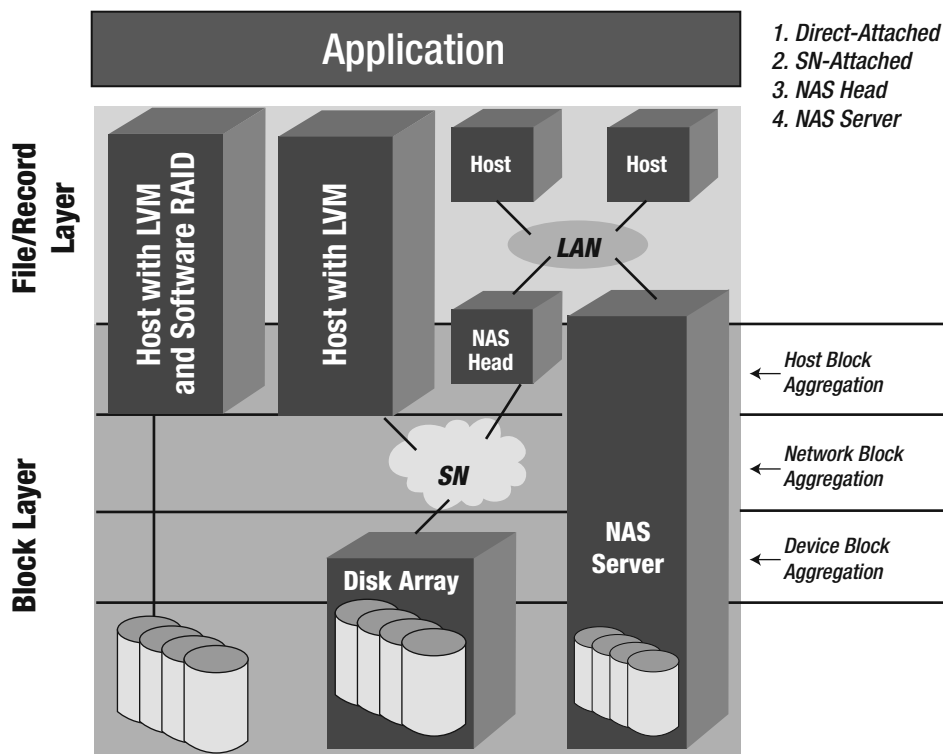


Figure 13-14. Combining block-based and file-based storage architectures

Direct-attach: The leftmost column shows local, private storage directly connected to a single host.

Storage-network attach: The second column shows a representative host connected to a (shared) disk array through a storage network (SN).

NAS head: The third column illustrates a dedicated-function *NAS head* (file server controller) interposed between the lower-level, block-based storage network and its clients, which are connected to it through a second network (generically an arbitrary second storage network but shown here as a LAN, as that is the most common form) and operate using client-server file system protocols. Note that block-aggregation functions can be used to support several NAS heads, as well as regular block-level hosts.

NAS server: This is shown in the rightmost (fourth) column and logically consists of a combined NAS head and its own local, private storage.

Access Paths

An *access path* (shown in Figure 13-15) is the list of components (hardware and software) that are traversed by read and write requests to the storage system and their responses. If we restrict ourselves to avoid cycles, there are eight possible paths from the application layer to the lowest storage layer through the elements of the SNIA Shared Storage Model.

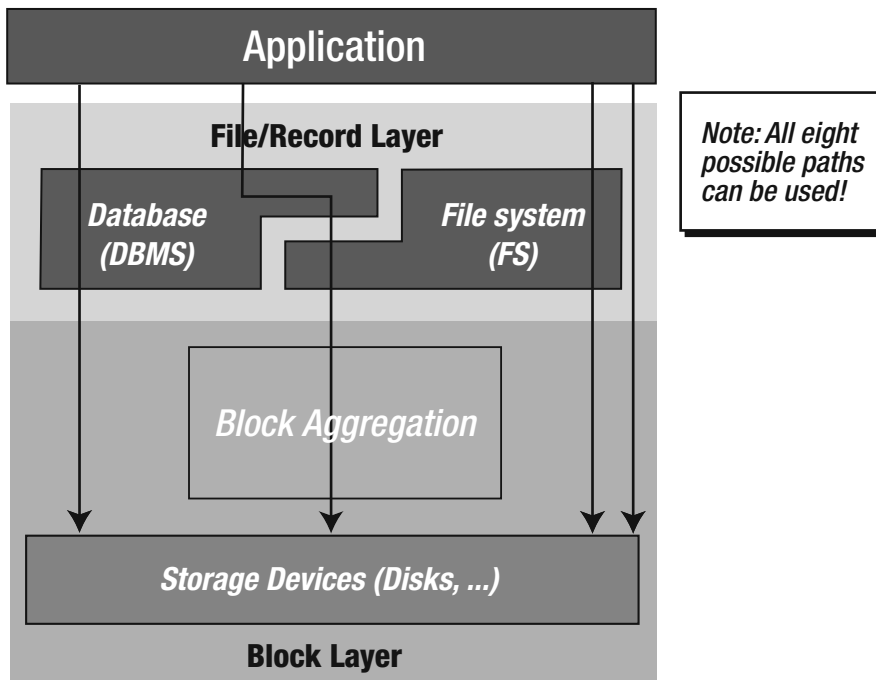


Figure 13-15. Access paths

Five examples of common paths are shown in the figure here, reading from right to left:

- Direct to the storage device (e.g., disk drive or disk array)
- Via a file system
- Via a file system that sits on top of a block aggregation function
- Via a database on top of a file system on top of a block aggregation function
- Via a database

Caching

Caching (shown in Figure 13-16) is designed to shorten access paths for frequently referenced items and so improve the performance of the overall storage system. Most elements of a storage system can provide a cache, and so such caches can be performed at the block or file/record layer—or both. Indeed, it is common to see several caches in operation simultaneously, for example a read cache in a file system, coupled with a write-back cache in a disk array, and a readahead cache in a disk drive.

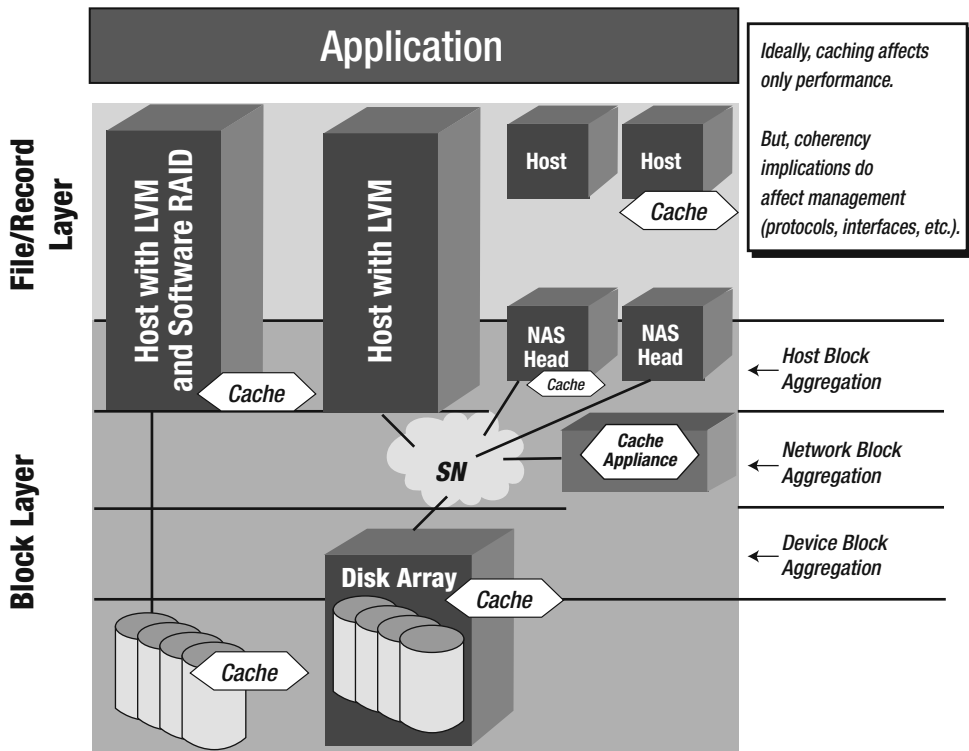


Figure 13-16. Data caching

The figure here illustrates this: almost *any* of the components in the system can be augmented with a cache. The figure also introduces a new component: a dedicated *caching appliance*, added to the storage network solely to provide caching functions.

Ideally, all that adding caching does is speed things up, but making this true for shared storage networks requires taking account of a multitude of detailed issues that occur in distributed systems, such as maintaining the coherency of multiple copies of data or metadata (data about data) and tolerating partial failures of the system. In particular, cache management (deciding which cache should—or does—hold what) is significantly more complicated when data may be cached in several places.

Access Control

A shared storage environment bestows the benefit of hosts being able to access their storage devices directly. But the *existence* of an access path should not be taken as equivalent to *permission* to exercise it. Access control is the set of techniques that enforce the decisions that encapsulate these permissions.

There are many different kinds of unwanted accesses possible, and the protection used against them has perforce to trade off the degree of protection against efficiency and the complexity of enforcement. The basic goals are to provide

- **Authentication:** “Proof that I am who I say I am”
- **Authorization:** “Proof that I am allowed to do this”
- **Privacy:** “Proof that I am allowed to see the contents”

Historically, storage systems have provided little or no support for any of these, other than via simple physical security—locking up the storage devices and the hosts that access them. This is likely to change significantly in the storage network world because the number of different threat types is so much larger.

Ultimately, all approaches to access control rely on some form of secure channel being established between the provider of data (or operations on that data) and its destination. The least secure, but also easiest to implement, solution imposes simple, coarse-grained accessor checks (of the form “is this host permitted to send requests to this storage device?”); at the other extreme lies cryptographic protection mechanisms that are resistant to a wide variety of impersonation, monitoring, and replay attacks and capable even of securely storing data on storage devices that cannot be trusted not to divulge (or lose) their contents.

Preventing unwanted accesses can be performed in several places:

At the host: This offers convenience of implementation, easy scalability, low execution cost, and potentially fine-grained control, but it must be pervasively deployed, and the enforcement mechanism should be resistant to tampering. With support from their host operating systems, file systems and databases commonly enforce access controls on data, and similar kinds of solutions are appearing at the block vector layer.

Networking stacks in the host can also use encryption to provide secure channels across various forms of network (e.g., IPsec). The performance of software versions of these schemes means today that they are best suited to use on relatively low-speed network links (such as a wide area network), but this is likely to change with the deployment of hardware accelerators.

In the storage network: Today, this is largely restricted to relatively low-level approaches that offer the illusion of private, dedicated subnetworks that permit a set of host and storage device ports access only to one another, hiding any other ports. (These are usually called *zones* in Fibre Channel or VLANs in the Ethernet world.) Because they operate on whole ports, such solutions are quite coarse-grained.

But, by analogy with the migration of functions such as load balancing into traditional IP networking components, it is reasonable to expect finer-grain controls appearing in storage network switches, such as the ability to enforce such virtual networks on per-logical-unit (LU) boundaries.

At the storage devices: Ultimately, storage devices will probably have to accept as much responsibility for enforcing access controls as the hosts do; they are, after all, the primary shared resource that storage networking is trying to make available. This has long been true of servers at the file/record layer such as file servers and NAS heads, and now solutions are appearing that perform a simple form of *LU masking* at the block layer, where only certain hosts (or host ports) are allowed access to a particular LU.

As storage networks span ever-greater numbers of devices and encompass greater heterogeneity of host types, host software, and distances, the importance of this issue will greatly increase.

The Services Subsystem

A very important part of the model is the set of services that lie “off to one side” of the critical data-access paths. The list provided in the graphic here is not meant to be exhaustive, but to give a flavor of the kinds of things that are handled by this *services subsystem*. Many of the services are “management” tasks and need to be tied into the larger systemwide service management tool processes and tools (see Figure 13-17).

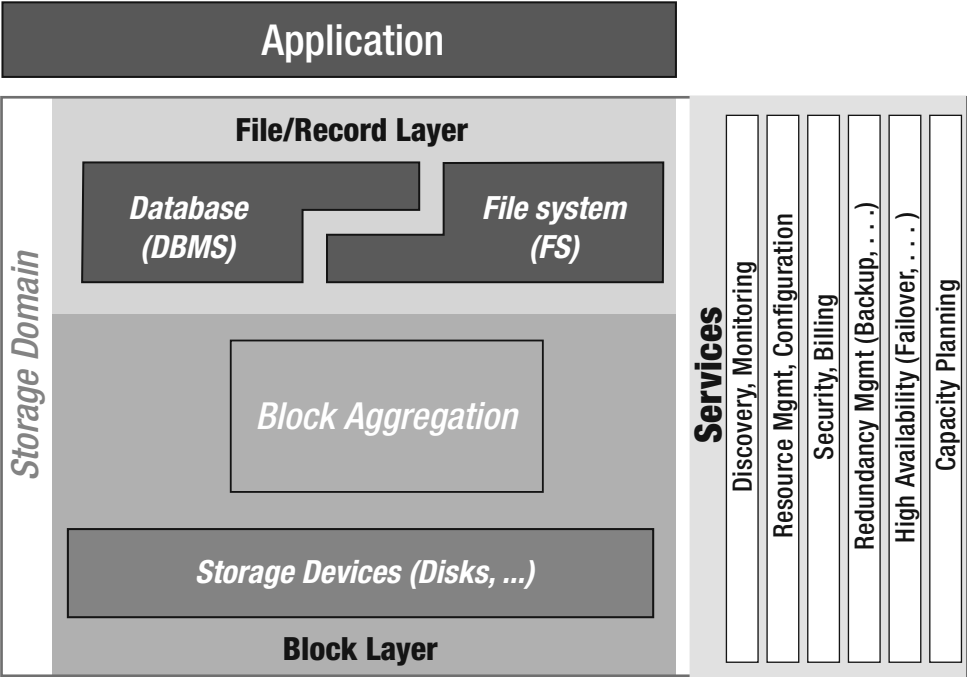


Figure 13-17. *The services subsystem*

Although such services are vital for successful implementations, they are not further discussed here; this version of the SNIA Shared Storage Model deliberately focuses on the data-access portion in order to allow us to communicate the model for discussion and use in a timely manner.

Refining the definitions, terminology, specifications, and interfaces associated with the services subsystem represents a major opportunity for the SNIA community. This issue will be explored in detail in future SNIA Technical Council reports.

Note The previous sections came directly from the SNIA Technical Council's *Shared Storage Model: A Framework for Describing Storage Architectures* (SNIA, 2003), pages 10–21. Now that you've seen the SNIA perspective on shared storage networking, it's time to examine the techniques used to apply this methodology.

Applying the SNIA Shared Storage Model

With an understanding of the SNIA Shared Storage Model under your belt, you'll now look at how the architecture is hitting the pavement. In the following sections, we'll cover the evolution of the SNIA architecture, as well as the terminology that it maps to in today's IT environments. You can design virtualization as a host-based, storage-based, or network-based architecture. We describe these architectures in the next three sections.

Understanding Host-Based Architecture

In the host-based architecture, storage virtualization is provided at the host level. This architecture has actually been in use for decades, but the term *virtualization* has rarely been used to describe it. Figure 13-18 shows a typical host-based virtualization architecture.

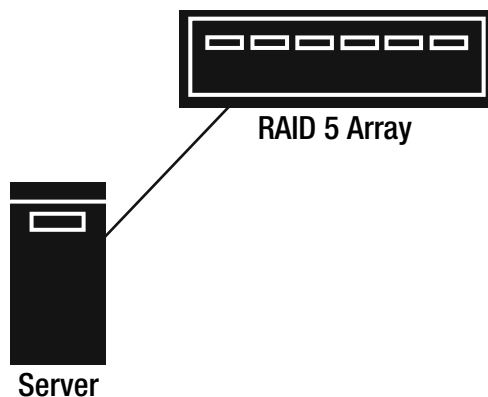


Figure 13-18. *Host-based virtualization used for RAID configuration*

With host-based virtualization, you can make a single physical hard disk appear as multiple logical hard disks by partitioning. Also, with the use of software RAID at the host level, you can configure several physical disks to appear as a single logical disk. The tools to perform this type of virtualization are embedded in most operating systems, such as Microsoft's Disk Management utility that allows you to partition disks and create software RAID volumes. You can also configure host-based virtualization by using hardware devices such as RAID controllers or other disk applications such as Veritas Volume Manager. When considering virtualization, host-based virtualization has a proven track record and is certainly the most widely implemented form of virtualization to date.

Aside from providing for a one-to-one relationship between storage and a host, host-based virtualization software can also allow several hosts to share physical storage. This is commonly accomplished with the aid of a SAN; Figure 13-19 illustrates this concept.

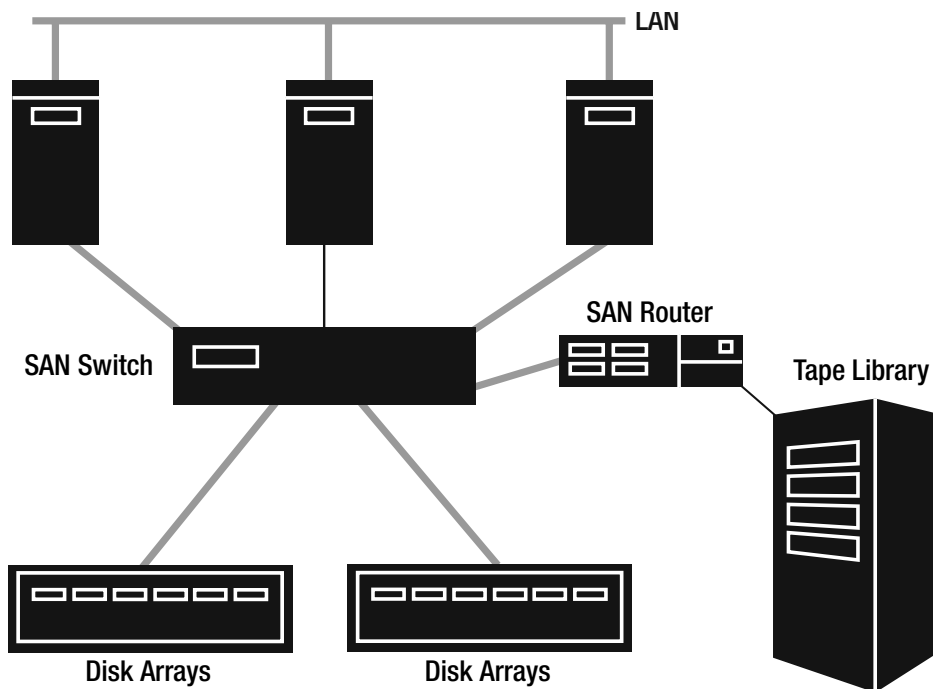


Figure 13-19. *Sharing physical storage devices with a SAN*

In the example shown in Figure 13-19, all hosts could potentially access storage resources on the SAN. This approach allows storage to be pooled and distributed amongst multiple hosts, but in this architecture the data on the SAN is still host-centric.

Understanding Storage-Based Architecture

With storage-based virtualization, the virtualization intelligence is provided at the storage device level. For example, consider using a hardware RAID controller. With this device, the intelligence of the storage controller allows you to configure the disks in the attached RAID array to appear either as a single disk or as a collection of independent disks. If your needs change, you can always reconfigure the disks using the disk controller's management software, which is likely loaded in the controller's BIOS. Storage-based virtualization is often proprietary to the storage device itself and thus normally doesn't provide a means to centrally pool and manage storage. This form of virtualization is often used when configuring storage devices for single servers or for clustered servers that share physical disks in an external storage array.

Understanding Network-Based Architecture

Network-based virtualization places the virtualization intelligence on the storage network. In offloading the virtualization software from the host, host performance won't suffer from the virtualization usage. However, this approach will add to the complexity of the virtualization solution. Network-based virtualization does, however, offer numerous advantages:

- Allows you to divide portions of physical disks to be used as individual logical disks by different servers
- Allows you to dynamically pool and allocate data to all servers on the SAN
- May provide the ability to dynamically change the size of pooled storage resources as needed
- Allows you to back up and recover pooled data as a whole without impacting the data's host servers (commonly known as *serverfree backup*)

Although this virtualization architecture offers several advantages, it has one primary disadvantage as well. When multiple servers need to collaborate with a virtualization device, some degree of latency is inevitable. Both hardware and software vendors alike are working to combat this problem in order to ensure the future of their virtualization investments. If all virtualization vendors use the SNIA Shared Storage Model, this will go far in ensuring compatibility between their virtualization devices.

When you design and deploy a network-based virtualization solution, you can configure it in one of two different ways: in-band or out-of-band.

As you'll see, the difference between in-band and out-of-band architectures lies in the location of the device providing the virtualization intelligence. This device is often referred to as the *virtualization appliance*. Now let's look at the differences between in-band and out-of-band network-based virtualization.

In-Band

With in-band network-based virtualization, the virtualization appliance resides directly in the data path between the servers and storage devices. Figure 13-20 shows this concept.

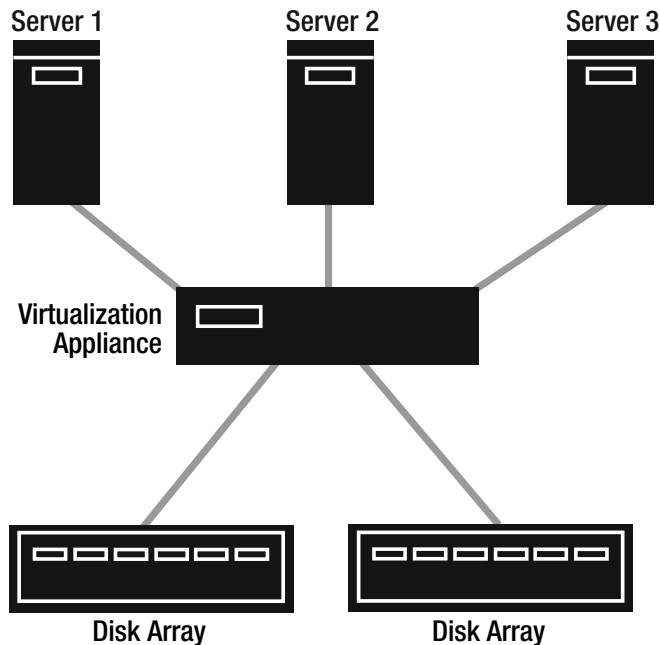


Figure 13-20. *In-band network-based virtualization*

In this approach, the virtualization appliance could even be a SAN switch, or the appliance could be between the servers and the switch or between the switch and storage devices. With the virtualization software off the host systems, storage access on the SAN is transparent. The systems will see what the virtualization appliance presents to them.

To accomplish this, the appliance will take disk requests from the host systems and forward the requests to the appliance's attached storage devices. As far as the host systems are concerned, they see the virtualization appliance as storage. Since the appliance presents itself as a storage device to hosts, you can quickly connect several different disks to a host without having to worry about drivers for each storage device. As long as the host can see the virtualization appliance, it will be able to see the drives that the appliance presents to it.

Two major storage vendors, Cisco Systems and Brocade, have produced SAN switches that have this level of capability. Cisco has also partnered with IBM to produce IBM's Total-Storage SAN Volume Controller, which provides the following features:

- Allows you to pool all storage resources on a SAN and logically divide and allocate storage to servers as needed
- Automates backups and data copy movements transparent to the servers on the network
- Provides free storage that can be reallocated as needed

Historically, chief information officers (CIOs) and network administrators have dealt with storage requirements by simply “throwing more storage” at a problem. This approach has often resulted in wasted storage for organizations because they often wind up with storage islands, where some servers have overallocated storage while others have underallocated storage. By allowing you to pool and share all physical storage resources, properly allocating storage resources is no longer an issue.

Out-of-Band

With in-band virtualization, the virtualization appliance is directly in the data path between systems and storage. Out-of-band virtualization uses exactly the opposite architecture. With out-of-band network-based virtualization, the virtualization appliance is completely out of the data path between the servers and physical storage devices. Figure 13-21 shows out-of-band virtualization.

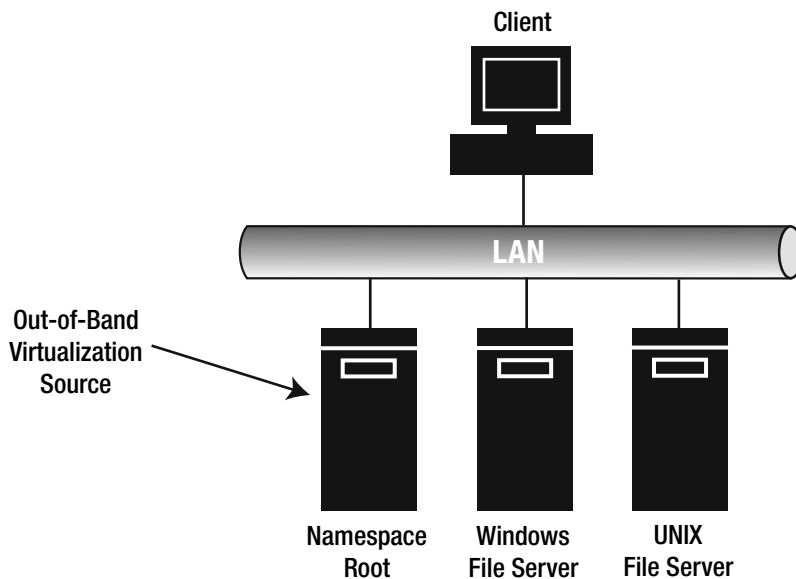


Figure 13-21. *Out-of-band network-based virtualization*

Figure 13-21 shows one of the most common implementations of out-of-band virtualization. Assume in the figure that clients are accessing all file server resources through a single server designated as the namespace root. The root server will transparently redirect clients to the appropriate physical storage location. You can implement this form of virtualization with DFS, with AFS, or with a commercial product such as NuView StorageX (<http://www.nuview.com>). For more information on DFS or AFS, refer to Chapter 8.

Adding Fault Tolerance to the SAN

With SANs, having data collectively together on the same network allows you to take full advantage of all of virtualization's possibilities, but it does so at a high cost of needed SAN devices and complexity. For virtualization to fully support fault-tolerant data access, one common technique with storage area networks is to build out redundant paths to access data. This technique is commonly known as *multipathing*. You can do this by doubling the switches, cables, and HBAs installed in the systems attached to the SAN. With redundant data paths, if a failure occurs on the SAN, each server should have another path in which to access its data.

The problem with redundancy is that it's effective only if the operating system can access data on an alternate path if the primary access path fails. For systems to access data via multipath paths, they must have multipath drivers installed. For systems that access data via a virtualization applications, they too will need the correct virtualization-specific multipath drivers installed. With multipathing incorporated and the correct drivers installed, you can configure all aspects of virtualization to provide transparency to data access on the SAN and to provide a means by which to allocate storage on the SAN.

Performing Backups

A major problem with backups has historically been needing to know exactly what tape a backup is on in order to run a restore. While this may not be a big deal for a small organization, it can be challenging in an enterprise where a night's worth of backup jobs may consume 20–100 tapes. Now imagine if you had to restore data that was backed up four months ago. This may mean that the backup could be anywhere within 1,000 or more tapes! To answer this dilemma, two major backup software vendors, Veritas (<http://www.veritas.com>) and CommVault (<http://www.commvault.com>), have been steadily developing policy-based data management. With this approach, policies map systems to storage and serve as a layer of intelligence between a server and a storage resource. Ultimately, this approach means that for you to do a backup, you don't need to designate a storage device as the backup target. This was already set when you would configure the server's associated storage policy.

Now when it's time to do a restore, you can just browse the available data at a point in time on a server (without needing to know which tape to browse!), pick out the files to restore, and click the Restore button. If any offline media is needed to complete the restore job, the backup software UI will tell you! How is that for convenience?

Introducing Hierarchical Storage Management

Hierarchical storage management (HSM) logically expands a physical disk's size by migrating old infrequently accessed files to another storage device, such as a tape library. With HSM, when a file is migrated off of a hard disk, a *stub file* is left behind. The stub will have the same name as the original but will contain an embedded pointer, often referred to as a *reparse point*. The size of the embedded pointer amounts to about 4KB, so you could have a file that was originally 200MB in size reduced to 4KB on a server's hard disk. When a user attempts to access the file, the server will automatically retrieve the file from the remote storage device, and it will load on the user's system. Since the file is being retrieved from most likely a slower form of storage, some degree of latency will occur.

Right now, you may be thinking, “Sounds great! But why would I need it?” If that’s the case, here are some common reasons for implementing an HSM solution:

- You must maintain a high level of files available in a near-line status for review purposes (such as legal contracts), and it isn’t feasible to dedicate hard disk space for the amount of data that would be required to keep everything online.
- You want to archive data for future auditing purposes.
- You want to address storage growth with a less expensive form of storage (tape) as opposed to continually purchasing or upgrading magnetic hard disks.
- As a research center, you may want to keep all collected documents available in a near-line capacity, allowing all collected data to be constantly available.

Again, HSM allows you to migrate infrequently accessed files to an alternate form of storage. You can usually accomplish this using a tape library. So, for example, a tape library may offer 2TB of storage capacity, and your server may have 200GB of hard disk space. With HSM, the server could potentially provide access for up to 2.2TB of storage!

Remember, users will have to wait while the actual file is being retrieved. Some HSM products offer a pop-up window to alert the user of the delay, while others don’t. This could mean an impatient user could repeatedly double-click the same file in an attempt to open it, resulting in numerous recovery jobs being launched. So, when considering HSM, user notification is often a key concern. If you’re looking for more information on HSM, here are some of the major HSM product vendors:

- **CommVault QiNetix DataArchiver:** <http://www.commvault.com>
- **Veritas NetBackup Storage Migrator:** <http://www.veritas.com>
- **Microsoft Remote Storage Service (RSS):** <http://www.microsoft.com>
- **EMC Legato NetWorker HSM:** <http://www.legato.com>

While tape libraries aid storage virtualization in the form of HSM, virtualization also provides additional flexibility for libraries. We’ll cover that next.

Using Virtual Tape Libraries

Virtual tape libraries provide a means to create logical tape libraries either from magnetic disk drives or from an actual physical tape library. With the magnetic disk approach, virtual tape libraries present a library entity to storage applications such as backup software. The virtual library comes complete with virtual tape devices, which amount to nothing more than files on a hard disk. In using virtualization to divide a physical library, you can allocate portions of a single library to several servers or applications that normally may require their own dedicated library.

Dividing Physical Libraries

Oftentimes, administrators have to manage several applications that all want to “own” a tape library. While application vendors don’t see this as a problem, many administrators do. Tape libraries are expensive! So, dedicating physical libraries to a single backup or HSM application isn’t always practical. To get around this, many organizations have turned to virtualization to allow a single physical tape library to appear as a collection of several logical tape libraries. This concept has been perfected by StorageTek (<http://www.storagetek.com>) with their Automated Cartridge System Library Software (ACSL), which has become the industry standard for tape library virtualization. This type of virtualization allows you to share tape backup drives among multiple servers on the network and allocate a specific portion of a library’s media to each server. With tape library virtualization, you can more easily manage storage by consolidating what may at one time have been several independent stand-alone tape drives into a single tape library attached to one server.

Writing to Magnetic Disk

One other type of tape storage virtualization is the use of virtual tape libraries and media. With virtual tape storage, you can create a logical library with as many drives as you want and create as many virtual tapes as you want in the library. The library will behave just like any physical library and will mount, unmount, and even rewind virtual tapes. This virtualization solution is useful for providing for fast media write access for backup applications that natively support backing up to tape devices only, for example. One popular product that supports virtual tape devices is HP’s Virtual TapeServer. For more information on this product, visit <http://www.hp.com>. Another popular virtual tape server appliance was developed by Network Appliance (<http://www.netapp.com>). This solution allows you to configure a Network Appliance NAS filer as a target tape backup library. The library emulation is provided by a FalconStor IPStor Virtual Tape Library (VTL). Figure 13-22 shows this configuration. With the FalconStor VTL solution, you can emulate both LTO-2 and DLT8000 drives.

Keep in mind that with a virtual tape library storing data on magnetic disk arrays, you can still use your backup software to make an alternate copy of data to actual tape. This allows you to have a disaster recovery (DR) copy of data. With a backup to magnetic and then to tape, you can get the best of both worlds with backups and restores. Backups to magnetic arrays are usually much faster than backups to tapes. This allows you to more easily fit your backup jobs into your backup window. In copying data to tape, you still have offsite DR protection. We’ll cover these scenarios in more detail in the next chapter.

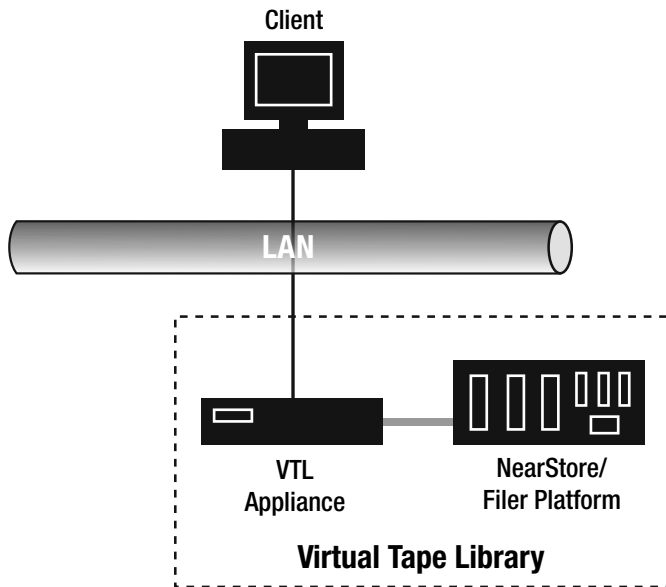


Figure 13-22. *Network Appliance: FalconStor VTL solution*

Summary

This chapter has taken you on a journey through the world of storage virtualization. We went from RAID to in-band virtualization in a SAN, jumped over to HSM, and wrapped up with tape libraries. As you've seen, plenty of possibilities exist. Before jumping into untested water, keep in mind that many of the solutions we discussed were provided by third-party vendors. Therefore, always look to ensure that any third-party product is already compatible with the hardware and software in your specific environment before virtualizing.

So far, we've conquered plenty of virtualization technologies on a one-by-one basis. In the next chapter, we'll tie them all together and provide examples of how to integrate the vast array of virtualization products in any network environment.



Putting It All Together: The Virtualized Information System

Now that you've seen the many facets of virtualization, it's time to put them all together. In this chapter, you'll examine some of the architectures present in today's virtualized networks and look at some new techniques as well. We'll start the chapter by recapping the major elements of the virtual information system (IS), and then we'll cover how virtualization assists data management. We'll then wrap up the chapter by covering how to add fault tolerance to your virtual machine infrastructure by maintaining standby VM host servers. Let's get started with a look at a truly virtualized information system.

Reviewing the Elements of the Virtual IS

Figure 14-1 shows a LAN with several virtualized resources.

Notice that the following technologies are deployed in the sample network:

- Failover cluster
- Load-balanced cluster
- Virtual machine host
- SAN
- DFS root

We'll quickly review the decision factors that surround each of these technologies.

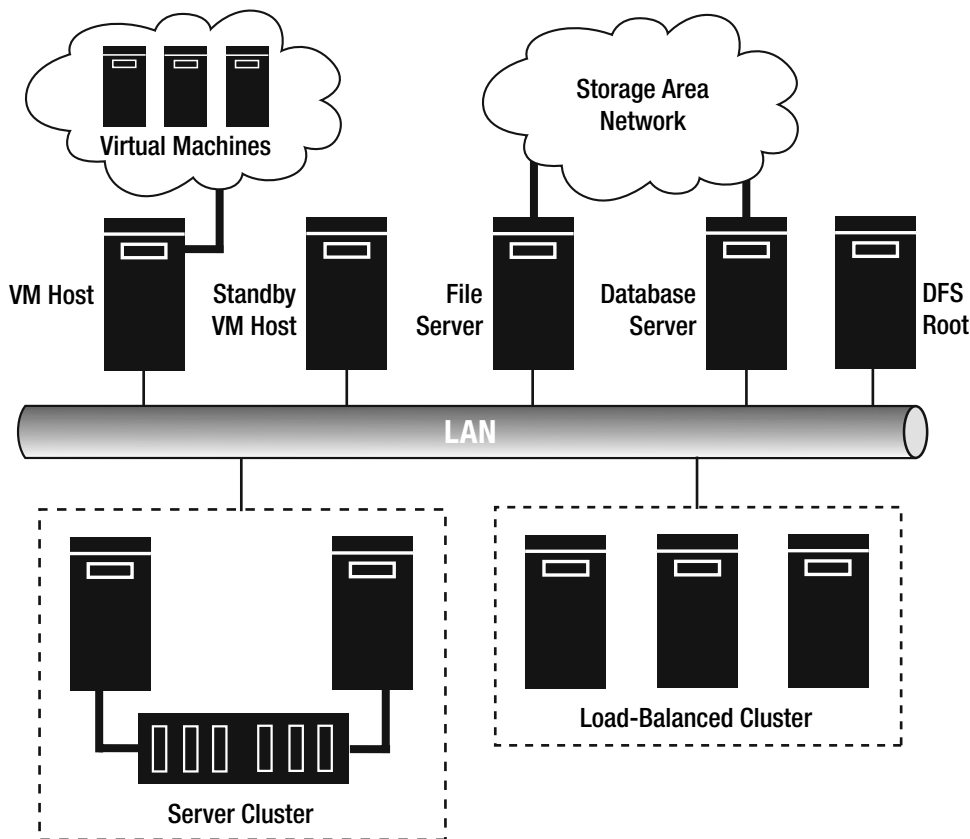


Figure 14-1. *Enterprise virtualization elements*

Failover Cluster

The failover cluster provides fault-tolerant access to read/write data. For this level of access, many large organizations employ failover clustering for e-mail servers such as Exchange 2003 servers or for database servers such as Oracle 10g.

Failover clustering solutions require you to configure shared storage for all cluster nodes as well as ensure your applications support failover in the first place. You can confirm an application's support for failover clustering through the application vendor. For shared storage, this will require an additional investment. The following are your three choices for shared storage:

- SCSI
- iSCSI
- Fibre Channel

If you decide to go with SCSI attached shared storage, the cluster will be limited to two nodes and won't be able to scale beyond that. iSCSI and Fibre Channel both support scalability up to eight or sixteen nodes, depending on the OS and application. So, if future scalability is a concern, then iSCSI or Fibre Channel is your best method for connecting the cluster storage. If you're looking to set up a simple failover cluster on a budget, then a SCSI solution is most likely what you need.

Now, if you're looking for either a Fibre Channel storage solution or an iSCSI storage solution for the cluster, then data throughput should be your primary concern when deciding on technologies. Ultra 320 SCSI hard disks, for example, offer 320MB/sec throughput. A 1Gb/sec iSCSI SAN, on the other hand, provides for 125MB/sec ($1,000\text{Mb} \div 8$) throughput. This is 60 percent slower than U320 SCSI. Now consider a 4Gb/sec Fibre Channel SAN. If you convert 4Gb/sec to megabytes, you wind up with 500MB/sec. So, this solution will offer performance that potentially exceeds that of SCSI.

Aside from bandwidth considerations, cost will also be a likely factor. Although SCSI solutions are the least expensive to implement, iSCSI would be the next in line. Since iSCSI leverages Ethernet technologies, you can use Gigabit Ethernet switches to drive the iSCSI storage network. As technology improves, you can replace the Gigabit Ethernet switches with faster ones and recycle the older switches into the production LAN. With Fibre Channel, you don't have this level of flexibility. A Fibre Channel switch can only ever be used with a Fibre Channel SAN. With Fibre Channel switches costing more than Gigabit Ethernet switches, Fibre Channel SANs are the most expensive to implement. However, Fibre Channel SANs offer additional features, such as potentially in-band storage virtualization, and additional data management features, such as SCSI-3 Extended Copy.

As you can see, the toughest design decision to make with failover clusters often involves how you should configure the cluster's shared storage resources. When designing for growth, it's always easiest to start with Fibre Channel or iSCSI and then add nodes as needed; starting with SCSI may result in having to rebuild a cluster from scratch.

Load-Balanced Cluster

Load-balanced clusters are most commonly used to provide fault tolerance and to distribute client access across several Web servers. Load-balanced clusters typically don't employ any form of shared storage, with each node maintaining its own copy of the cluster's data. From a management perspective, it will be up to you to maintain data synchronization between the nodes.

Load-balanced clusters are most frequently deployed as front-end Web servers, such as Apache servers. Also, many organizations use load-balanced clusters as front-end SMTP e-mail forwarders to offload tasks such as spam filtering from the production mail servers. FTP is another popular service configured in a load-balanced cluster.

When deciding on how to implement a load-balanced cluster, you can go in one of three directions:

- Hardware load balancer
- Windows NLB cluster
- LVS

With a hardware load balancer, the load-balancing functionality is transparent to the actual servers in the cluster. Instead, a hardware load balancer handles the load-balancing activities. The most popular vendor in the hardware load balancer market is Coyote Point Systems (<http://www.coyotepoint.com>). Coyote Point offers several hardware load balancers with pricing that starts at the typical cost of a single server.

Windows load balancing operates architecturally differently than LVS. With Windows NLB clusters, the total number of nodes required to create the cluster is equivalent to the number of desired nodes in the cluster. With LVS, load balancing is handled external to the cluster, meaning that two additional systems are required for a fault-tolerant Linux load-balanced cluster. Since the load-balancing operations occur transparently to the actual cluster nodes, applications and services on the cluster nodes don't have specific requirements to be able to participate in the LVS cluster.

Virtual Machine Host

The virtual machine host can service several VMs, reducing the amount of physical systems required to support your enterprise servers. In addition to hosting production VM servers, the virtual machine host can also provide avenues for testing, training, and help desk personnel. Support teams at some organizations run a workstation-class virtualization product such as VMware Workstation on their desktops. This allows them to open running environments for each different operating system they support, giving help desk workers the ability to easily walk users through problems with specific operating systems.

SAN Integration

The enterprise-class VM products offered by both VMware and Microsoft offer a great deal of flexibility in production VM deployment. VMware's ESX Server product has maintained its position as the leader in enterprise-class features, especially in terms of seamless SAN integration. Microsoft, with its Virtual Server product line, is working to close the feature gap.

Figure 14-2 shows a sample VMware ESX Server deployment. Note that the production host provides resources for two running virtual machines. Each VM directly attaches to the host's physical resources in the SAN. This allows you to back up VMs directly to the SAN, thus maximizing backup performance. To support this type of configuration, most backup vendors require that you run a backup agent as well as media server software on each VM. An alternative to this configuration is to configure the VM host as the media server and just run backup agents on each VM. With this approach, backup performance will likely be throttled by either physical or emulated network limitations. For example, earlier in this chapter, in the "Failover Cluster" section, you looked at the performance differences between 100Mb Ethernet and 4Gb Fibre Channel. Remember that 100Mb Ethernet bottlenecks at 12.5MB/sec, whereas 4Gb Fibre Channel bottlenecks at 500MB/sec. While bridging the VMs to a 1Gb LAN can aid in performance (125MB/sec maximum), it will still not equal the potential throughput of directly bridging the VMs to SAN resources.

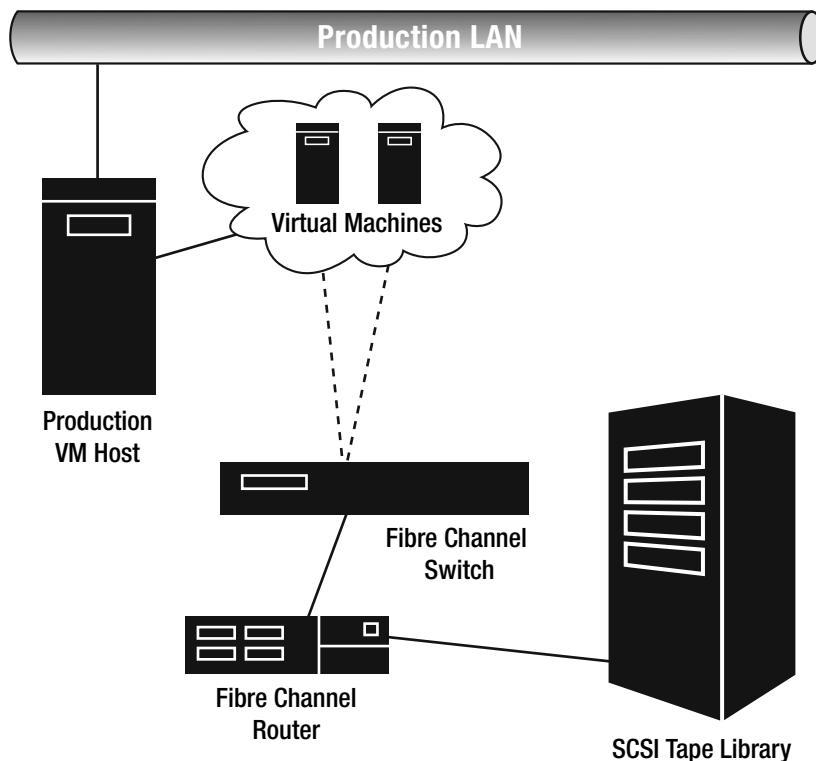


Figure 14-2. *VMware ESX Server SAN-attached deployment*

VMs can take further advantage of SANs by directly accessing physical disk resources on the SAN. The advantage to this is that you won't have physical disk I/O contention between multiple VMs residing on the same host. Directly mapping each VM to its own physical disk resources provides for separate I/O for each VM. If this isn't possible, and your planned deployment involves using virtual disks, then consider placing each VM on a separate physical drive on the host server.

Note When connecting VMs to a SAN, be sure to use zoning and LUN masking to ensure that multiple hosts can't attempt to write to the same physical disk resources simultaneously, thus preventing any chances for data corruption.

Consolidation Craze

Arguably the greatest benefits of consolidating to a VM infrastructure are hardware costs and maintenance savings. Some organizations have quantified this level of savings in hundreds of thousands of dollars. For example, one organization that we worked with was able to consolidate 30 physical servers onto individual VMs running on a single VMware ESX Server. This level of consolidation was possible because each consolidated server never fully utilized its available resources. This concept is true for many production servers, especially when most application vendors all but insist their products run on dedicated boxes. Being able to consolidate to virtual machines that collectively utilize all the hardware resources of their host allows you to maximize your hardware investments.

Many migrations to Microsoft's Virtual Server have produced similar results, with organizations achieving 16:1 consolidation ratios or higher.

Standby VMs

In Chapter 7, we talked at length about the many ways to back up and recover virtual machines. However, one production practice we didn't cover in Chapter 7 was using standby VMs. For years, standby, or *warm*, servers have been used as "a poor man's cluster." Basically, this approach involves maintaining another server that mirrors a production server. Usually data is synchronized between the production server and standby server through scheduled backups and restores.

With a standby VM host in place, you can maintain warm VMs in case a live production VM fails or in case the primary VM host fails. Later in the "Automating Standby Server Startup" section, we'll show how to monitor a VM host for failure and in turn how to start a standby VM if a failure occurs.

Storage Area Network

Although Figure 14-1 showed only two servers connected to a SAN for simplicity, many organizations that have SANs connect nearly all their servers to the SAN. With the typical organization having from several hundred gigabytes to terabytes of data to manage, configuring a dedicated network for data movement is practically a "no-brainer." This is especially true when you're faced with the challenge of backing up possibly a terabyte of data within an eight-hour span. For many shops, this task just isn't possible over a production LAN, so going to a dedicated SAN that provides up to 4Gb/sec of bandwidth is the only viable choice. Sure, you could directly SCSI attach a library to each server, but this solution is difficult to scale and expensive to boot.

To SAN or Not to SAN

If you're trying to consolidate and pool your storage resources, then SAN is likely a good bet for you. When you set up a SAN, you're building a separate network dedicated to your storage resources. This gives you the ability to both move and protect your data without having to impact your LAN. With backup devices and disk storage devices on the same storage network running at 1Gb/sec or higher, you'll have a much easier time completing your backup within your backup window when compared to a traditional LAN-based backup over a 100Mb/sec LAN.

Of course, you may find yourself thinking, "Well, I can run backups over 1Gb/sec Ethernet, so why go with a SAN?" This is a question we hear all the time. Alternatives exist to SAN for high-speed backups, but SAN hardware vendors such as Brocade have been busy the last

few years expanding the capabilities of the SAN. For example, you can perform block-level copies of data from one volume on a SAN to another, without consuming any CPU cycles on your network servers. SANs also give you the ability to dynamically allocate drives in a tape library to whatever server attached to the SAN needs them at the moment (current backup vendors that support this capability include CommVault, Veritas, and EMC Legato). In pre-SAN storage configurations, a library or tape drive would have to be SCSI attached to a single server, meaning any access to the drive would have to be through one server. With a SAN, all servers attached to the SAN have the potential to access any storage device on the SAN. Your hardware and software decisions when implementing the SAN will ultimately determine its capabilities.

Since with a SAN you have the ability to build out a network of storage devices, such as disk arrays and tape libraries, your storage will have almost infinite scalability. With the SAN in place, you'll find it much easier to add and allocate storage as needed, without having to throw additional servers at a storage problem. Server requirements will go back to being based on user load instead of being driven by the need for additional storage. With in-band virtualization appliances, such as those offered by Cisco Systems, disk resources can be collectively pooled and distributed as needed to a server. Although there will always be some guesswork involved with predicting future storage resources, with pooled resources in a SAN there are no consequences for being wrong. If project storage for one server is overallocated, some of the physical disks can be reallocated to another server attached to the SAN.

Although many can visibly see the benefits of SAN, most in small to midsize companies find themselves concerned with the cost. With cost, the good news is that unlike in the late 1990s and early 2000s, SAN is now a mature technology with countless competing hardware vendors offering viable solutions. For consumers, this means a SAN can be acquired for a relatively low price. For example, you can build a simple entry-level SAN consisting of a switch, router, and two HBAs for connecting two servers for less than \$10,000. A small business looking to start with used equipment can set up a SAN for less than \$3,000. Purchasing the router with the initial SAN will allow you to connect existing SCSI equipment into the SAN, meaning you won't have to be concerned with purchasing new Fibre Channel-equipped storage devices.

Fibre Channel or iSCSI

Fibre Channel's greatest advantage over iSCSI is that it was the first storage networking protocol to hit the street. In terms of industry acceptance, Fibre Channel technologies had been integrated into enterprise-class environments for nearly five years before the first iSCSI adaptations.

iSCSI, while still working to gain the market acceptance of Fibre Channel, can be configured at a substantially lower price. Setting up a software-based iSCSI concentrator, such as with the offering from Rocket Division Software (<http://www.rocketdivision.com>), requires no additional equipment. While prone to latency (12.5MB/sec) over a 100Mb LAN, iSCSI is a viable alternative for setting up a simple Ethernet-based SAN. Fibre Channel is a unique protocol with its own hardware requirements, meaning that you can't set up a Fibre Channel SAN without investing in new hardware. An inexpensive approach with iSCSI is to use a software concentrator and to connect devices on the iSCSI SAN over a 1Gb Ethernet switch. The 1Gb performance will offer up to 125MB/sec of throughput, which is acceptable for most storage applications. As Gigabit Ethernet technology improves, speeds up to 10Gb/sec (1250MB/sec) will be possible.

If you're most interested in the full-blown SAN features currently available, such as storage virtualization and serverless backups, then Fibre Channel is the clear choice.

SAN Hardware

When selecting SAN hardware, reliability and compatibility are key. If you plan to continue using your existing storage devices, your planned SAN should be compatible with your existing storage components (libraries, disk arrays, and so on). Also, you'll need to be prepared for the management and backup and recovery considerations mentioned later in this section. In particular, here are some elements to consider when planning your SAN deployment:

Do the planned SAN devices meet your long-term scalability goals?

Do the planned SAN switches support bridging multiple SANs over an IP network? This is accomplished through FCIP or iFCP support.

Do the planned SAN switches offer E_ports to facilitate easy expansion of the SAN fabric?

Are the planned SAN components certified by your backup software vendor? With enterprise-class backup software implementations ranging in cost from tens to hundreds of thousands of dollars, replacing your backup software may not be an option after the SAN has been implemented. If possible, build out your SAN using devices currently supported by your backup software vendors.

If you plan to build a fault-tolerant SAN using a redundant SAN fabric that incorporates multiple paths to SAN devices, you'll need multiple HBAs in each server. Each server will need to have a multipath driver installed for each HBA. Also, you'll need to ask the following questions to each of your storage applications: Does this application support multipathing? If so, which HBAs are supported? Lack of planning for multipath support can affect storage applications such as backup software and clustering.

Does your planned Fibre Channel router have the appropriate ports to support your existing SCSI storage devices?

Tip For more information on SANs, enroll in the free online SAN courses offered by EMC Legato at <http://www.sanacademy.com>.

Now that we've covered several important considerations for adding a SAN to your virtualized infrastructure, we'll cover DFS in the next section.

Distributed File System

Remember, the whole concept of virtualization deals with removing the physical dependencies associated with network resources. For most of us who have worked in IT long enough, we find ourselves blinded by our own habits. A user asks us where to find the company handbook, and we reply, “Just go to the \\Max\\Public\\HR\\EmployeeDocs folder! Duh!” Of course, to the typical user, this task isn’t so easy. To make life simpler for the users, we overcome the problems with UNC paths by giving them mapped drives. Now a user just needs to know which mapped drive letter is associated with which network resource. Try to change mapped drive letter assignments on your users, and you may face a revolt.

With DFS, administrators can configure a single starting point known as the *DFS root* for all network file system access, which could be associated with a single mapped drive letter for users. The DFS root will contain links to all other shared network resources, thus giving users and applications a single starting point to access all network resources. Furthermore, with a single logical access point for network resources, administrators are free to move data to different physical locations on the network without impacting users. All you’d need to modify is the link at the DFS root that points to the physical location on the network. With DFS, you could also set up replicas for each DFS link, allowing you to automatically provide for fault tolerance of link data by having one logical link point to replicated data that exists in two or more physical locations. Also, using redundant links offers the following advantages:

- Administrator’s can take down file servers for maintenance without impacting users.
- Backups can be run on a standby server that’s a replication target in the DFS hierarchy, giving you greater flexibility with your backup window.
- The network infrastructure is automatically resilient to system failure, without impacting the users.

Even if your organization has as few as two file servers, you can still start with a DFS root. In an Active Directory domain environment, you can configure domain controllers as domain-based DFS roots. This gives you the ability to map user drive letters (if desired) to a domain name instead of a server name (further removing physical dependencies and adding fault tolerance to the DFS root). This means a user’s DFS root will most likely be the domain controller closest to their system. Again, if your network uses mapped drives to assist users in locating network resources, then they will need only a single mapped drive to the DFS root. All other shared resources on the network will appear as subfolders under the DFS root folder. Since the subfolders are merely logical links with pointers to physical resources and locations, you can configure the subfolders under the DFS root in any means that best works for your organization.

Figure 14-3 shows a sample DFS deployment. In this example, NuView StorageX assists in the management of the DFS infrastructure.

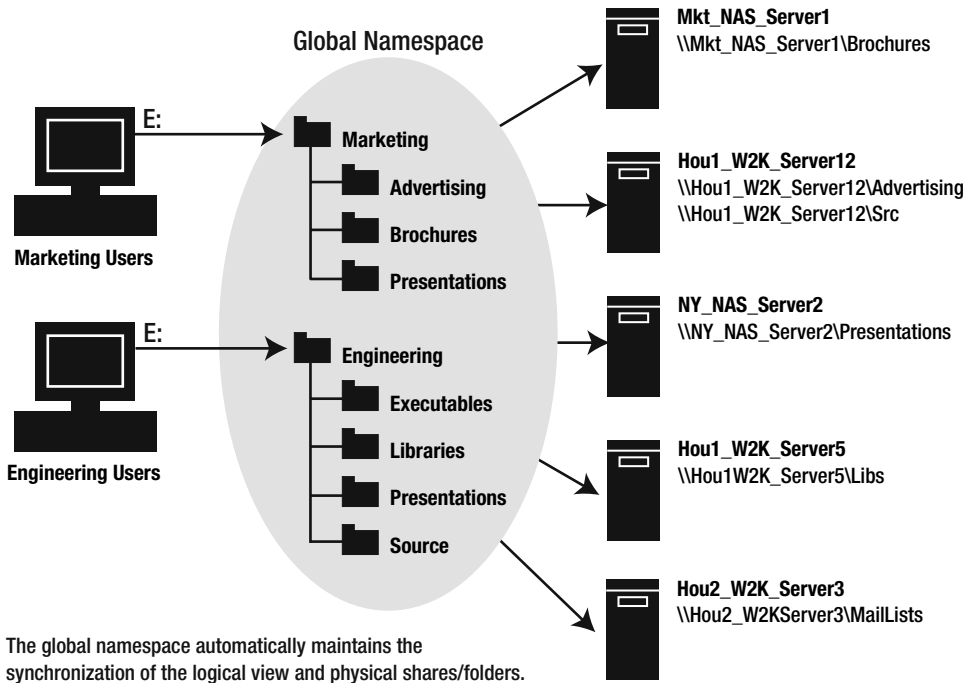


Figure 14-3. Enterprise DFS implementation using NuView's global namespace

Note We've included a full evaluation copy of StorageX on this book's companion CD. For more information on how StorageX adds value to the management of file system data, turn to the appendix.

Now that we've recapped the array of virtualization possibilities at your disposal, we'll wrap things up by covering the process for maintaining a warm standby VM server.

Maintaining a Standby VM Server

One of the greatest benefits of virtual machines is their hardware independence. Without having to associate a server with a specific physical system, your VM servers will have the freedom to be hosted by any capable system on your LAN. If you've ever tried fully recovering a Windows server by performing a full system restore to a system with different hardware than the originally backed-up system, you've probably experienced a great deal of pain.

A major dilemma with Windows server backup and recovery is the Microsoft concept of *System State*. In short, the System State represents all the core files needed to power an OS. To keep the core OS files synchronized for backup and recovery purposes, Microsoft allows System State contents to be backed up and restored only as a complete set. On paper, this is a great concept. No system files becoming out of sync should lead to more stable servers. However, if a server crashes and as a result you're faced with restoring a server's backup to a system with different hardware than the original, you'll probably be left with a server that's unable to boot. The problems you're faced with after the restore result from the backed-up system's configuration and device drivers differing from the new system's configuration. Of course, you can manually repair all the incompatibilities and eventually get the restored system to boot, but all of this takes time. If you're faced with having to restore a production server to a new system, odds are that time is something you don't have.

With VM applications emulating system hardware, there are no problems with running a VM on one physical host and then moving the VM to an alternate host. Since the hardware emulation is the same, the VM won't miss a beat on the new host system. In terms of supporting availability and recovery, the portability of VMs is a system administrator's dream. In addition to the fact that a VM backed up on one host can be restored to another host, you can also stage a standby VM on an alternate host. What's a standby VM, you ask? To answer that, consider the standby VM to be a mirror image of any production VM, simply waiting to start up in the event that the production VM fails.

Besides just providing for increased availability of production VMs, standby VMs can fill many other roles. For example, not too long ago we worked on a consulting project involving the staging of backup servers at a DR site. The client's dilemma was that they had five separately managed production sites and would need to stage five backup servers at their DR site. When asked of the probability of ever having to recover two sites simultaneously at the DR site, they responded with "next to zero." This opened the door for VMs, since only one backup server would ever need to be online at a time at the DR site. Configuring five backup servers as VMs and saving them on the same host system at the DR site proved to be the perfect answer. Each backup server VM was preloaded with the backup application and necessary services. In a DR scenario, all that would be required would be for the needed backup server VM to be brought online, then for its backup metadata to be restored (bringing the backup server up to date with all previous backup jobs), and finally you would need to connect the VM backup server to the LAN to launch any needed restore jobs to other systems.

Although this specific type of example may not apply to your circumstances, keep in mind that by thinking "VM first" when planning for any recovery or high-availability project you can often maximize the results of the project with a significantly smaller hardware investment.

Aside from a DR site scenario, you may find that maintaining a standby VM is perfect for a simple high-availability project on your LAN. In the following sections, we'll cover how to deploy and manage a warm standby replica VM file server and how to automate the startup of the standby server in the event that the primary server fails. To do this, we'll walk you through the process of setting up the VM infrastructure shown in Figure 14-4.

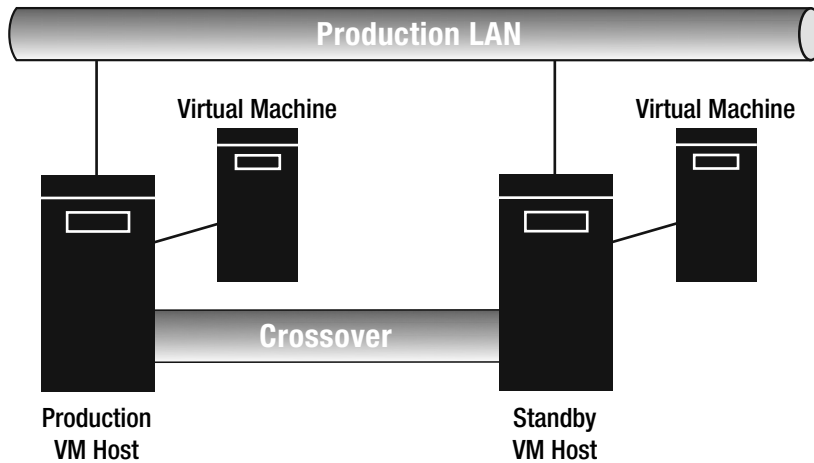


Figure 14-4. *Standby VM setup*

Note that the configuration shown in Figure 14-4 is a relatively simple network consisting of a single VM running on a production VM host. A backup version of the same VM will reside on the standby VM host. To synchronize new data between the two VMs, a crossover cable is used.

Setting Up the VM

The first step in the configuration process is to start powering down the production VM and copying all of its contents to the standby host server. Both VMs should be configured to use bridged networking and will employ identical network configurations in order to ensure transparency for client access. Of course, this also means that both the production VM and the standby VM can never be on at the same time.

While this setup works fine for Linux virtual machines, it presents a problem for Windows VMs. With Windows operating systems, the machine account password is automatically changed every seven days. So, if the machine account password that's used by the production VM becomes out of sync with the password on the standby VM, the standby VM won't be authenticated by a domain controller. One way to correct this is to rejoin the standby VM back to the domain following a failover. However, this is a manual and time-consuming process. Another approach is to disable computer account password changes on the production VM (prior to making the backup copy to the standby VM). To do this, you need to open the Registry Editor (run `regedit`) on the production VM and change the `HKLM\System\CurrentControlSet\Services\Netlogon\parameters\DisablePasswordChange` registry value to 1. Figure 14-5 shows this configuration change.

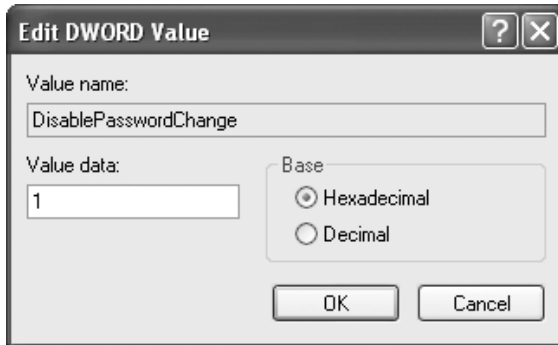


Figure 14-5. *Disabling Windows computer account password changes*

Having a standby VM in place is the easy part. Keeping its data synchronized with the production VM is another story. Of course, the easiest approach to synchronization is to perform periodic copies by powering down the production VM, copying its contents to the standby VM location, and then restarting the production VM. If the VM can be offline momentarily, then this may be a workable solution for you. For example, you could easily use a batch file or shell script to copy the VM's files to the alternate location on a nightly basis. You can find examples of scripts to copy VM files in Chapter 7.

However, if the VM must be online all the time, then you may want to consider scheduled backups and restores.

Maintaining a Standby Server with Scheduled Backups and Restores

With scheduled backups and restores, you could install a backup agent on the production VM and perform frequent backups of its live data. The frequency of the backups should be determined by your tolerance for data loss. Following each backup, you need to bring up the standby VM and restore the backup data to it. With identical IP addresses, this approach requires you to disconnect the standby VM's production LAN interface (or disable it at the least) during the restore. As you can imagine, this approach is filled with many complexities and is certainly not easy to automate.

Maintaining a Standby Server with Shared Storage

Before you say, "This standby server stuff isn't for me!" the solution presented in this section might be just what you need. If you have access to redundant shared storage (such as a RAID 5) on a Fibre Channel or iSCSI SAN, you can store the VM's virtual disk files there. Or you could even configure the production and standby VM to use the shared storage as a physical disk resource. With this approach, you're creating a failover cluster without needing the cluster service. Remember, if you're on a budget, Rocket Division Software (<http://www.rocketdivision.com>) offers an inexpensive software iSCSI solution.

Figure 14-6 shows the shared storage approach to standby VM maintenance.

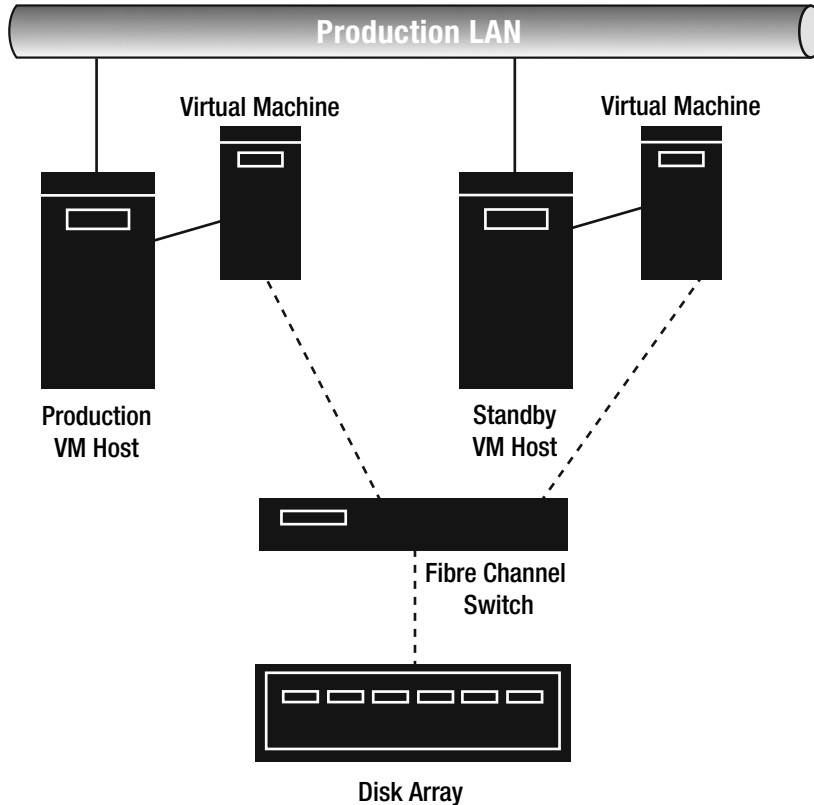


Figure 14-6. *Shared storage standby server configuration*

With this approach, either the VM's virtual disk could be configured to be stored on the shared disk array or the VM could directly access the shared storage array as a physical disk. In other words, your choices would be to have the VM either use a virtual disk file or link directly to the physical disk. With the virtual disks in a single shared location, all you'd need to reside on the standby host system would be a copy of the VM's configuration file. Also, you'd need to ensure that each system accessed the shared storage the same way (either via a drive letter or via a mount point).

With shared disk corruption being a concern, you can avoid this by mounting only the volume to the active host. For example, on Windows you can do this by using the `mountvol` utility. During the startup of a standby VM, you can execute `mountvol` with the `-p` parameter to dismount the shared volume on the production host or shut down the production host entirely using `shutdown.exe`.

If you're on an even smaller budget and you're looking for redundancy for just Windows VMs, Windows dynamic disks provide one additional option.

Maintaining a Standby Server with Disk Mirroring

With disk mirroring and shared storage such as with iSCSI, you can create real-time replication between a production and standby VM. While this procedure is a little complex, we've had success with it in our test lab. Note that we don't cover the specific steps for the iSCSI configuration here, but you can find them in Chapter 11.

Figure 14-7 shows the specific standby configuration covered in this section.

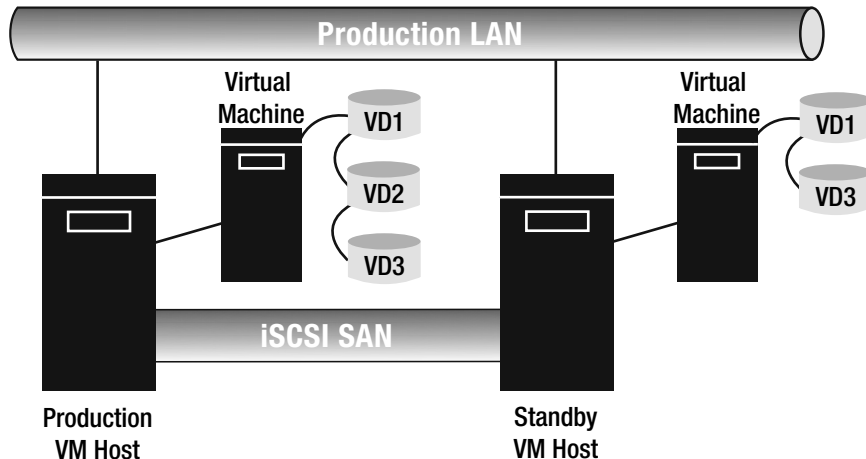


Figure 14-7. Standby VM synchronized via software RAID 1

The general guidelines for this configuration are as follows:

- The production server will contain three virtual disks: an OS virtual disk and two disks mirrored together as a mirror for critical data.
- One of the virtual disks in the mirror will physically reside on the standby VM. The other virtual disk will physically reside on the production server.
- Connection to the standby VM can be secured either by a shared network path or by an iSCSI mount to the standby VM's physical volume.
- The production VM will automatically synchronize its critical data with the standby VM since the second disk in the mirror is associated with the standby VM and is physically located on the standby VM server.

With an idea of where we're going with this, you'll now start configuring the production VM server. The key to this configuration is to have all critical data on its own disk, separate from the OS and application executable data. Since OS and application executable data isn't dynamic, there's no need to replicate it via the mirror. So, for a virtual file server, all user files would need to be stored on the data disk.

When you configure the production VM, its first two virtual disk files should reside locally. For the third virtual disk, include a path (network or iSCSI) to the standby VM. Ideally, you should have a 1Gb/sec network crossover between the production and standby server. This allows for 125MB/sec of disk throughput, which is sufficient in most applications.

With the disks configured, your next task is to install the guest OS into the VM. During installation, ensure that the OS is installed to the first virtual disk. Following the installation, it's now time to configure and partition a software RAID 1 on the two remaining virtual disks. To do this, you must first convert the disks to dynamic.

Caution Converting a basic disk to dynamic will cause all file systems mounted to the volume to be temporarily dismounted. Perform these steps only when users or applications don't have any open files on the disk.

You can do this by following these steps:

1. Click Start ► Administrative Tools ► Computer Management.
2. In the Computer Management MMC, click Disk Management (located inside the Storage parent snap-in).
3. As shown in Figure 14-8, right-click the existing data disk and select Convert to Dynamic Disk.

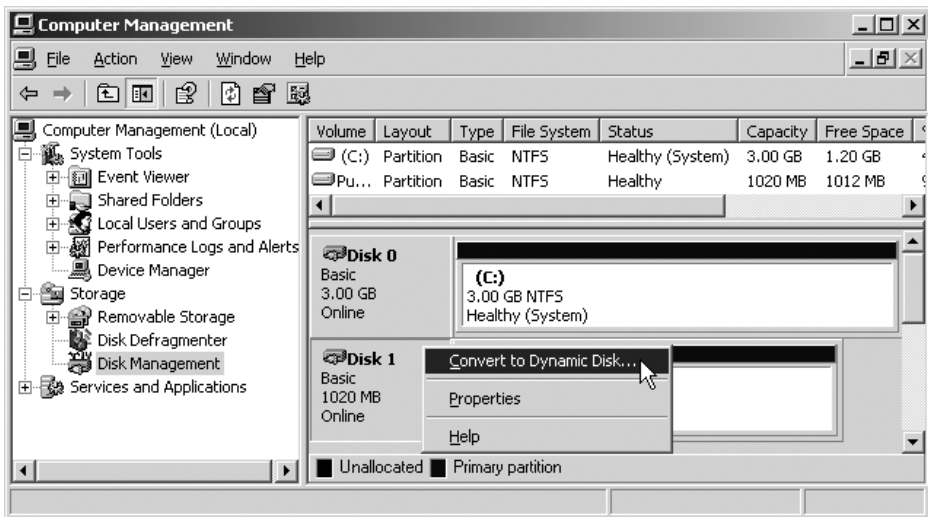


Figure 14-8. *Selecting to convert a disk to dynamic*

4. In the Convert to Dynamic Disk dialog box, you'll see all disks listed. Check the Disk 1 and Disk 2 boxes. (Disk 0 is used by the OS and doesn't need to be converted.) When finished, click OK.
5. In the Disks to Convert dialog box, click the Convert button.

6. You should now receive a prompt that you won't be able to start other operating systems on the volume once it's converted to a dynamic disk. Just click Yes to acknowledge the warning.
7. The file systems mounted to the volume will now be dismounted, thus making the volume temporarily unavailable. Once no users or applications are currently accessing the volume, click Yes.
8. You should now see the disk listed as Dynamic. Any partitions on the disk should now be listed as Simple.
9. At this point, you can close the Computer Management MMC.

Prior to creating a mirror, you first need to create a volume from the unpartitioned space on one of the dynamic disks. You can do this by right-clicking the unpartitioned space and selecting New Volume. You'll then be prompted to label and format the volume. Once you've done this, you can add a mirror to the volume. To configure the mirror set, follow these steps:

1. In the Computer Management MMC, click Disk Management.
2. Right-click the volume you want to mirror, and select Add Mirror (see Figure 14-9).

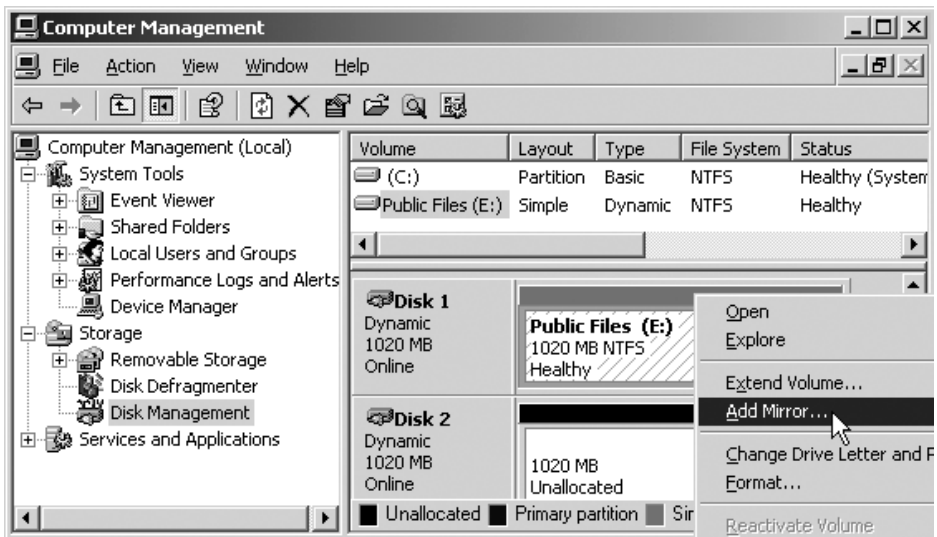


Figure 14-9. *Selecting to add a mirror*

3. In the Add Mirror dialog box, select the newly installed and configured dynamic disk and click Next.

The new mirror should now be created. Depending on the size of the original data disk, it may take several minutes for the mirrored disk to synchronize. Once the mirror is finished, you should now shut down the VM in order to create its replica. With the VM shut down, you'll

need to navigate to the VM's folder on its host system and copy the folder contents to the location of the standby VM. You shouldn't copy the primary data disk on the original system to the standby system's folder. For a Windows Server 2003 file server VMware VM, here's a sample of the files that need to be copied:

- IDE01.vmdk, IDE01-f001.vmdk, IDE01-f002.vmdk (boot/system disk)
- nvram
- vmware.log
- winNetEnterprise.vmx

Basically, all files in the source VM's folder will be copied, with the exception of its primary data storage disk. (This setup assumes that the OS boot and system partitions run on a separate disk.) In our configuration, the data disk's storage files are SCSI0-0.vmdk and SCSI0-0-f001.vmdk.

With the files in place for the standby VM, you'll need to make one additional modification. The VM will have a copy of the operating system's boot/system disk and host the mirror data volume (not the original). However, its configuration will still include information for both the source and the mirror volume disks. Since the source volume will be stored on the same host as the original VM, you'll need to remove it from the backup VM's configuration. To do this, open the backup VM and access its configuration settings. Then delete the virtual disk for the primary volume of the mirror set. You should actually see an error message as soon as you attempt to click the virtual disk for the primary mirror volume, since the virtual disk file won't be accessible to the standby VM. Figure 14-10 shows the configuration of the backup VM using VMware GSX Server.

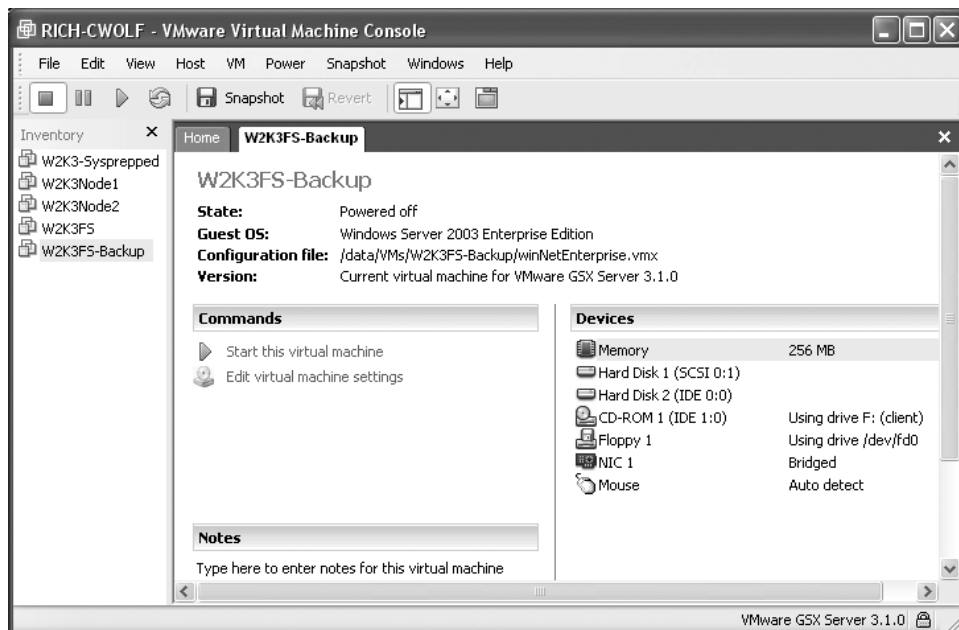


Figure 14-10. Standby VM configuration

Notice that the backup VM's configuration has only two virtual disks listed, whereas the original VM (in our example) has three. At this point, the standby VM is ready to go!

In this section, we cover a few ways to keep a standby VM synchronized with a production online VM. Of course, the standby VM is worthwhile only if it can automatically start up once the production VM fails. We'll show how to do that next.

Automating Standby Server Startup

One of the most significant aspects of standby server maintenance is the importance of automating the startup of the standby VM in the event that the production VM fails. This way, the replacement server can be fully online within minutes following a failure.

With this in mind, the key to automating the standby server startup is the ability to perform the following tasks:

- Monitor and detect failure of the primary VM
- Execute a script that starts the backup VM
- Notify an administrator when a failure occurs

Luckily, several third-party applications provide this required functionality. You can find many of these tools at either <http://www.download.com> or <http://www.sourceforge.net> by performing a search using the keywords *server monitor*. The tool we chose in this example was GFI Network Server Monitor, which is available at <http://www.gfi.com>. Network Server Monitor is capable of monitoring both Windows and Linux operating systems, performing simple ping checks for availability, and going much further by being able to query the status of system services such as DNS, FTP, HTTP, LDAP, SSH, Exchange, and Oracle. This means you can target monitoring on a VM to a specific service. If a failure has occurred, you can take action such as rebooting a system, running a script, restarting a service, and sending an alert e-mail. Since Network Server Monitor runs as a service, you don't need to be logged on for it to operate.

Configuring VM Monitoring

In our test scenario, we installed GFI Network Server Monitor 6.0 on our standby VM host. The first time the tool starts, you'll be prompted for the name or IP address of an SMTP server for e-mail alerts, as well as to set the criteria for a system to be monitored. If you elect to configure the monitoring features later, you can do so by following these steps:

1. Right-click the Monitoring Checks Configuration object in the GFI Network Server Monitor UI, select New, and then click Monitoring Check.
2. In the New Check dialog box, click the service to be monitored on the production VM. In our example, we chose to do a simple Internet Control Message Protocol (ICMP) ping test. When finished, click Next.

3. You'll next be prompted to set the monitoring criteria for the selected service. In our example, we used the default ICMP ping criteria, as shown in Figure 14-11. When finished, click Next.

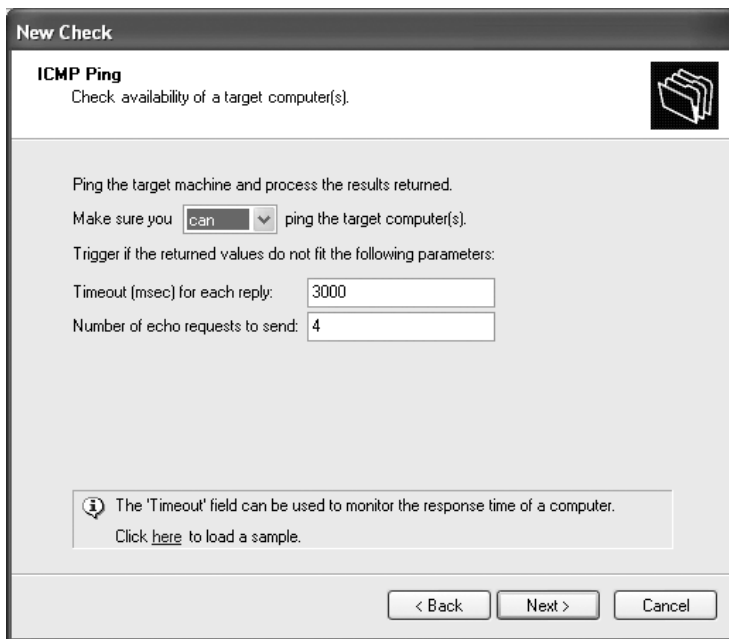


Figure 14-11. *Configuring monitoring criteria*

4. Enter a description for the check, such as **FS1 VM ICMP availability test**, and click Next.
5. Enter the name or IP address of the system to monitor, and click the Add button. Then click Next.
6. Click Finish.

With the monitoring check configured, you'll see it listed in the right pane of the Network Server Monitor UI. You now need to configure how the check responds to a failure. To do this, follow these steps:

1. Right-click the check, and select Properties.
2. In the Properties dialog box, click the Actions tab.
3. In the Run External File portion of the window, clear the Inherit Run External File Parameters from Parent Folder check box. Then click the Settings button.
4. In the Settings – Program Actions dialog box, check the When the Check Fails Execute the Following List of Files box. Then click the Add button.

5. Enter the full path to the standby VM startup script in the Settings – Program Actions dialog box (see Figure 14-12), and then click OK.

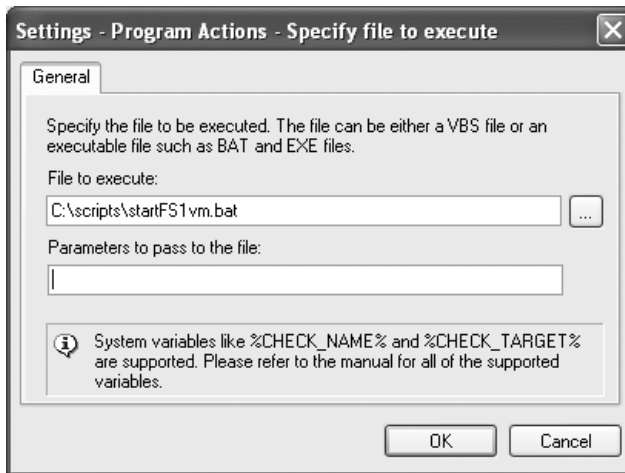


Figure 14-12. *Specifying VM startup script*

6. In the Settings – Program Actions dialog box, click OK.
7. Click OK in the Properties dialog box.

At this point, the alert and response are now configured. By default Network Server Monitor will send an e-mail message to the account specified during the program's initial setup if a failure occurs. This is why you don't need to configure the e-mail alert. If you want to manually configure the e-mail alert, you return to the Actions tab of the monitor's Properties dialog box and then clear the Notifications box. At this point, you'll be able to manually configure the e-mail notification settings by clicking the Settings button in the Notifications portion of the window.

Although we mentioned a startup script, we have yet to include one in this chapter. With that in mind, we'll cover two simple scripts that can automate the startup of the standby VM.

Scripting the VM Startup

In the following sections, we'll cover how to script the startup of a VM running on GSX Server and also how to script the startup of a Virtual Server 2005 VM. We'll start by covering how to automate GSX Server VM startup.

Note For additional examples of methods for starting up VMs from the command line, take a look at the scripts mentioned in Chapter 7.

Scripting the GSX Server VM Startup

Listing 14-1 is a simple script that automates the startup of a virtual machine residing on a Windows GSX Server. This is one of the simplest scripts used in this book, as it requires only a single variable. The lone variable is called `VMXPath`, which points to the script's target VM configuration file.

Listing 14-1. *Starting a GSX Server VM*

```
:: Filename: WinGSXVMStartup.bat
::
:: Purpose: Automates the startup of a GSX Server VM named WinFS1.
:: You would need to alter the VM name and paths for this script to run in your
:: environment.
::=====
@echo off
:: Set Variables
set VMXPath= E:\VMs\W2KAS\WinFS1.vmx

:: Start VM Using VMware-cmd
"D:\Program Files\VMware\VMware VmPerl Scripting API\VMware-cmd" %VMXPath% start
```

As you can see, there isn't much needed to start a VMware VM from the command line. In fact, if you didn't want to create a script, you could simply launch the `VMware-cmd` file with the appropriate parameters as a way to start a VM. Virtual Server VM startup is a little more complex, as it can only be accomplished with a Visual Basic script. We'll cover that next.

Scripting the Virtual Server 2005 VM Startup

The VB script in this section will allow you to start a single VM running on a Microsoft Virtual Server 2005 host system. Here are the variables required by the script:

- **strVMName:** Provides name assigned to virtual machine (visible in Virtual Server Web UI)
- **objVirtualServer:** Provides connection to Virtual Server application
- **objVirtualMachine:** Provides connection to specific virtual machine

With the variables out of the way, Listing 14-2 shows the script.

Listing 14-2. *Starting a Virtual Server VM*

```
' Filename: VS2005VMStartup.vbs
'
' Purpose: Automates the startup of a single virtual machine on a host system.
' =====
Option Explicit
'Declare variables
Dim strVMName
Dim objVirtualServer, objVirtualMachine

'Define Script Variables
strVMName = "WinFS1"

'Instantiate Virtual Server Object
Set objVirtualServer = CreateObject("VirtualServer.Application")

'Instantiate Virtual Machine Object
Set objVirtualMachine = objVirtualServer.FindVirtualMachine(strVMName)

'Start VM
objVirtualMachine.Startup()
```

When this script executes, the virtual machine WinFS1 will automatically start. To configure this script to run in your environment, the only modification required would be to change the value assigned to the strVMName variable.

Testing Standby Server Startup

Once you've saved and configured the scripts, you can now test the performance of the monitoring tool and startup script. The easiest way to do this is to isolate the standby VM from the production VM. Rather than interfere with the running state of the production VM, you can test failover by disconnecting the standby VM host's associated switch from the rest of the network. This causes the host and monitoring application to no longer be able to reach the production VM host, which in turn starts the standby VM.

Once you're satisfied with the test, power down the standby VM and connect the standby VM host's associated switch back into the rest of the network. Voila! You now have a fully automated high-availability virtual machine infrastructure!

Summary

In this chapter we recapped the vast array of virtualization technologies currently in many information systems. We also covered how VM technology can greatly assist in maintaining the high availability of crucial servers. Ultimately, the only foreseeable limits to virtualization are those of your own imagination. In the appendix, you'll get a glimpse of virtualization's future by looking at the technical contributions of several leading virtualization vendors.



Virtualization Product Roundup

Virtualization products have grown predominantly because of the forward thinking of many IT vendors. In this appendix, we'll cover a sample of ideologies from the various IT vendors that are developing tomorrow's virtualization technologies.

Global Namespace: The New Paradigm in Distributed Data Management

IT administrators spend a great deal of time on file management tasks (such as adding users, adding file servers, rebalancing storage, setting up failover, and so on) and data movement tasks (such as replication, migration, consolidation, and data distribution). These tasks are tedious and time-consuming for administrators, disruptive to users, and expensive for companies.

Therefore, companies are looking for better ways to scale and manage their file systems. A global namespace provides the answer.

Note A *global namespace* is a logical layer that sits between clients and file systems for the purposes of aggregating multiple, heterogeneous file systems and presenting file information to users and applications in a single, logical view. The benefits of a global namespace are clear and compelling: users (and applications) are shielded from physical storage complexities; administrators can add, move, rebalance, and reconfigure physical storage without affecting how users view and access it; and a global namespace provides a platform for developing value-added functions such as data migration, server consolidation, and business continuity applications.

With a global namespace in place, the administrator can perform data management and data movement tasks in less time, without disrupting user access to files. When files are moved, links in the namespace are automatically updated, which reduces manual administration and ensures continuous client access to data.

The following white paper discusses how a global namespace simplifies file management, how to create and deploy a namespace, the solutions it enables in an enterprise environment, and how StorageX from NuView builds upon the global namespace foundation to deliver a comprehensive network data management platform.

Note The following material comes directly from “Global Namespace: The New Paradigm in Distributed Data Management,” written by NuView (2004).

The Problem: Network Data Management and Movement

Stephens Company has 600 marketing and engineering users who are accessing files across five file servers that are shared by two departments and located in Houston and New York City. Marketing users are accessing files via multiple drive letters that are mapped to two filers and a Windows server, and engineering users are accessing files on three servers and one filer (see Figure A-1).

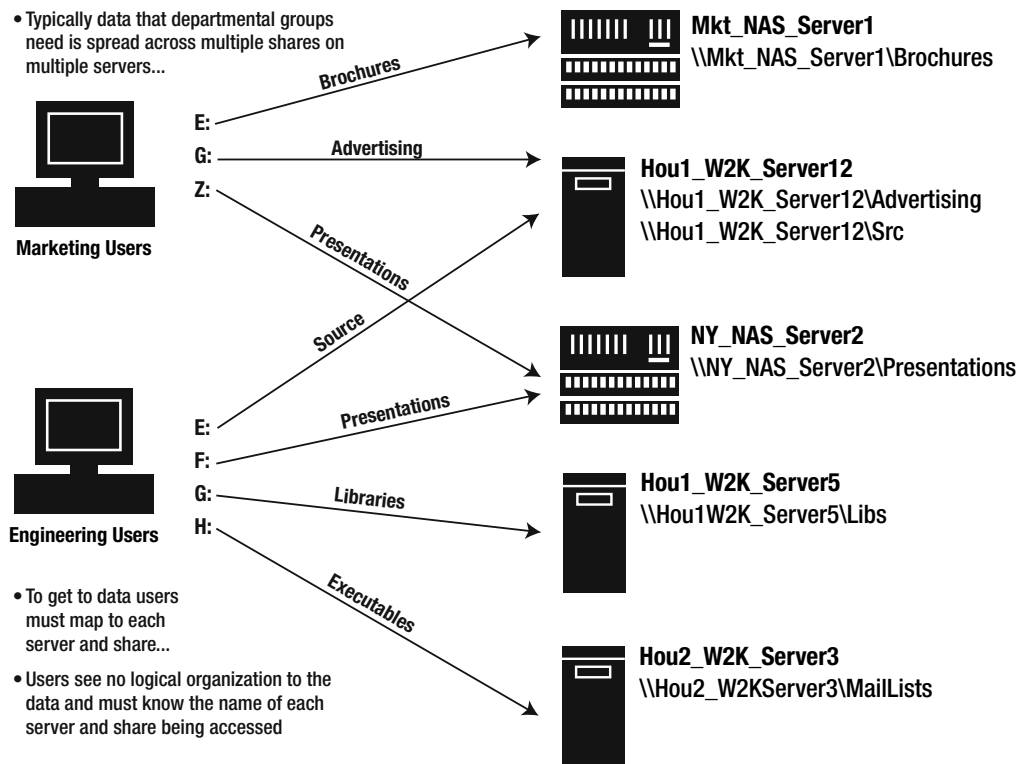


Figure A-1. Current file system environment

There are several issues with Stephens Company's current file system environment:

- Users find it difficult to locate and access files via multiple drive letters, which are increasing.
- File server Hou1_W2K_Server12 is at 90 percent capacity, while NY_NAS_Server2 is at 20 percent—this means users are beginning to receive errors as they try to save large graphic files to Server12.

- To migrate files and rebalance storage between Hou1_W2K_Server12 and NY_NAS_Server2, the administrator must disable user access to the files that are to be moved, move the files to the NY filer, reboot and bring both devices back online, revise all marketing and engineering user login scripts, and inform users that the files have a new location so their PCs can be reconfigured to access them—this will require at least 12 hours of downtime and manual reconfiguration of every desktop and application that accesses the migrated files.

The Solution: Global Namespace

There is a simple, long-term solution to Stephens Company's data management and data movement issues. Deploying a global namespace will simplify data management and enable transparent data movement.

Figure A-2 shows the new configuration, in which a global namespace has been inserted between users and physical storage. Users now access their files through shares called `\\namespace\users\marketing` and `\\namespace\users\engineering`. This was a nondisruptive installation, as the namespace was installed on top of the existing infrastructure. Users continue to access files in the same manner as before, with no retraining needed.

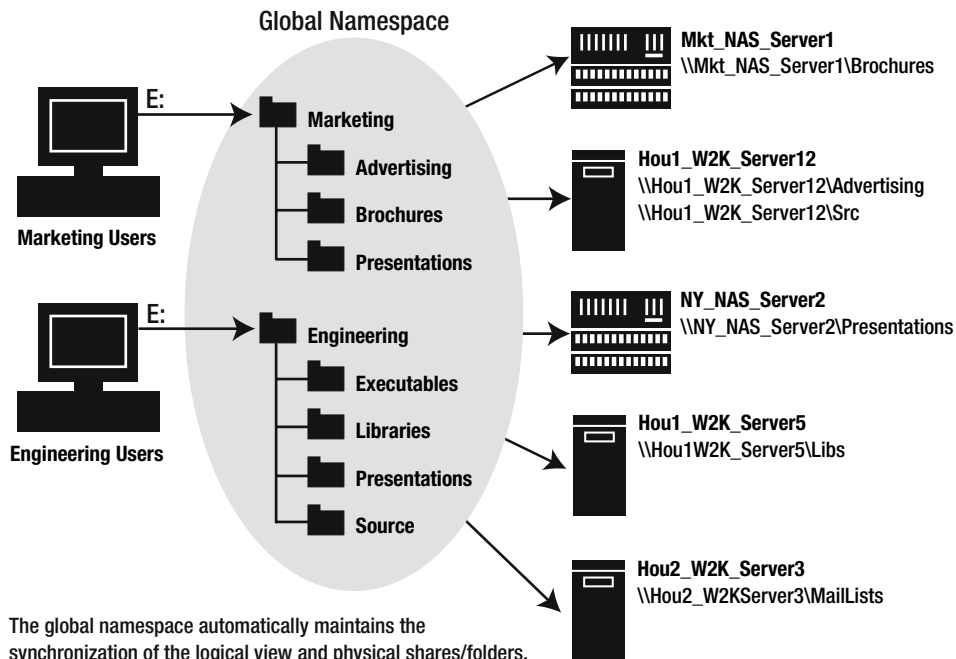


Figure A-2. File system environment with a global namespace

Note how the file system environment is changed by the introduction of a global namespace:

- All users see a single, logical view of files through the namespace.
- Users access all their files through a single drive letter—this will not grow and allows them to continue accessing files in a familiar way.
- Data can be organized and presented to users in a way that makes sense to them, regardless of how or where the data is stored.
- Data management and data movement are performed “behind the veil” of the namespace.
- Data changes are automatically updated in the namespace and require no client reconfiguration.
- Administrators can expand, move, rebalance and reconfigure storage without affecting how users view and access it.
- Data management and data movement require far less physical administration and are performed in less time than before.

Having a global namespace in place makes it easy for IT managers to accommodate the changing needs of an organization while also reducing storage costs.

How It Works

The global namespace does for files what DNS does for networking—it provides a directory service. A global namespace resides on a namespace server, which can be any Windows 2000 or Windows 2003 server within the namespace configuration. Clients (users and applications) view and access files through the namespace.

When users select a file from the namespace, a request is sent from the client machine to the namespace server. The namespace server receives the request, provides file location information back to the client machine, and redirects the client to the file in its physical location. The entire process occurs instantaneously and is transparent to clients (see Figure A-3).

This architecture provides several significant advantages over other file system aggregation approaches:

- There is nothing in the path between clients and the data, which is important for two reasons: performance is not impeded, and there is no single point of failure.
- The namespace server may be clustered to provide high availability of the namespace.
- There is no client installation, which simplifies namespace deployment and management.
- It takes only minutes to install the namespace software and less than an hour to create and populate the namespace—an administrator can deploy a namespace and begin receiving namespace benefits in a single day.
- Physical desktop and server management is significantly reduced, as data movement and management are simplified and no client reconfiguration is required—ever.

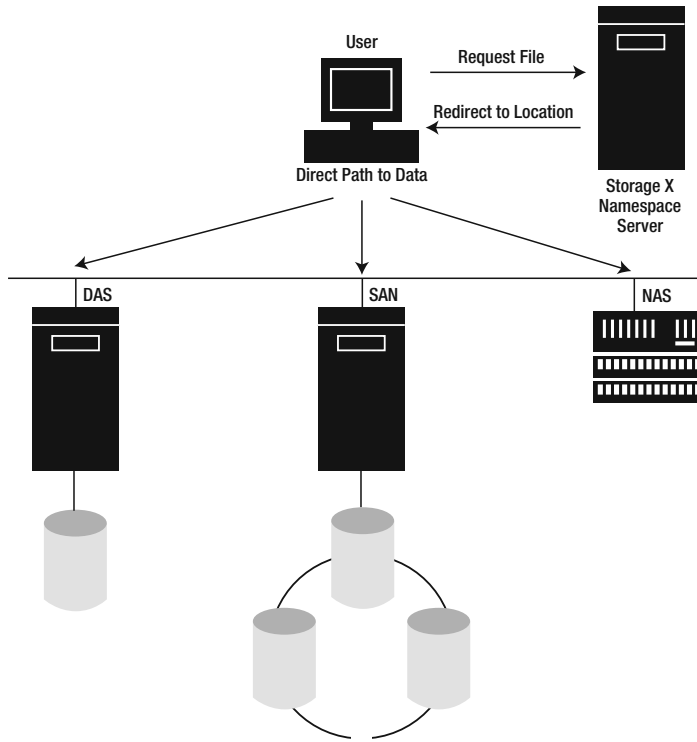


Figure A-3. *Global namespace architecture*

Creating and Deploying a Namespace

There are several key requirements to the effective use of a namespace. The global namespace solution available today that meets these requirements is NuView StorageX. Table A-1 lists global namespace requirements.

Table A-1. *Global Namespace Requirements*

Requirement	How StorageX and VFM Meet the Requirements
Tools for namespace creation and population	Discover shares and automatically construct a namespace that the administrator can modify as needed.
Ability to synchronize the namespace with the underlying physical storage	The namespace is automatically updated when storage is added, moved, changed, or reconfigured.
Namespace monitoring and management	Provide distributed file system and namespace monitoring and management from a single console.
High availability of the namespace	The namespace can be clustered for high availability and to eliminate any single points of failure.
Easy to change or reconfigure the namespace when needed	Both products provide simple processes for nondisruptive namespace reconfiguration.

Using Global Namespaces in the Enterprise

The global namespace is a powerful new paradigm in storage management architecture that answers a host of challenges faced by IT administrators today. Some of the scenarios in which a global namespace should be deployed are discussed below.

Managing User Home Directories

In Windows environments, users are typically given access to their home directory through a user profile and/or login script. During login, the user's computer is physically mapped to the storage device/devices that contain their data. This method requires a great deal of administrator intervention when data and directories are moved from one device to another. First the administrator must notify all users that the data is being migrated; then all of the users must log off in order for the migration to be successful. Once the data is migrated, the administrator has to deal with the error-prone nightmare of modifying the login scripts of all users. This process typically takes a few hours. However, in many cases users are unable to connect to their relocated directories without requiring administrator support. This process is problematic in an environment with 1,000 users and 10 file servers but becomes exponentially more complex as the number of users and devices increases.

A global namespace significantly reduces the time and effort required to manage user home directories. Using a global namespace, administrators can move home directories across devices without having to modify login scripts or user profiles. Users also benefit because they are not required to log out and log back in when their data is migrated. With a global namespace in place, a single administrator is able to easily manage thousands of users, links, and devices, which provides for a substantial decrease in the cost of managing storage.

Sharing Data Across Multiple Departments

Most corporate network users are mapped to a drive letter that connects them to the data for their department, but they also need to access data “owned” by other groups. For example, a marketing manager needs access to the marketing data on the network (brochures, presentations, collateral), as well as engineering road maps and project information. In order to connect to the engineering file shares, the user is required to physically map a drive letter to the specific server name where the engineering file share is located. In this scenario, it is cumbersome to share data across multiple groups, as the process of physically mapping drive letters to servers is not seamless for the end user. Additionally, users are required to know distinct, often cryptic, server names in order to access data, and they may eventually run out of drive letters.

A global namespace makes it simple and seamless for users to share data across departments. The global namespace presents users with a logical view of all of the shares they have access to. Users are no longer required to know where data is physically stored, nor are they required to know and map to specific server names. In the above example, the marketing manager connects to a single drive letter that provides a logical view of both the marketing information and the engineering information through a single namespace.

Optimizing Exchange Usage

Since file sharing without a global namespace is cumbersome and inefficient, many corporate users attach documents and presentations to e-mails for distribution to other users. This practice has led to a data explosion, as most e-mail recipients tend to save the files, either locally or on a network drive. If 24 users are e-mailed a document, then there are 26 copies of the same document being stored somewhere on the corporate network. This type of file sharing eats up valuable network bandwidth, especially when the file being e-mailed is several megabytes in size.

A global namespace provides an efficient means of sharing files among users. Instead of e-mailing files as an attachment, users are able to send a link that points to the location of the file. This not only saves network bandwidth but it also provides version control during the document review process. Through the global namespace, multiple users can make revisions and track their changes in a single document.

Lifecycle Management of Reference Data

Reference data is information that is fixed but needs to be kept available for regulatory, informational, or business purposes. Some examples of reference data include historical financial statements and patients' medical imaging files that are generated by an X-ray or MRI system. This type of data produces very large file sizes. For example, a single MRI scan generates 20,000 individual files. These files are required to be online and available at all times, which can be costly to maintain on primary, high-performance storage systems.

Implementing a tiered storage architecture is a highly efficient approach to meeting reference data storage needs. A global namespace enables the implementation of a tiered storage architecture by transparently facilitating the movement of reference data off of expensive primary storage onto less-expensive secondary storage based on administrator-defined criteria. In the above example of medical imaging files, the hospital IT administrator could use the global namespace to automatically and transparently migrate medical imaging files to secondary storage 30 days after the patient's visit. Moving data to secondary storage not only saves hardware acquisition costs but also enables administrators to match backup policies and investment with the business value of data.

Managing Web Site Data

A typical Web site consists of thousands of files and hundreds of directories. To add to this complexity, each Web page often contains dozens of URLs that link to specific directories on specific servers. Any changes in the physical devices hosting a Web site can affect numerous pages/files and can result in a malfunctioning Web site.

Using a global namespace to manage Web site data can yield significant benefits to both developers and administrators. Web site files that are stored on multiple devices can be pooled together into a single logical namespace and managed as a single entity. Administrators can move files around "behind the veil" of the namespace without breaking links, and multiple Web sites can be hosted transparently on one server. With a global namespace, the transition from test to production servers is quick and seamless. With the click of a mouse button, administrators can push out multiple Web pages simultaneously.

StorageX Uses Global Namespace to Deliver a Complete Network Data Management Platform

StorageX is the industry's leading network data management platform. It is an integrated set of applications that logically aggregates distributed file storage utilizing the StorageX Global Namespace as its foundation. StorageX provides administrators with policies that automate data and storage services, such as heterogeneous network data management, data migration and consolidation, business continuity, storage optimization, data lifecycle management, remote site data management, and data classification and reporting. More important, the StorageX Global Namespace provides administrators with the ability to perform these tasks without causing downtime to users. Users also benefit from simplified access to their data, as the StorageX Global Namespace delivers a unified, logical view of data distributed across heterogeneous storage platforms (see Figure A-4).

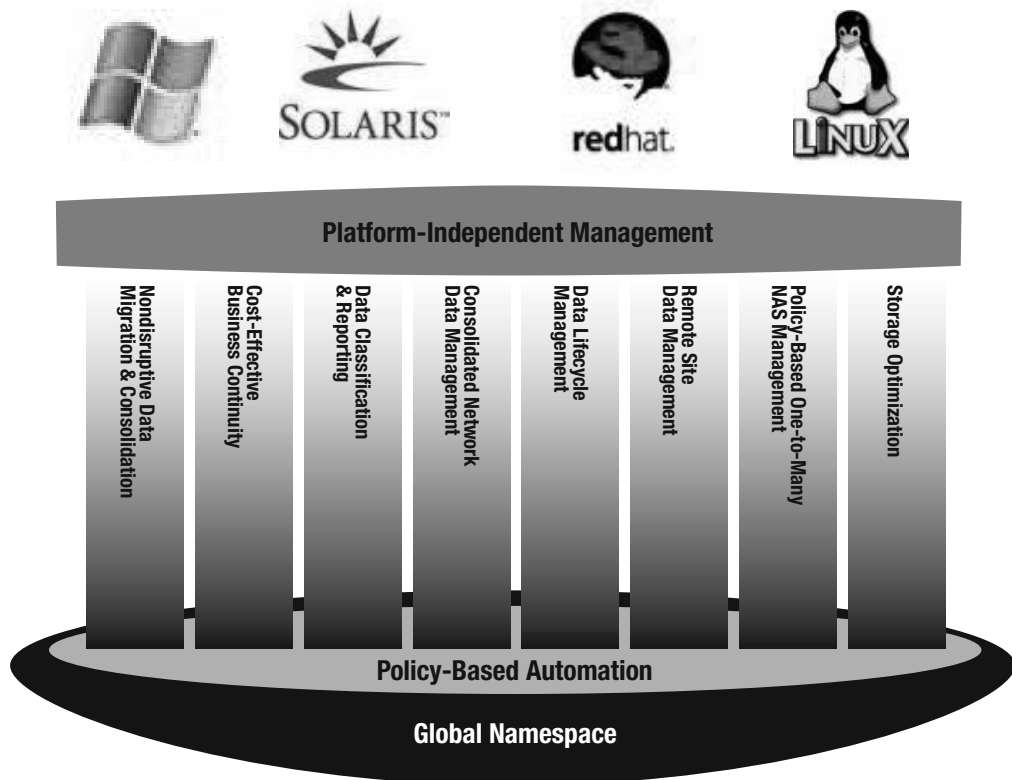


Figure A-4. *StorageX Global Namespace*

What Is Unique About the StorageX Global Namespace?

There are a number of products on the market today that tout global namespace capability. However, there are key elements to the StorageX Global Namespace implementation that deliver a compelling, unique value proposition to enterprise customers.

Data Directory Services

Similar to the way LDAP delivers user authentication services and DNS delivers network-related information services, the StorageX Global Namespace provides directory services for networked storage. The StorageX Global Namespace serves as a central repository for storage related information and functions and provides administrators with a single location to manage network data. One way to conceptualize the data directory services provided by a global namespace is to think about the Yellow Pages. Like the Yellow Pages, a global namespace is a directory. In the same way that a listing can appear in multiple sections of the Yellow Pages, a global namespace can provide multiple views of the same data set. The global namespace concept is often used within the context of clustered file systems; however, a global namespace delivered via a clustered file system is like the White Pages. Only one listing can appear for a particular entry in the White Pages and in clustered file systems, only one view can be provided for a given data set.

Complete Global Namespace Manageability

StorageX provides administrators with a comprehensive tool for global namespace creation and management. StorageX enables administrators to create and manage multiple global namespaces through a single, intuitive console and provides a means to manage both the global namespace and the underlying file system from the same console. StorageX provides administrators with the ability to dynamically populate the global namespace based on existing shares/export naming conventions or based on security in the enterprise. It also provides administrators the ability to monitor, scale, increase availability, audit, back up, restore, and snapshot the global namespace. This set of features delivers complete manageability of the StorageX Global Namespace and enables administrators to scale their deployments from a simple global namespace to a complex group of global namespaces.

Not a Proprietary File System

The StorageX Global Namespace is a layer that sits on top of the existing file system and does not require any modifications to the existing network infrastructure. Therefore, customers can continue to benefit from the advantages inherent in their existing file systems (WAFL, NTFS, VxFS) such as performance, journalizing, point-in-time recovery, encryption, compression, and security. StorageX does not require any changes to network operations, such as snapshot and backup processes. It utilizes time-tested CIFS and NFS network protocols that are offered by the existing file system vendors (Microsoft, Network Appliance, and EMC), which is a major advantage over solutions that require the introduction of a new protocol. Utilizing the underlying file system enables the StorageX Global Namespace to deliver significant benefits over aggregation solutions using a proprietary file system.

Standards-Based Platform

StorageX is an open, standards-based platform that can be seamlessly and nondisruptively introduced into an IT infrastructure. It is built on open, existing standards (DFS, Automounter, NFS, and CIFS); it runs on industry standard hardware and is compatible with Windows (95, 98, ME, NT, 2000, 2003), Linux, and Solaris. StorageX is an out-of-band solution and does not introduce any performance or latency issues when used to create and manage the global namespace.

No software or agents are required to be installed on the clients accessing the global namespace, and unlike hardware-based solutions, StorageX does not require the introduction of a new protocol on the network.

Platform for Unlimited Scalability

There is no limit to the scalability of a global namespace implemented with StorageX. Administrators can use the StorageX Global Namespace to aggregate multiple file systems and manage them as a single entity. This enables administrators to overcome the scalability limitations of individual file systems and manage tens of thousands of directories and trees through a single global namespace.

Simple to Install and Use

Deploying a StorageX Global Namespace is a simple process. It takes only minutes to install the software and less than an hour to create and populate the global namespace. Using StorageX, an administrator can deploy a global namespace and begin receiving its benefits in a single day. Administrators can even use permissions established via Active Directory to automatically create and manage global namespace, thereby applying the established network security framework to the new global namespace.

Conclusion

The global namespace technology delivers immediate, significant benefits to companies that are dealing with data management, data movement, and data availability challenges.

By creating a universal namespace, users are never disrupted, and administrators can change, add, grow, or shrink without ever affecting the environment. This is the way everyone will manage their NAS devices in the future.

—Steve Duplessie, Enterprise Storage Group

Global Namespace offers advanced capabilities to customers and provides an answer to problems in a networked storage environment, namely, the ability to scale NAS, file data availability, and manageability of distributed file storage.

—Randy Kerns, senior partner for Evaluator Group, Inc.

Solving the multiple namespace problem will prove to be the next big thing in information technology.

—Dan Tanner, senior analyst for Storage and Storage Management

Note Again, the previous material comes directly from “Global Namespace: The New Paradigm in Distributed Data Management,” written by NuView (2004).

Server Consolidation and Beyond: Enhancing Virtual Machine Infrastructure Through Automation

Today's data center consists of an increasingly complex mixture of server platforms, hardware, operating systems, and applications. Data centers have accumulated and assimilated a large variety of new technologies that over time have become "legacy" technologies that never go away. Windows servers and Linux servers exist in large numbers and collectively increase data center costs because of power consumption, temperature conditioning, and floor space. As complexity increases, so does the number of people required to manage the data center. As a result, the probability of human error and the risk to the underlying business services that the data center supports increases dramatically.

The advent of server virtualization technology is allowing data centers to reverse the complexity in the data center to a degree by allowing many physical servers to be hosted and isolated from each other on fewer physical machines. Virtualization abstracts the operating system layer and the application layer away from the hardware layer, thereby allowing for greater OS and application flexibility and portability between server hardware. By setting up virtual partitions, existing applications and users "see" each VM as an independent physical server although they share common CPU, disk, memory, and network resources (see Figure A-5).

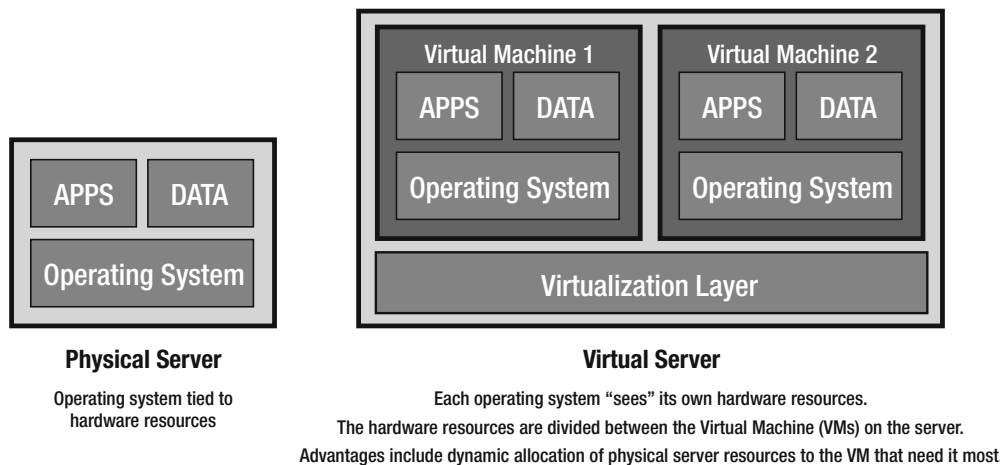


Figure A-5. *Virtual server architecture*

Like many other new technologies, data centers will employ the use of virtual machines, where suitable, alongside their physical counterparts. While server virtualization alleviates the complexity of the data center to some extent through consolidation, it also increases heterogeneity as well by introducing yet another infrastructure platform into the data center's technology portfolio.

Note The following material comes directly from the white paper "Server Consolidation and Beyond: Enhancing Virtual Machine Infrastructure Through Automation," written by PlateSpin (2004).

Evolution of Virtualization in the Data Center

As with all data center technologies, virtualization will be an evolution, not a revolution in server infrastructure. Many virtualization projects will begin with low-risk deployments such as replicating test lab environments and supporting high-availability requirements to reduce the need to buy large amounts of hardware. As virtual technologies mature and comfort levels with the technology rise, virtual machines will be selectively deployed into environments that have low-to-medium server resource requirements. As more and more virtual machines are used in production, and resource utilization rates increase due to consolidation and virtualization, there will be a need to balance load between virtual machine host servers as business service levels change. The abstraction of operating systems and applications away from the hardware layer will allow virtual machines to become very portable, given the right level of automation and conversion technology. Virtualization technologies will enable enterprise data centers to balance load, not by changing applications or adopting clustering technologies but by moving virtual machines between hardware resources.

Enhancing Data Center Flexibility Through PlateSpin PowerP2V™

In order to increase flexibility in a multiarchitecture environment, data centers must eliminate the main inhibitor of server portability: human manual effort. To convert a physical server to a virtual machine, the user must undertake a series of manual steps to provision a virtual machine from scratch. This includes installing operating systems, service packs, and applications, as well as configuring network, memory, and drives.

PlateSpin PowerP2V changes this limitation by offering a 100 percent automated way of moving servers between physical and virtual machine environments. With PowerP2V, data centers can easily discover physical servers and virtual machine hosts and perform Windows and Linux physical-to-virtual (P2V) and virtual-to-virtual (V2V) conversions. PowerP2V saves many man-hours of work over manual methods and conversion wizard utilities. It reduces the probability of downtime due to human error by eliminating the need to manually intervene during various points of the conversion process and requires little knowledge of virtual infrastructure.

Here is a summary of PlateSpin PowerP2V's features:

- Convert physical servers to VMware virtual machines (P2V)
- Perform virtual-to-virtual (V2V) conversions between virtual hosts
- Perform virtual-to-virtual conversions within a virtual host
- Convert Windows- or Linux-based servers
- No agents to install, simply connect to the network and convert remotely
- Autodiscovery and inventory of physical server and virtual machine hardware resources, operating systems, installed patches, and applications
- Simple to use: simply drag and drop a source server to a virtual machine host
- Reconfigure target VMs on the fly as part of the conversion process
- Perform multiple conversions simultaneously
- Perform conversions where the source and target servers are geographically distributed

Comparing Other Methods of Converting Between Physical and Virtual Machine Infrastructure with PlateSpin PowerP2V

PlateSpin PowerP2V is the only server conversion solution on the market that allows for *full* automation of P2V or V2V conversion processes for Windows and Linux servers. It allows data centers to easily virtualize their physical servers in an extremely rapid time frame compared to manual or semimanual methods of conversion.

Although assistant- or wizard-based P2V tools are excellent for small-scale virtualization projects (e.g., three to five conversions), as the number of server conversions increases they can become quite manually intensive and time-consuming for users. To illustrate, here is a summary (see Table A-2) of the steps that must be taken when converting a single physical server to a virtual machine as compared to PowerP2V.

Table A-2. *Physical-to-Virtual Migration Comparison*

PlateSpin PowerP2V	Semimanual P2V Tools for VMware
1. Drag and drop source machine to target VM host (no agents to install, no boot CD required—conversions can be done without physically visiting the source or target machines).	1. If converting Windows NT, install P2V tool on Windows NT; otherwise, install on Windows 2003 or XP.
2. Configure CPU, disk, memory shares, and network parameters.	2. Go to source machine, insert boot CD and reboot.
3. Press Start.	3. Select network interface, and set up networking.
	4. Scan for additional network devices if necessary.
	5. Define network settings (IP, network, gateway, TCP port).
	6. Record source server IP address and port number.
	7. Go back to P2V tool helper machine, and start wizard.
	8. Enter source server IP and port number from step 6.
	9. Select source disk.
	10. Specify target disk.
	11. Create new virtual disk—specify path and size.
	12. Locate current service pack or hotfix files (if Windows NT).
	13. Set partition sizes.
	14. Locate service pack (if Windows NT SMP conversion).
	15. Begin disk cloning.
	16. Restart tool.
	17. Perform reconfiguration.
	18. Browse for virtual disk to reconfigure.
	19. Create new virtual machine.
	20. Import virtual disk file.
	21. Power on target VM.
	22. Attach installation media (Windows NT only).
	23. Remove stale devices (NT only).
	24. Install VMware Tools.
	25. Reconfigure services and devices as necessary.

Manual methods typically take one to two days of effort to provision a virtual machine from scratch. Semimanual methods, such as assistant- or wizard-based tools usually can improve that slightly to a few hours of user time. With PowerP2V, it is essentially a three-step process that usually requires only five to ten minutes of user time per conversion. The following screenshots (shown in Figure A-6 to A-8) illustrate what the user would need to do to perform a conversion using PowerP2V.

Step 1: Select Source Machine

PlateSpin PowerP2V automatically performs a network discovery, identifying all physical machines that are attached to the network. The user can select a server or a virtual machine host and remotely retrieve all hardware information such as CPU, disk, memory, network cards, and all software information such as operating system, patch level, and applications installed. To convert a physical or virtual source machine to a virtual machine, simply drag and drop any server to any virtual machine host. All of this is accomplished remotely by the user, without having to physically visit the source or target machine. The source and target machine can also be located in different geographical regions, and conversions may occur over large distances across a private network.

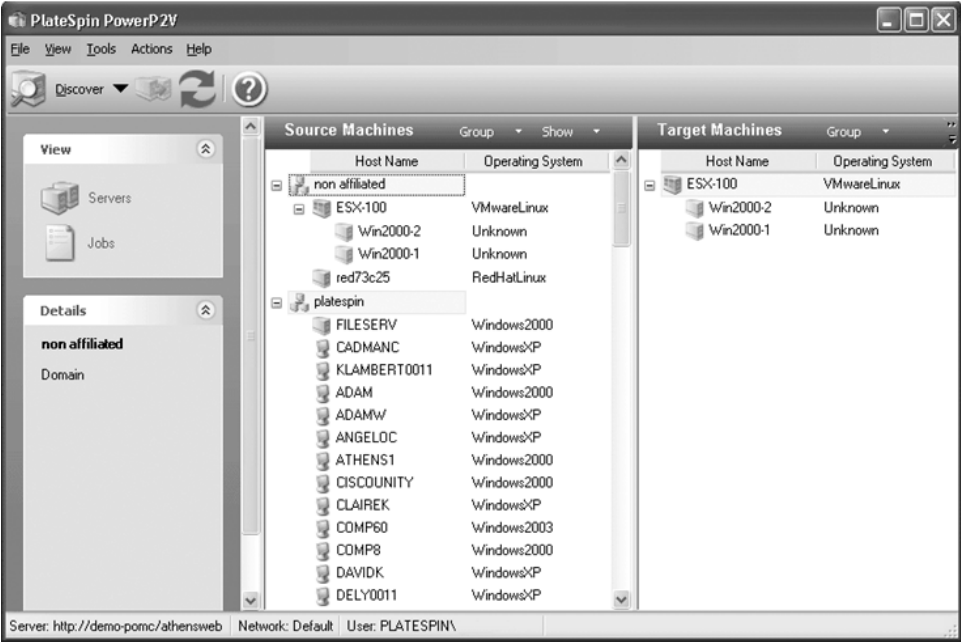


Figure A-6. Selecting the source machine

Step 2: Configure Credentials, Network, Disk, and Memory Parameters for the Target Virtual Machine

PlateSpin PowerP2V enables users to quickly and easily set up target virtual machine parameters before the conversion begins. The discovered details of the source machine are used to autoconfigure the conversion job with the same machine parameters, including networking, drive configuration, and memory allocation. All of these parameters can be changed, enabling the user to right size the server in addition to the conversion. This process usually takes five to ten minutes of a user's time. When the user is finished with the configuration, he or she simply presses Start to initiate the conversion. Conversion times vary depending on network bandwidth and amount of data that needs to be transferred from the source server.



Figure A-7. Configuring target VM parameters

Step 3: Press Start, and Monitor the Conversion Job(s) Live

The user is able to monitor the progress of multiple conversions in granular, step-by-step detail from a single station. PlateSpin PowerP2V can handle multiple conversions at the same time.

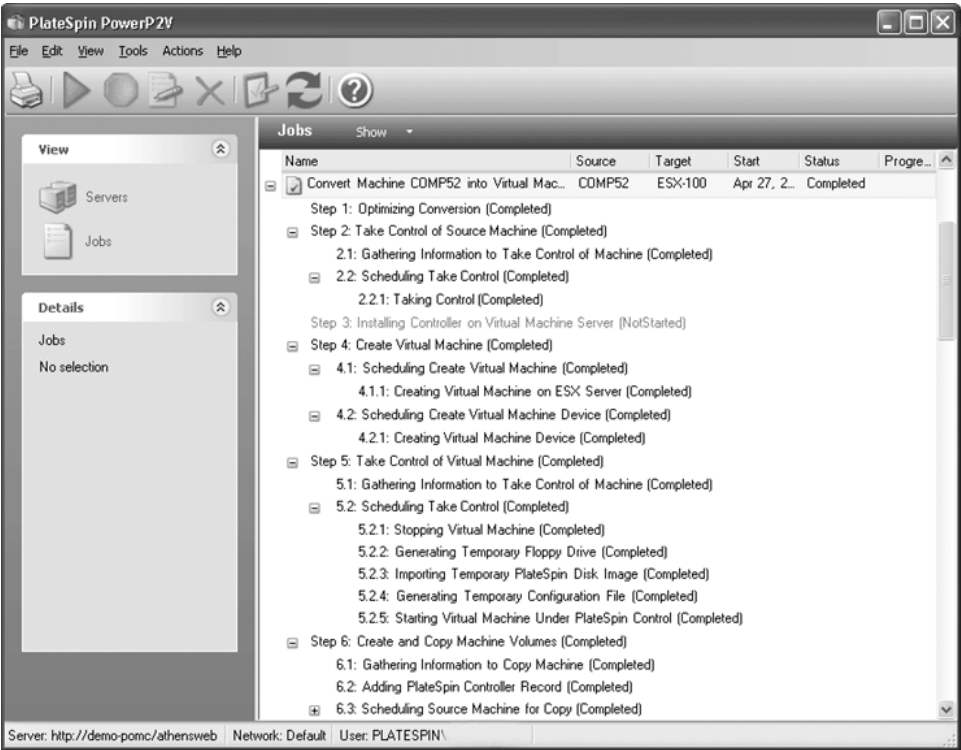


Figure A-8. Monitoring the conversion jobs

Replicating Entire Test Lab Environments Using Virtual Machines

Data centers in 2004 are already adopting virtual technology in their test lab environments. Most companies are starting with smaller-scale projects, such as consolidating their test lab hardware into virtual machine infrastructure. Data centers can use virtual environments to replicate an entire farm of production servers on a fraction of the number of hardware that it would normally require. By stacking five, ten, or even fifty virtual machines into one virtual machine host on a single physical server, data centers can realize substantial savings in hardware expenditures.

However, the act of consolidating those servers usually means an undue amount of manual effort is required to replicate or rebuild test systems in a virtual environment. PlateSpin PowerP2V allows users to perform conversions between physical and virtual environments with almost no manual effort and little knowledge of virtual environments.

Using Virtual Machines As Hot Backup Servers for Planned and Unplanned Downtime

Data centers in 2004 are also starting to use virtual machines for high availability during planned and unplanned outages. Planned outages occur when data centers need to shut down servers temporarily to perform hardware or software maintenance. Typically, planned outages are done during offpeak hours, when transactions are low and customer impact is minimal. Ideally, users are not affected at all; however, in reality the only alternative is to keep the downtime window as small as possible. The amount of sophistication in a high availability environment can be separated into two categories:

- **Disaster recovery (tape backup):** Can accommodate lower priority servers that can afford to be down for days
- **High/managed availability (point-in-time replication):** Suitable for servers that can only afford to be down for hours
- **Continuous availability (continuous replication or clustering):** Suitable for servers that cannot afford to be down at all—instant failover required

Using PlateSpin PowerP2V, data centers can replicate a production server to a virtual machine on a point-in-time basis to achieve the second level of availability—high/managed availability—on Windows and Linux servers. In order for a user to initiate the replication process, the user simply selects a source server and drags it into a virtual machine host. The production server is taken offline for a brief period of time while the conversion takes place (length of time depends on network bandwidth and the amount of data that needs to be transferred). When the conversion is complete, the hot backup is ready to begin serving customers on demand. For example, if the production server needs to be shut down for maintenance, the user takes the production server offline and resets the network configuration of the hot backup server, and end users only see a brief moment of service interruption while the transition from production server to hot backup server is taking place. While the hot backup is active, data center staff now has as much time as necessary to perform the hardware or software maintenance on the physical production server while the virtual replica continues to process transactions and serve users and customers (see Figure A-9).

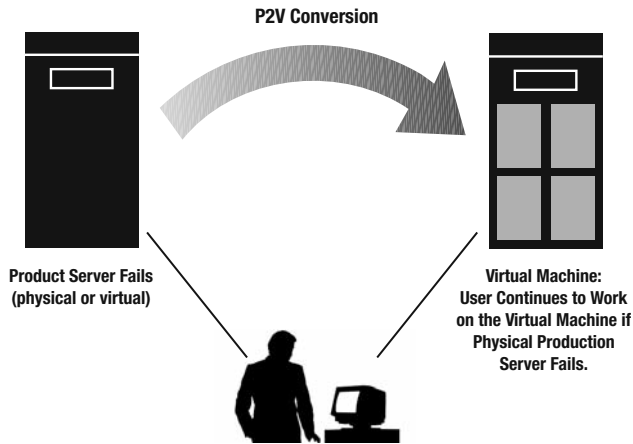


Figure A-9. *P2V conversion*

From a disaster recovery perspective, PowerP2V can be used to replicate a server to a virtual machine on a periodic basis to provide manual hot failover in the case of an unexpected production server outage. If a production server goes down, a user can simply start the hot backup server and reconfigure network settings, and the server can resume processing transactions with minimum interruption to users (minutes instead of hours or days). The primary server can then be restored from tape, while the virtual machine replica continues to serve users.

The bottom line is that using fully automated P2V conversion technology can drastically reduce planned downtime from days to hours and be an efficient addition or alternative to tape backup solutions, which are often very slow to recover.

Moving a Virtual Machine from One VM Host to Another (V2V)

PowerP2V enables users to quickly and easily convert virtual machines from one virtual machine host to another (VMware to VMware), using a process that is similar to a P2V conversion.

Having the ability to easily move a virtual machine from one host to another allows users to easily balance resources between physical machines simply by moving virtual machines around (see Figure A-10).

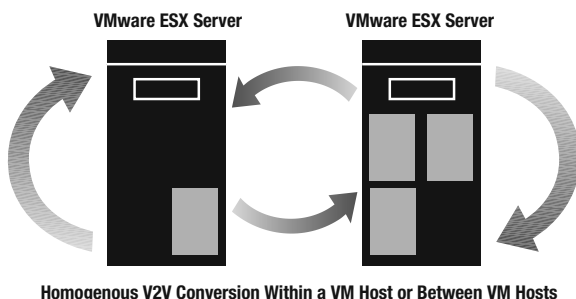


Figure A-10. *Homogeneous V2V conversion*

Using PowerP2V to perform V2V conversions does not require a SAN, and the conversions can be performed across dissimilar hardware. Some other benefits of having an easy way to move virtual machines around between hosts include creating an easy method of upgrading between virtual machine host versions (for example, upgrading between an ESX 1.5.2 host and an ESX 2.1 host or converting GSX virtual machines over to ESX virtual machines across physical hosts).

Production Server Virtualization

As data centers realize efficiencies in the test lab and high availability environments using virtual infrastructure, they will begin to virtualize some of their production environment in the effort to increase utilization of server resources. Today's data center is riddled with production servers that are underutilized and that therefore represents a large amount of potential savings.

Production servers, from a consolidation perspective, can be characterized two ways: batch based and transaction based.

Batch Production Servers

Production servers that have point-in-time processing duties are referred to as *batch-based servers*. Examples of batch-based servers are ones that are used in nightly bank account processing, payroll processing, business analytics processing, or data warehousing. These servers tend to spike to very high levels of resource utilization (90–100 percent CPU) during batch processing times, while utilization rates are near zero during times when the server is not performing batch processing.

Transaction-Based Servers

In contrast to a batch server that performs a few large processes at specific times on a schedule, a transaction-based server performs small chunks of processing in real time. Examples of transaction-based servers include Web servers and application servers. Processing characteristics for a transaction-based server tends to be in the low-to-medium range (10–30 percent CPU utilization on average) and is fairly consistent over time relative to the batch servers, which tend to spike from low to very high utilization for specific time intervals.

Virtualizing the production environment can offer several advantages. Server consolidation can help data centers reduce the number of servers while at the same time improving the utilization of the servers overall. Unused CPU power can be more efficiently utilized (e.g., virtualizing four or five servers with average utilization of 15 percent into one virtual host gives the new server a combined average utilization of 60–75 percent).

In the case of batch-based servers, data centers will take a different approach to determine how to consolidate: the notion of virtualizing several physical servers that perform batch processing based on a staggered resource demand approach. For example, one server may require 100 percent of a server's processing power but only for 20 percent of the time, let's say in the morning hours. Another server may require 100 percent of a server's processing power but only in the evening. By consolidating those two servers into a virtual environment, the combined physical server would realize a higher utilization rate without affecting performance of the individual machines (see Figure A-11).

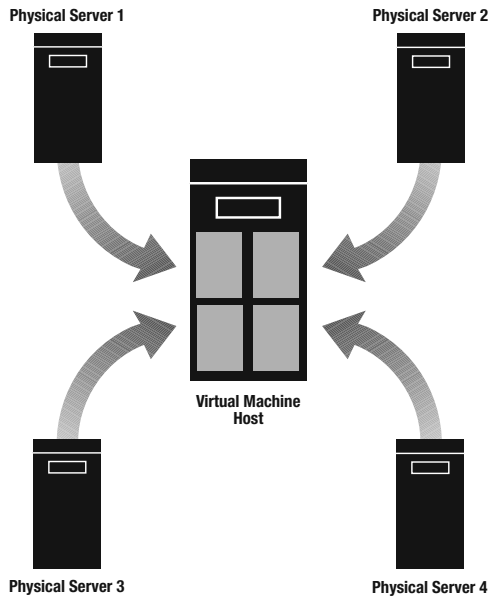


Figure A-11. Consolidating many underutilized physical servers into one VM host

Server Migrations Across Geographic Regions

Geographic data center migration projects normally require users to physically transport servers from one location to another. PlateSpin PowerP2V can help data centers move some or all of its Windows and Linux servers across data centers that span multiple geographic regions without needing to physically transport hardware. Since PowerP2V does not require the user to make direct physical contact with the source or target servers, migrations can be accomplished remotely by simply providing network access between them.

The Need for Continuous Resource Analysis and Rebalancing

After a migration project is complete, the enterprise data center will need to continually optimize and adapt to changing business conditions. For example, a sudden increase in number of customers may place a greater strain on invoice processing or fulfillment processing, or a new service that is brought online may change the resource loads across the server environment. This process of rebalancing will first start off as point-in-time manual efforts. When response times become unacceptable, people will make decisions based on resource utilization statistics to move virtual machines from one physical server host to another to balance the load. While acceptable in the short term, this is largely a reactionary system: people only react when things (SLAs, business services, etc.) are not performing as required, which inevitably leads to adverse customer experiences or problems in business operations.

Rather than being reactive, enterprises stand to gain from adopting a proactive approach. By proactively rebalancing virtual machines on demand, business service levels can be sustained without interruption or degradation. The enabler for this type of dynamic load balancing is server portability: the ability to move the services that a server is providing from one hardware platform to another with zero manual effort.

Dynamic Virtual Machine Portability: Using Virtual Machines to Prevent SLA Violations

Virtual machines allow data centers to more easily adjust resource priorities to match resource demand conditions. For example, if a particular VM is consuming more than its fair share of resources at the expense of other VMs on the same machine that are also contending for those same resources (i.e., the virtual machine host is overutilized), a user has two alternatives:

- If the virtual machine host isn't overloaded, the user can adjust CPU and memory shares for all the virtual machines (VM resource rebalancing).
- Move the resource intensive VM to another VM host that is relatively underutilized (revirtualization).

Having the ability to move virtual machines from one virtual host to another allows enterprises to dynamically rebalance load based on the hardware resources they consume. For example, if a virtual machine is consuming an intense amount of CPU resources from a particular host and another virtual host is available that has more CPU power, the load can be effectively balanced by moving the resource intensive virtual machine away from the overutilized virtual host to the underutilized virtual host (see Figure A-12).

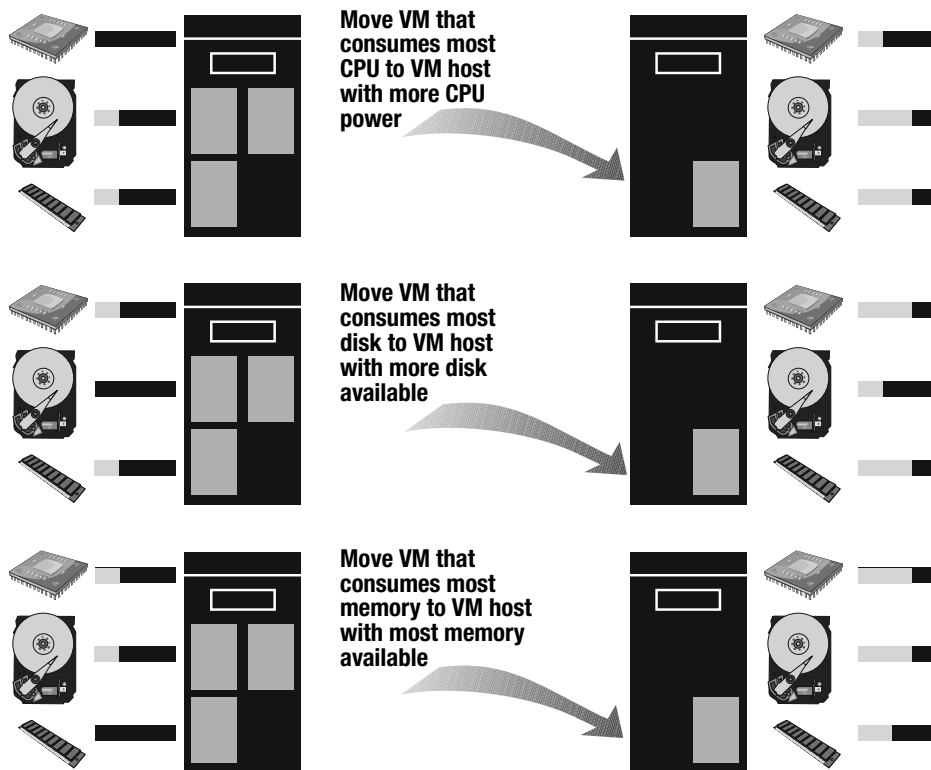


Figure A-12. Having the ability to move VMs across virtual hosts

Integrating PowerP2V with an SLA management tool can give data centers complete virtual load-balancing capabilities through server portability. Load balancing can take place easily, without specialized applications or complex clustering setups, by simply moving virtual machines between physical host systems.

PowerP2V can be integrated easily with third-party resource monitoring applications through its open Web services API. A third-party resource monitoring tool, which can monitor resources on physical servers or virtual machine hosts, can trigger conversions to occur in PowerP2V dynamically across Windows and Linux virtual environments and physical environments without scripting, customization, or manual effort.

How Dynamic Virtual Machine Portability Can Enhance Business Service Management

Business service management (BSM) has become a buzzword in the IT community in the last couple of years. BSM is also referred to as *business activity monitoring*, *on-demand computing*, *autonomic computing*, and *self-healing systems* by various vendors and industry associations. It introduces the notion that data centers should dynamically and automatically adapt to changing business conditions based on a combination of business service metrics and IT metrics. Combining IT service metrics and business service metrics will allow enterprises to provide a tighter linkage between IT and business operations. IT staff will see more business relevance, and business managers can see what IT systems are involved in supporting their business operations.

Today's data center is typically static. To cope with increasing workloads that cause SLA violations, the data center goes through a long cycle of obtaining new and faster hardware, manually provisioning the new servers, and installing patches and applications. Additionally, all changes are based on IT service levels as opposed to business service levels. The drawback of using an IT-only approach to optimizing load is that server resources are not being prioritized based on underlying business services, which means that more IT resources may be given to business services that are lower in priority and fewer IT resources may be given to business services that are higher in priority.

Let's look at some business services for a simplified version of a financial institution. In this bank, there are five business services (shown in Figure A-13). The online stock trading service has been experiencing very slow response times when end users want to place orders (it is taking one minute to fill a market order, which is much greater than the established acceptable length of time of only ten seconds). Another business service, the check processing service, is overachieving its business metric: it only takes five days to process a check, which is well under the fifteen-day processing minimum.

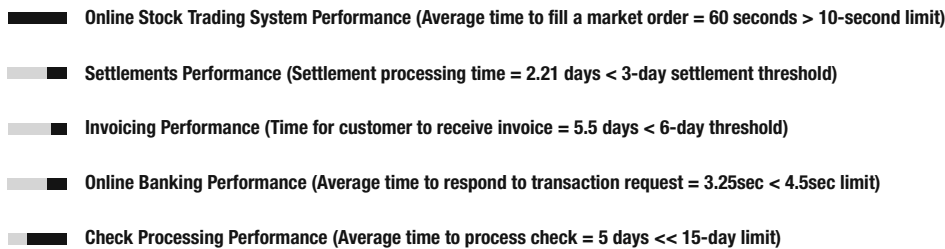


Figure A-13. Five business services

From an IT perspective, the two business services can be illustrated in Figures A-14 and A-15. For simplicity, let us assume that all servers are virtual machines.

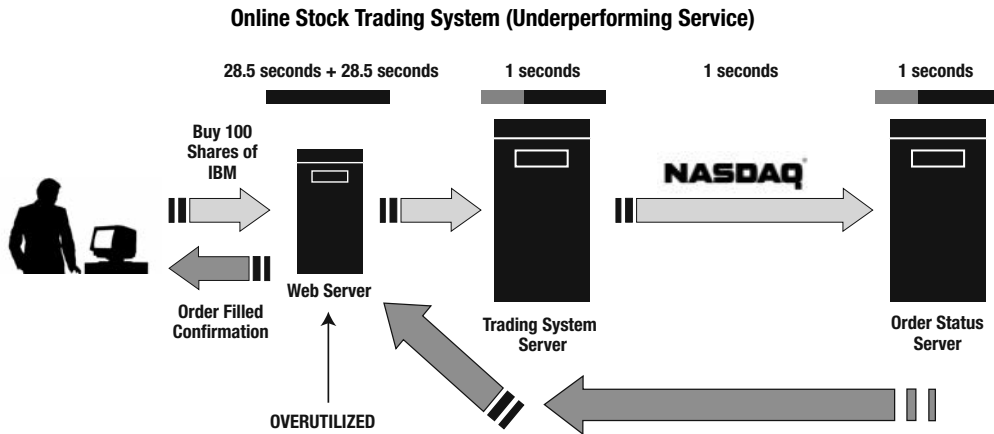


Figure A-14. Underperforming service

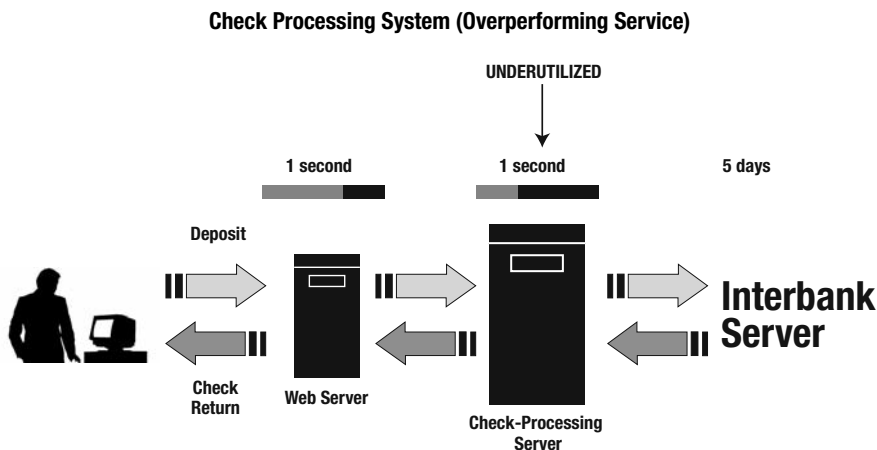


Figure A-15. Overperforming service

The online stock trading system is an underperforming business service because it takes sixty seconds to fill a market order, which violates the minimum level of response time of ten seconds by a large margin. You can see in this simple example that the Web server is the “bottleneck” in this scenario. On average, it takes 28.5 seconds to send an order and 28.5 seconds to issue the order confirmation to the end user. The Web server’s CPU, memory, and disk resources are being overused, which is leading to slow response times for the user. The Web server therefore needs to be moved to a larger server with more resources.

In contrast, the check processing system is a business service that is overperforming because it takes only five days to process a check, which exceeds the minimum level of response time of fifteen days by a large margin. It appears that the check processing server is

taking only one second to process a check. Closer examination of the check processing server reveals that its CPU, memory, and disk resources are very underutilized.

One way to alleviate the problem without having to buy new hardware for the Web server is to swap the check processing server with the Web server using PowerP2V. A business service monitor would be able to see this discrepancy automatically and initiate a V2V conversion for both servers by issuing a Web service call to PowerP2V to initiate the conversions (see Figure A-16).

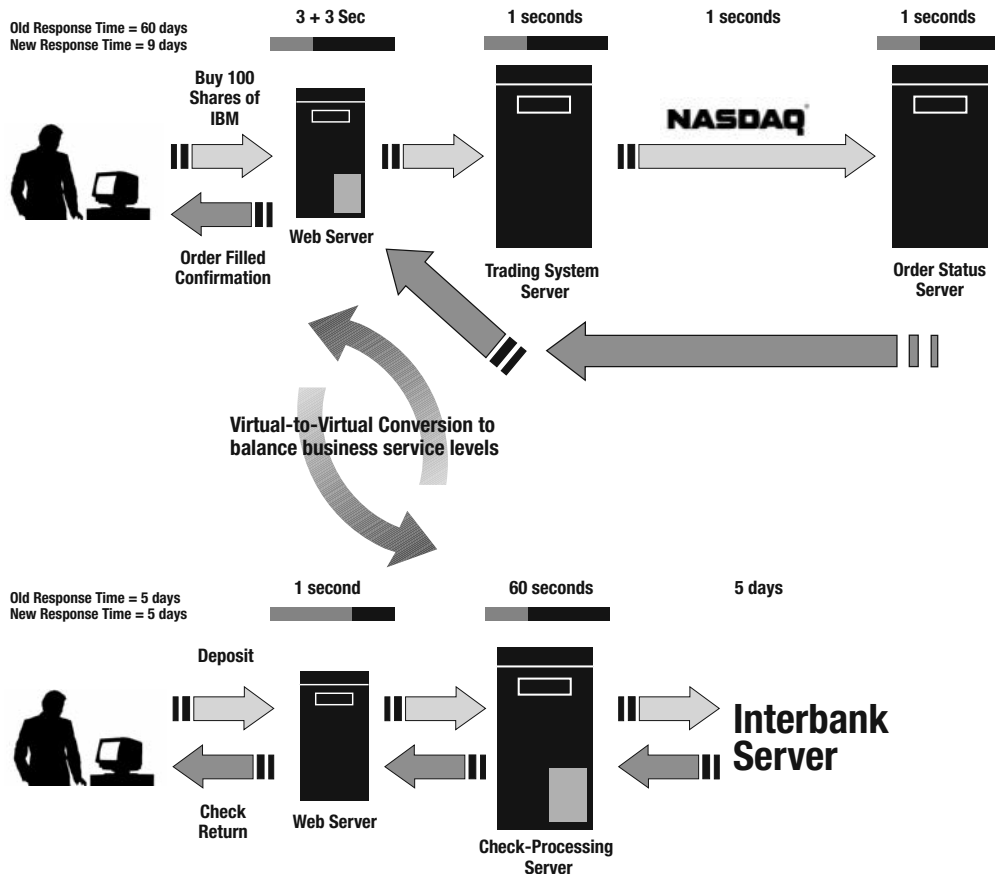


Figure A-16. V2V conversion causes improvement in business service performance.

After the servers are swapped, business server performance has improved: the Web server now resides on the physical hardware that was used for check processing, and the check processing server now resides on the physical server that was previously used for the Web server. The Web server now processes a transaction in 6 seconds rather than 28.5 seconds, and the check processing server processes a transaction in sixty seconds, rather than one second. The result is an improvement in business service level for online trading, at an insignificant performance hit for check-processing services.

Conclusion

The use of server virtualization is beginning to become a popular solution for various data center issues. In order to facilitate the adoption of virtual infrastructures, virtualization conversion technology, such as PlateSpin PowerP2V, will be adopted in several phases: (1) to consolidate test lab hardware and accelerate test lab deployments, (2) to rapidly create backup servers in managed availability environments, (3) to perform virtualization of production server environments easily and increase utilization, (4) to perform virtual-to-virtual conversions for the purposes of load balancing, (5) to provide a means to migrate servers across geographical regions without traveling, (6) to integrate with SLA management tools for dynamic virtual machine load balancing, and (7) to integrate with BSM tools to enable on-demand balancing of workload based on business services.

Note Again, the previous material comes directly from “Server Consolidation and Beyond: Enhancing Virtual Machine Infrastructure Through Automation,” written by PlateSpin (2004).

Rocket Division Software

The following sections provide an overview of Rocket Division Software's iSCSI target and iSCSI initiator software products. Both products are included on the companion CD and are available for download at <http://www.rocketdivision.com>.

Note The following material comes directly from the white paper “StarWind (iSCSI Target) and StarPort (iSCSI Initiator),” by Rocket Division Software (2004).

iSCSI Target

Overview

StarWind is a complete iSCSI target implementation for Windows. It enables practically everybody to benefit from extremely low-cost IP network storage solutions. Utilizing the state-of-the-art iSCSI protocol, StarWind enhances the Windows sharing of storage subsystems over an existing IP network.

Benefits

Cost effective: StarWind is industry's most affordable iSCSI target solution for Windows—any existing computer can be morphed into a high-performance iSCSI to Fibre Channel switch, ultracapacity SATA external storage enclosure, “networked” DVD burner or tape drive, incremental backup, and disk-to-disk backup facility. And this is for maybe one-tenth the price that a pure hardware solution (which is slower, single-purpose, and expensive to install, maintain, and own) would cost.

Easy to install and run: StarWind can be installed and configured over the network in seconds with the help of a standard Telnet application or the StarWind network configuration console. Anybody, even without Windows administrator skills, can do this. New network services and managed iSCSI devices will be ready to use without rebooting and without server service downtime.

High efficiency: StarWind wisely uses all possible network and storage device interfaces at their full-stroke speed. StarWind can happily coexist with any network services (like file and printer sharing) already running; installing StarWind is a fast and easy way to add SAN (Storage Area Network) features to an already present NAS (Network Attached Storage) server.

High availability: StarWind adds a network interface to any of the storage devices it has been configured to manage. Bright new features like the ability to create locally accessed RAID with “networked” storage devices and the ability to do direct backups to optical and tape devices over the network highly improve application availability.

Storage centralization: StarWind provides excellent features that help migration from DAS (Direct Attached Storage) to true network storage. A whole storage pool can be managed universally; there's no need to mess with an “islands” of disks gathered around every single server.

Easy to upgrade: StarWind provides excellent hardware and software upgrade capabilities. New hardware attached to a running machine can be configured to be managed by StarWind immediately. New StarWind modules (like custom iSCSI devices such as sophisticated incremental backups, disk-to-disk-to-tape backups, or tape drive emulation from third-party companies) can be installed also without server downtime. You just need to apply the upgrade application and restart the StarWind service.

Features

Easy to explore and use: The StarWind network configuration console dramatically reduces software learning time. True gurus can use Telnet and edit configuration files directly, however. Such flexibility allows StarWind software to be used by anybody with some basic knowledge.

Native hardware support: StarWind out-of-the-box supports all storage devices available to Windows: Serial and Parallel ATA, Serial and Parallel SCSI, Fibre Channel, FireWire, USB, etc. Every storage device can be “networked” by StarWind, and StarWind does not intrude into bypassing command and data traffic (this is *very* different from most of the other iSCSI targets dealing with virtual hardware only). Such a feature allows true hardware to be redirected transparently: a locally mounted DVD burner would be accessed from the remote machine just as it would be physically mounted on a remote machine. No DVD-burning application would notice the completely invisible network “redirection” layer. The same statement is true for tape backup applications and tools dealing with hard disks at the sector level (hard disk backup, data migration, file system and partition management, huge databases, etc). This allows you to treat the whole network as a single megacomputer. It does not matter where physical device resides—it can be accessed from everywhere if needed. Any application working with local storage device automatically works with the “networked” device just fine.

Hardware virtualization support: StarWind also takes care of virtual hardware. It not only maps hardware devices *as is* but it also can emulate hard disks from partitions and image files. This means a single Windows dynamic disk can be used to store multiple iSCSI hard disk images (as raw partitions or just as an image files residing on host file system). When needed, the dynamic disk can be increased in size, and new iSCSI hard disks would be created. This allows a more flexible approach in storage management. Image files can be backed up to tape or optical media, and they can be compressed or encrypted at the file-system level; sparse (unallocated until used) files can be used to handle iSCSI disk images also. StarWind also can emulate DVD media. This means standard ISO or MDS DVD images can be exported as iSCSI DVD drives. StarWind out-of-the-box comes with DVD library management capabilities. Another quite important feature is RAM disk support. This is extremely useful to tune network and configure hardware in an optimal way.

Flexibility and extendibility: StarWind comes with documented APIs for plug-ins. This means every single company-purchased developer's license can create its own virtual hardware (best suiting company needs) in just a matter of days. Storage hardware prototyping has never been so easy. There is no need to invest a small country's budget in expensive storage hardware: StarWind plus custom plug-in modules do the required job much better on already present hardware.

Low resources usage: StarWind does not require any expensive hardware or software to run. Even a three-year-old department server running Windows 2000 can be converted into a powerful external RAID box. And it still takes care of printer and file sharing and e-mail delivery.

Network security: StarWind supports CHAP for user and initiator authentication and should work on the top of existing VPN (Virtual Private Networking). If required, a single intercepted storage traffic byte means nothing to an unauthorized person.

iSCSI compliant: StarWind is completely interoperable. This means it can work with any iSCSI initiator completely supporting the iSCSI protocol. Operating system/processor architecture "on the other side" just does not matter—any should work.

StarPort iSCSI Initiator, RAM Disk, and Virtual DVD Emulator

StarPort is a complete iSCSI initiator, RAM disk, and virtual DVD emulator implementation for Windows. It enables practically everybody to benefit from extremely low-cost IP network storage solutions, huge amounts of installed RAM, and hard disk space converted into virtualized hardware. Utilizing the state-of-the-art iSCSI protocol, StarPort enhances Windows sharing of storage subsystems over an existing IP network, pairing client software with iSCSI targets. Also, it does an excellent job of adding features not available to Windows out-of-the-box (RAM disk and virtual DVD emulation).

Benefits

Cost effective: StarPort is industry's most affordable iSCSI initiator solution for Windows. Plus, for iSCSI initiator features, it provides services (RAM disk and virtual DVD emulation) available only as commercial add-ons. Thus, StarPort is capable of replacing the generic iSCSI initiator, the RAM disk emulator, and the DVD emulator. This is a three-in-one value package.

Easy to install and run: StarPort can be installed and configured in seconds. Anybody, even without Windows administrator skills, can do this. New network services (iSCSI) and managed virtual hardware devices (RAM disks and virtual DVDs) will be created and be ready to use without needing a reboot.

High availability: StarPort adds a network interface to any of the storage devices it has been configured to connect to. Bright new features like the ability to create locally accessed RAID with “networked” storage devices and the ability to do direct backups to optical and tape devices over the network highly improve application availability (if paired with the proper iSCSI target, StarWind works just great).

Features

Easy to explore and use: The StarPort configuration console dramatically reduces software learning time.

Any hardware support: StarPort out-of-the-box supports all storage devices “networked” by the iSCSI target (as StarWind). It acts as a dumb redirector, not intruding into any storage device data or command traffic.

Low resources usage: StarPort does not require any expensive hardware or software to run. Virtually any Windows box with GbE (or even 100Mb) works fine.

Network security: StarPort supports CHAP for user and initiator authentication and should work on the top of existing VPN (Virtual Private Networking). If required, a single intercepted storage traffic byte does not mean anything to an unauthorized person.

iSCSI compliant: StarPort is completely interoperable. This means it can work with any iSCSI target completely supporting the iSCSI protocol. Operating system/processor architecture “on the other side” just does not matter—any should work.

Mission and Technologies

Rocket Division Software is a rapidly growing company that specializes in providing cutting-edge system software solutions for Windows NT/2000/XP/2003 and various Unix operating systems.

We’re a leading provider of top-notch, high-performance technologies for the data storage and networking industry. Our “know-how” and development services cover a wide range of existing and emerging storage and networking technologies, such as

- CD/DVD recording and mastering
- Tape backup and restore
- iSCSI
- Virtual storage
- TCP, SCSI, and file-system filtering
- Storage-over-IP (SAN and NAS)
- Local and network file system design

Our cross-platform StarWrap (runtime library) SDK, StarPort (monolithic SCSI port) SDK, StarBurn (CD/DVD recording and mastering) SDK, and StarTape (tape backup and restore) SDK dramatically reduce development time for virtually any of the device driver projects (especially network and storage ones, of course).

Offering a wide variety of services including training, consulting, and testing, Rocket Division Software is keeping in touch with the end-user market. Our technologies are driving not only other companies' flag products but they also act as a core for our own software solutions targeted at the end users.

Market Focus

The key to Rocket Division Software success is its very special focus. We're not strictly limiting our activities with Windows NT/2000/XP/2003 systems-level software solutions but rather doing our best in making such a solution as portable as it can be in theory.

Software developed by using our StarWrap (Run-Time Library) SDK should run out-of-the-box on all the platforms our RTL supports (currently that's NT native, NT kernel, Win32, and Linux user-mode environments, with more ports coming soon).

Such flexibility allows our customers to neglect the initial market orientation for their software solutions and avoid dealing with the system-level programming at all. We'll take care of all this!

Note The previous material comes directly from the white paper "StarWind (iSCSI Target) and StarPort (iSCSI Initiator)," by Rocket Division Software (2004).

Network Instruments' Observer

Network Instruments' Observer provides a robust set of network monitoring features that can assist in the analysis of your virtualized infrastructure. In the following sections, you'll see a case study that portrays Network Instruments' Observer in action.

Note The following material comes directly from "Case Study: Jack in the Box," by Network Instruments (2004).

Too Much Traffic in the Kitchen

Jack in the Box, Inc., uses Observer Suite to monitor broadcast storms, gain more efficiency from the corporation's national networks, and profit from proactive network management.

Two Unique Networks Working Together

The IT staff at Jack in the Box oversees a vast corporate network beginning at headquarters in San Diego to regional field offices nationwide. The company maintains a second network

linked via satellite that shares traffic from headquarters to over 2,000 different locations across 30 states. The entire infrastructure consists of over 4,000 nodes.

“We separate our corporate traffic from our restaurant traffic,” explains Jim Antoshak, network support manager. “Corporate traffic is much more varied and comprehensive, and we don’t need to make our restaurant traffic overcomplicated.”

Antoshak chose to deploy Observer Suite and remote probes from Network Instruments across both company networks. In working with various routers across the network, Antoshak faced many challenges while monitoring different types of traffic being passed from site to site. For example, a WAN deals with potentially lower speeds than a LAN.

“We’re comparing sub-T1 speeds with fast Ethernet speeds,” Antoshak explains. “We needed to quickly view all the WAN traffic that was out there to maximize our pipe.”

Maximizing the Network

Deploying Observer and remote probes from Network Instruments was the answer. The Jack in the Box IT team can now view WAN traffic in total and by specific device. Right away, the team determined there was much unneeded traffic being sent at inopportune times.

“We were quickly able to make our WAN more efficient by controlling what data was being sent and when,” said Antoshak. “This was the key aspect to relieving our network congestion.”

Complete Network Control

Observer allows network professionals to understand and ultimately control their network. For example, with Observer, network administrators can run baseline reports immediately to gauge what is normal for the network. Once a baseline is complete, administrators may compare the network over time to the baseline to recognize unusual activity, congestion, and other network issues. Before installing Observer, Jack in the Box did not have a method to determine current network traffic levels. Now, increased traffic, broadcast storms, and congestion are instantly recognized. By using Observer’s Real-Time Expert, network managers are immediately alerted to network abnormalities and given plausible solutions.

“First I’m notified about issues on particular segments, like broadcast storms, through alarm devices that I configure,” said Antoshak. “Then Observer provides me with ideas of causes and solutions to the problems. The Observer Expert Analysis feature is excellent for this.”

Capacity Analysis

Observer’s “What-if” Analysis lets Jack in the Box forecast the impact of possible network or application changes without investing in new equipment. This “live-modeling” predicts network-bandwidth and response-time impacts for topology changes (e.g., upgrading from 10/100 to Gigabit Ethernet) or by changes in variables such as average packet size, client-to-server packet ratio, latency, server load, and number of users.

“The key benefit with Observer is that instead of throwing bandwidth at our network every time we had congestion problems, we’ve been able to take our existing capacity and make it more efficient,” said Antoshak.

Proactive Management

With Observer, Jack in the Box network administrators can not only monitor their networks but also proactively manage it. Through “What-If” Analysis, Network Trending/Reporting, and other features, they’re preparing to meet tomorrow’s challenges.

“A lot of times we’re preventing future problems. It’s not that I don’t want users to notify me with issues, I just don’t want them to see problems to begin with,” said Antoshak. “By pushing the existing topology, we’re embracing proactive management of our entire network.”

About Jack in the Box, Inc.

With over 44,000 employees and serving over half a billion customers in 30 states, Jack in the Box, Inc., is among the nation’s leading fast-food hamburger chains. Founded in 1951, the company operates and franchises over 1,900 Jack in the Box restaurants, franchises over 100 Qdoba Mexican Grill restaurants, and also owns about a dozen proprietary Quick Stuff convenience stores.

Challenge

Jack in the Box wanted to lower IT costs by gaining more efficiencies out of the corporation’s current network. A complete solution was required to monitor all WAN traffic as well as a method of relieving network congestion without adding more equipment.

Solution

By introducing the Observer console and remote probe solution into the network, Jack in the Box can now monitor all WAN traffic in total and by specific device. Expert Analysis warns administrators of incoming broadcast storms and other abnormalities and then provides possible solutions. Through Observer’s “What-If” Analysis, Network Trending/Reporting, and Event Alarms, the IT organization can proactively manage the corporate network.

With Observer, instead of throwing bandwidth at our network every time we had congestion problems, we’ve been able to take our existing capacity and make it more efficient.

—Jim Antoshak, network support manager, Jack in the Box, Inc.

About Real-Time Expert

Observer’s Real-Time Expert allows critical events to be collected (through Expert Summary), breaks down conversations into subprotocol groups (Expert Events), and identifies problems as they happen.

About Advanced Single Probes, Advanced Multi-Probes, and Advanced Expert Probes

Through the deployment of Network Instruments’ probes, network administrators can remotely monitor the entire network from one location. Administrators are then able to view statistics in real time, produce long-term trending reports, and capture packets for in-depth

analysis on any remote network. Network Instruments also offers hardware-based gigabit probes and hardware-based WAN probes that passively capture network traffic—offering no disruption to the network.

Advanced Single Probes monitor different locations on the network and offer single session and single interface support. Advanced Multi-Probes supports multi-interface and multisession monitoring. With multi-interface, one Observer console can simultaneously monitor several probes (in various locations) that use multiple NICs to view different topologies. Multisession permits multiple administrators in different locations to monitor the same probe at the same time—allowing for problem-solving collaboration. Advanced Expert Probes perform full Expert Analysis and packet captures and decodes at remote locations for faster troubleshooting. Only results, or screen updates, are transferred back to the console, reducing network traffic.

About Network Trending

Observer offers Network Trending to help administrators understand what is normal for their network. Network Trending provides the baselining necessary to collect, store, view, and analyze network traffic statistics over long periods of time. Through Network Trending, administrators can continually monitor their network health and recognize signs and symptoms of network inefficiencies.

About “What-If” Analysis

Observer’s “What-If” Analysis offers a predictive tool for modeling potential response times, utilization or packets-per-second at different network speeds. With “What-If” Analysis, network administrators can test different conventional and network metrics to predict changes in performance.

About Network Instruments, LLC

Network Instruments is the industry-leading developer of distributed, user-friendly, and affordable network management, analysis, and troubleshooting solutions. The award-winning Observer family of products combines a comprehensive management and analysis console with high-performance probes to provide integrated monitoring and management for the entire network (LAN, 802.11a/b/g, Gigabit, WAN). All Network Instruments products are designed utilizing Distributed Network Analysis (NI-DNA) architecture. With NI-DNA, the Observer solution set simplifies network troubleshooting and management, optimizes network and application performance, and scales to meet the needs of any organization. Founded in 1994, Network Instruments is headquartered in Minneapolis, Minnesota, with offices in London, Paris, and throughout the United States and distributors in over 50 countries. More information about the company, products, innovation, technology, NI-DNA, becoming a partner, and NI University can be found at <http://www.networkinstruments.com>.

Note Again, the previous material comes directly from “Case Study: Jack in the Box,” by Network Instruments (2004).

Index

Numbers and symbols

- # (pound sign), for Perl script comment lines, 192
- “ ” (double quotes), using when modifying the Target field, 129
- ; (semicolon), ending for all Perl statements, 192

A

- abstracted operating systems, defined, 2
- abstraction, defined, 2
- access control
 - basic goals of, 459
 - in SNIA Shared Storage Model, 459–460
- access control lists (ACLs). *See* ACLs (access control lists)
- access paths, from application layer to lowest storage layer, 457
- Access Rights, setting for VMs, 174
- access time, effect of on VMs, 57
- Account Configuration screen, in ESX Server installation process, 159
- ACE. *See* VMware ACE
- ACLs (access control lists), used by AFS for providing security, 290–291
- Active Directory, adding Samba to, 285–286
- active termination, for single-ended (SE) SCSI devices, 420
- active-active failover clusters, primary reasons for using, 319
- active-active server clusters, basic configuration of, 312–313
- active-passive failover clusters
 - basic configuration of, 312
 - example of two node, 312
- adapter bonding, from ESX Server, 50
- adapter type, choosing for your VMs, 175
- Add Hardware Wizard
 - for adding virtual hardware to VMs, 96–97
 - Specify Named Pipe dialog box in, 100
- Add Nodes Wizard, installing a second node in, 405–406
- Add Target Portal dialog box, adding an iSCSI target portal in, 401
- Add/Edit Port Rule dialog box, entering filter parameters in, 368
- ADDLOCAL and REMOVE options, qualifiers, 144
- Administration Web site
 - configuring for Virtual Server, 140–141
 - getting a certificate for, 141–142
- Advanced Digital Information Corporation, web site address, 427
- Advanced Maryland Backup Disk Archiver (AMANDA), Linux backup tool, 238
- Advanced Settings dialog box, configuring CHAP authentication in, 402
- Advanced TCP/IP Settings dialog box, showing NLB cluster IP address, 363
- Affinity, setting for NLB clusters, 367
- AFS (Andrew File System). *See also* Andrew File System (AFS); ArlaAFS; OpenAFS
 - implementing, 292–301
 - installing for Linux systems, 302–307
 - introduction to, 288–292
 - security provided by, 290–291
 - suggested tertiary pathnames, 289
 - using blanket permissions with system groups, 291
 - using replication and caching to increase performance, 290
- AFS cell, standard naming conventions for first, 290
- AFS client
 - cell name configuration, 300
 - configuring, 300
 - drive mapping, 301
 - installing from the Choose Components screen, 298–299
- AFS database services, configuring to start for Linux VMs, 303
- AFS file server, four roles any can perform, 292
- AFS filespace, considerations when creating, 288
- AFS namespace, guidelines for creating, 288–289
- AFS partition, creating in AFS Server Quick-Start Wizard, 296
- AFS server
 - adding AFS client functionality to Linux VM, 304–306
 - assigning and setting a cell name for Linux VM, 303
 - configuring encryption key, 304
 - creating user accounts and directories in, 297–298

- AFS server (*continued*)
 - creating volumes in, 298–299
 - services to make the DFS function properly, 289
 - AFS Server Manager dialog box, creating volumes in, 298–299
 - AFS Server Quick-Start Wizard
 - Administrative Information screen in, 295
 - for configuring your Windows system server, 294
 - creating an AFS partition on the file server in, 296
 - host cell final configuration screen, 297
 - agent-based backups, performing
 - traditional, 224–230
 - AMANDA. *See* Advanced Maryland Backup Disk Archiver (AMANDA)
 - Andrew File System (AFS), introduction to, 263, 288–292
 - APIC (Advanced Programmable Interrupt Controller) mode, using when installing ESX Server, 156
 - APIC/graphical mode, choosing when installing ESX Server, 156
 - append mode disks, used with VMware ESX Server, 15
 - application virtual machine (AVM), function of, 3
 - applications
 - removing unused to recover disk space, 80
 - selecting to run on load-balanced clusters, 351
 - arbitrated-loop topology, for Fibre Channel networks, 427–428
 - arcade emulators, 4
 - ArlaAFS, web site address, 288
 - asynchronous transfer mode (ATM) network, using NAT to connect virtual networks to, 19
 - ATM network. *See* asynchronous transfer mode (ATM) network
 - authKeys file configuration, for Linux H-A clusters, 336–337
 - Automated Cartridge System Library Software (ACSLs), by StorageTek, 468
 - Autorun, disabling before installing Virtual Server from CD-ROM, 140
 - AVM. *See also* application virtual machine (AVM)
- B**
- backing up, VMware Workstation, 111–114
 - backup agent
 - defined, 224
 - running on the host, 230
 - Backup Job Information dialog box, using in Windows Backup utility, 233–234
 - backups
 - performing traditional agent-based, 224–230
 - tools supporting most popular, 231–239
 - base image, steps for sharing, 14
 - batch-based production servers, virtualization of, 513
 - BDC (backup domain controller). *See also* Microsoft backup domain controller
 - creating to provide redundancy to Samba PDC, 284
 - Beginning Perl*, Second Edition (Apress, 2004), by James Lee, 192
 - Beginning SUSE Linux: From Novice to Professional (Apress, 2005), by Keir Thomas, 154
 - best practices, implementing for a virtual machine host, 35–36
 - BIOS, for virtual machines, 24–25
 - Black Screen of Death, in ESX Server, 160
 - block aggregation
 - function of, 453–456
 - how it is done, 454–455
 - where it can be done, 454
 - block-based storage architectures
 - application of SNIA Shared Storage Model, 455–456
 - combining with file-based storage architectures, 456–457
 - block-level striping with distributed parity. *See* RAID 5 (block-level striping with distributed parity)
 - board form factor, importance of when selecting a motherboard, 44
 - Bochs
 - computer emulator written in C++, 4
 - open-source virtualization application, 4
 - web site address, 4
 - boot process, sequence of for computers, 25
 - bridged networking
 - benefits of, 47
 - supported by Microsoft and VMware, 20–21
 - of virtual machines, 52
 - bridge/router. *See* Fibre Channel bridge/router
 - Brocade, web site address, 426
 - Brocade SilkWorm 2400 switch, configuring zoning on, 429
 - buffer cache, on physical hard drives, 57
 - bus sharing, in ESX Server, 66
 - bus types, researching and choosing for VM host motherboards, 42–43

business service management (BSM), how
dynamic VM portability can
enhance, 516–519
byte conversion comparison table, 413

C

cache. *See* buffer cache
caching, using to increase AFS performance,
290
caching appliance, adding to storage
networks for caching functions, 458
“Case Study: Jack in the Box,” by Network
Instruments (2004), using Network
Instruments’ Observer, 523–526
CD/DVD Drive option, for changing drive
mapping for guest VMs, 107
CD/DVD drives, adding for Microsoft Virtual
Server VMs, 195
CD-ROM drive, *.filename value is mount
point for device, 208
Cell, defined, 288
cell and server information screen, in AFS
Server Quick-Start Wizard, 294
CHAP authentication, configuring, 402
Checkpoint Manager, in Microsoft Cluster
Service (MCS), 321
child differencing disks, 14
Choose Components dialog box, OpenAFS,
293
“Choosing an Availability Strategy for
Business-Critical Data”, in Windows
Server 2003 Deployment Kit, 270
Cisco Systems, web site address, 426
classic storage model, 446–447
classless interdomain routing (CIDR)
notation, defined, 226
CLI executable arguments, case sensitivity of,
131
client access licenses (CALs), Microsoft
requirements for guest and host VMs,
31
Close setting option, controls displaying
message on shut down, 108
Cluster Administrator
adding new resources to a group, 330
analysis of the cluster configuration by,
326
installing Windows Server 2003 Cluster
Service with, 405
moving a group to another node, 329
user interface, 328
using, 328–330
cluster communications, defined for failover
clusters, 311
Cluster Configuration folder, function of in
Cluster Administrator, 329

cluster disk, defined for failover clusters, 311
cluster hardware, configuring, 319–320
cluster IP address, setting for NLB clusters,
366
cluster network configuration,
considerations when planning, 316
Cluster Network Driver, in Microsoft Cluster
Service (MCS), 321
cluster node, defined for failover clusters, 311
Cluster Parameters dialog box, for
configuring NLB cluster parameters,
408
cluster resources and groups, in Microsoft
Cluster Service (MCS), 320–321
Cluster Service. *See also* Microsoft Cluster
Service (MCS)
resources controlled by, 320
cluster service software, for managing shared
storage access, 310
clustering. *See also* failover clusters
defined, 49
cluster-node heartbeat, changing timing and
failure criteria, 356
CMOS batteries, importance of immediate
availability of, 45
code example
for adding a search directory to a path in
Linux, 150
for adding ESX Server serial port support,
183
backing up a Linux VM on a Windows
host, 245
for backing up all VM’s on a Virtual Server
host, 254
for backing up a single VM on a Virtual
Server host, 256
for backing up a VM on a Linux GSX Server
host, 250–251
for backing up a VM on a Virtual PC host,
251–252
for backing up a VM on a Windows GSX
Server host, 249
backing up a Windows VM on a Linux
host, 246–247
for backing up a Windows VM on a
Windows host, 242–243
for backing up Linux VMs with dump
command, 235
for backing up Linux VMs with the tar
command, 237
for configuring static routes on the Linux
media server, 226
for configuring static routes on the
Windows media server, 226
for configuring your Linux DFS server, 274

- code example (*continued*)
 - for connecting to a VM-hosted Samba server, 279
 - for converting a preallocated disk to a dynamic disk, 211
 - for creating a directory to mount USB thumb drive, 189
 - for creating a self-signed certificate with `makecert.exe`, 142
 - for creating a virtual IDE disk, 211
 - for creating new virtual disk for ESX Server, 212
 - creating Samba user account for root, 279
 - creating shared directory and configuring permissions, 279
 - of entire syntax for Virtual PC.exe with options, 132
 - for enumerating shares, share type, and comments, 279
 - to export virtual disk from VMFS partitions, 212
 - if not installing DHCP and NAT services for GSX Server, 144
 - to import a virtual disk into a VMFS partition, 212
 - for increasing the size of an existing virtual disk, 211
 - for installing Virtual PC in unattended silent mode, 132
 - for installing Virtual PC with the GUI for current user, 132
 - instructing Samba to become workgroup member, 279
 - for joining the VM to the Samba domain, 283
 - for Linux H-A clusters `authKeys` file, 336–337
 - for Linux H-A clusters `ha.cf#` file, 337–341
 - for Linux H-A clusters `haresources` file, 342–345
 - to list VMFS virtual disks for ESX Server, 212
 - for loading the current module for USB thumb drive, 190
 - for making VMFS volume writable, 212
 - for mounting a virtual disk, 111
 - for mounting/unmounting Linux virtual CD-ROM, 95
 - of a PDC `smb.conf` file, 281
 - for performing an unattended Virtual Server installation, 213
 - of Perl script for mounting a remote share, 177
 - of Perl script for unmounting a remote share, 178
 - for pingging the iSCSI target, 393
 - for renaming a virtual hard disk, 211
 - for restoring a Linux VM tar backup file, 237
 - for restoring data backed up with the `dump` command, 235–236
 - of a Samba configuration file using Kerberos server for authentication, 284–285
 - of script for unmounting USB thumb drive, 192–193
 - showing Samba `dir_share` section, 283
 - showing Samba `homes` section, 283
 - showing Samba `netlogon` section, 282
 - showing Samba `profiles` section, 282
 - for silent installation of GSX Server for Windows, 144
 - for starting a GSX Server VM, 492
 - for starting a Virtual Server VM, 493
 - typical Windows guest VM configuration file, 53–54
 - for writing script to mount USB thumb drive, 192
- collision domains, hosts placed into by switches, 24
- colon commands, in command mode for Vi editor, 162
- COM ports, for the Virtual Server VMs, 197
- Com Ports option, for creating a serial port for a VM, 107
- command-line management, performing, 211–213
- command-line options, using multiple, 129
- Communications Manager, in Microsoft Cluster Service (MCS), 321
- CommVault, for policy-based data management software, 466
- CommVault QiNetix DataArchiver, web site address, 467
- computer account password changes, disabling, 482
- computer components (hardware), introducing those involved with VMs, 8–11, 21–26
- Computer Management MMC, selecting to convert a disk to dynamic, 486
- computer memory. *See* RAM (computer memory)
- Configuration Database Manager, in Microsoft Cluster Service (MCS), 321
- configuration files, finding with Vi editor, 162–163
- Confirm Restore dialog box, using in Windows Backup utility, 233–234
- Connection profiles dialog box, for PenguiNet, 244

- controller chipset, importance of when researching motherboards for VM host, 40–41
- convergence, understanding, 356–357
- convergence events, validating by checking Windows System Event log, 356
- Convert to Dynamic Disk dialog box, for converting a disk to dynamic, 486–487
- Correctix, Microsoft acquisition of Virtual PC from, 26–27
- Counters, for identifying GSX Server network bottlenecks, 220
- Coyote Point Systems, load-balancer hardware provided by, 474
- CPU, for virtual machines, 9
- CPU speed and quantity, importance of when researching motherboards for VM host, 40
- CpuUtilization counter, for Virtual Server baseline and trend analysis, 220
- Create Shortcut Wizard, for creating a shortcut in VM service shortcut folder, 127
- Crimson Editor, for editing Virtual PC *.vmc configuration files, 117
- cron daemon, automating Linux backups with, 238–239
- crontab file, text format, 238–239
- Crossroads Systems, web site address, 427
- Ctrl+Alt
 - for releasing focus from guest VM to host system, 94
 - for releasing the mouse focus back to host system, 179
- Ctrl+Alt+F1 through F12, for switching between shell sessions in ESX Server, 161
- Ctrl+Alt+F2, for gaining ESX Server shell access, 161
- Ctrl+Alt+F7, for returning to the ESX Server Congratulations screen, 160
- Ctrl+C, for breaking out of looping echo requests and replies, 161
- Custom Setup dialog box, VMware Tools for Windows, 93

D

- daisy chained devices, connected to a single SCSI bus, 415
- data caching, in SNIA Shared Storage Model, 458
- data center
 - enhancing flexibility through PlateSpin PowerP2V, 506
 - evolution of virtualization in, 506

- database management systems,
 - implementation of in file/record layer, 451
- dead keys, enabling during ESX Server installation, 156
- dedicated secondary node (active-passive) model, advantages of for failover clusters, 318
- dedicated secondary node (active-passive) split model, for failover clusters, 318
- Defining Disk Partitions screen, in ESX Server installation process, 158–159
- desktop settings, changing to preserve resources, 79–80
- Device Manager, checking for list of installed components, 79
- DFS. *See also* Distributed File System (DFS)
 - AFS server services to make it function properly, 289
 - checklist for deploying in your infrastructure, 265
 - implementing Linux, 271–284
 - limitations of, 266
 - network protocols and ports required to support, 265
 - performing a generic configuration, 266–270
 - web site address for documentation about setting up in Windows, 270
- DFS dependencies, table of, 265
- DFS filespace, publishing in Active Directory to make more accessible, 269
- DFS link
 - creating, 268–269
 - defined, 264
- DFS namespace
 - defined, 264
 - FRS used for, 270
- DFS path, defined, 264
- DFS root
 - configuring for all network file system access, 479–480
 - creating easy access to for Windows and Linux clients, 288
- DFS shares, setting up on Samba server, 286–288
- DFS Support Tools, on Windows 2003 CD-ROM, 264
- dfscmd.exe command-line tool, for scripting tasks related to roots, links, and targets, 264
- dfsgui.msc snap-in, for remote administration on Windows XP SP1 Machines, 264
- dfsutil.exe /pktinfo, for viewing contents of the referral cache, 264

- dfsutil.exe utility, to help in many administrative tasks, 264
- DHCP
 - passing network configuration information to hosts with, 17–18
 - used by VMs to communicate, 17
- DHCP lease process, table of, 18
- dialog boxes
 - Add Target Portal, 401
 - Add/Edit Port Rule, 368
 - Advanced Settings, 402
 - AFS Server Manager, 299
 - Backup Job Information in Windows Backup utility, 233–234
 - Cluster Parameters, 408
 - Confirm Restore in Windows Backup utility, 233–234
 - Connection profiles for PenguiNet, 244
 - Network Load Balancing Properties, 361
 - New Link for creating new DFS links, 269
 - New Resource for adding resources to a group, 330
 - OpenAFS Choose Components, 293
 - Operating System in New Virtual Machine Wizard, 103
 - Restore and Manage Media in Backup utility, 234
 - Scripts Properties for Virtual Server, 196
 - Send Commands for PenguiNet, 244
 - Specify Named Pipe in Add Hardware Wizard, 100
 - using Convert to Dynamic Disk, 486
 - Virtual Hard Disk Location in New Virtual Machine Wizard, 104
 - Virtual Machine Settings, 96
 - Virtual Machine Settings for VM snapshot options, 259
 - Virtual Machine Settings for VMware Workstation, 120
 - Virtual PC Performance options, 137
 - VM Memory selection in New Virtual Machine Wizard, 103
 - VMware Tools Custom Setup, 93
- differencing disks, function of, 14
- differential (HVD) SCSI buses, function of, 416
- DigiAnywhereUSB, for making USB devices available to guests, 187
- direct-attach storage devices
 - function of in block-based storage architectures, 455
 - function of in combined block and file/record layers, 456
- Disk Druid tool, for configuring software RAID for Linux OSs, 443
- disk duplexing, implementation in RAID 1, 437–438
- Disk Management utility, for configuring software RAID for Windows OSs, 443
- disk mirroring. *See also* RAID 1 (disk mirroring)
 - configuration guidelines for standby VM server, 485
 - configuring the mirror set, 487–488
 - copying the files for, 488
 - implementation in RAID 1, 437–438
 - maintaining a standby VM server with, 485–489
- Disk Partitioning Setup screen, for creating partitions in ESX Server, 158
- disk type, specifying for virtual machines, 16
- DiskMaxLun, and DiskMaskLUNs field settings, 65
- DiskMount Utility
 - command for mounting a virtual disk, 111
 - command options, 73
 - default directory for, 111
 - installing, 73
 - web site address for downloading guide for, 113
- diskpart
 - output in Windows XP, 112
 - web site address for assistance with, 112
- disks
 - append mode used with VMware ESX Server, 15
 - differencing, 14
 - dynamic and dynamically expanding, 12–13
 - fixed and preallocated, 13
 - linked and physical, 13
 - persistent and nonpersistent independent, 14–15
 - resizing, 15–16
 - undo and undoable, 13–14
 - virtual hard and virtual, 12
- Display option, available after installation of Virtual Machine Additions, 108
- distributed data management, global namespace as the new paradigm in, 495–504
- Distributed File System (DFS)
 - impact of running across a WAN, 263
 - introduction to, 263–264
 - using in virtual information system, 479–480
- Distributed Lock Manager (DLM) software, for managing shared storage access, 310
- DLLs, forcing to unload on application close, 83

- DNS server, importance of proper configuration of, 274
- domain-based DFS root
 - creating before configuration process, 266–270
 - DFS link limitations, 266
 - vs. stand-alone root considerations, 266
- DOSBox, open-source virtualization application, 4
- double quotes (“ ”), using when modifying the Target field, 129
- dual-channel technology, for motherboard candidates for VM hosts, 42
- dump command, for performing full and incremental backups, 234–236
- DVD/CD-ROM drives, adding to your guest VMs, 98, 182
- dynamic disks
 - advantages of using, 13
 - effect of on VM performance, 175
 - vs. fixed disks for VMs, 89–90
 - specifying for virtual machines, 175
- Dynamic Host Configuration Protocol (DHCP). *See* DHCP
- dynamic VM portability
 - how it can enhance business service management (BSM), 516–519
 - using VMs to prevent SLA violations, 515–516
- dynamically expanding disks, 12–13

E

- ECC RAM, considerations for using in VM hosts, 42
- edit mode, in Vi editor, 162
- educational environments, determining disk size needs for, 61
- EIDE controllers, on good-quality motherboards, 43
- EMC Legato NetWorker HSM, web site address, 467
- Emulex, web site address, 425
- enableSSL script
 - using to reconfigure ESX Server, 165–166
 - web site address for downloading, 165
- end of life operating systems, 6
- Enhanced Integrated Digital Electronics (EIDE) controllers. *See* EIDE controllers
- enterprise data center, need for continuous resource analysis and rebalancing, 514
- enterprise servers
 - deploying and managing production VMs on, 173–222
 - installing and deploying VMs on, 139–171

- EOL (end of life) operating systems, 6
- error correcting code (ECC) RAM. *See* ECC RAM
- Escape key, using to enter Vi editor command mode, 162
- ESX Server
 - adapter bonding from, 50
 - adding parallel port connectivity to guest VMs, 184
 - bus sharing options, 66
 - disabling USB support in, 185
 - enabling serial port support for, 183
 - enabling USB support on, 185–191
 - finding DNS server configuration with Vi editor, 163
 - function of vmkfstools command, 212
 - limitations of virtual switches, 24
 - monitoring performance of, 215–218
 - physical hard drive creation for, 181
 - removing/adding VM hardware devices, 180
 - scripting USB connectivity, 191–193
 - steps for moving VMs to other ESX hosts, 203–204
 - support for LUNs, 64
 - using USB drives to run guest VMs, 186–187
 - web site address for manual, 212
- ESX Server administration, Vi editor commands to memorize for, 162
- ESX Server configuration files, viewing, 161–163
- ESX Server console, making USB thumb drives and external USB hard drives available to, 189–191
- ESX Server installation
 - accepting license agreement, 167
 - configuring network after completing Configuration Wizard, 168–169
 - creating a virtual Ethernet switch, 170
 - increasing and activating swap space, 170
 - network configuration, 168–169
 - reconfiguring after smooth installation process, 170
 - resolving configuration problems with, 166–169
 - Security Settings screen, 169
 - setting speed and duplex of network adapter, 170
 - specifying memory reserved for system console, 167
 - storage configuration, 167
 - swap file configuration, 168
- ESX Server VMs, persistent binding available for, 66

- esxtop command
 - identifying system bottlenecks with, 218
 - interactive commands and command-line options, 217
 - for monitoring ESX Server performance, 217–218
- Ethernet adapter
 - adding to VMs, 99, 183
 - controlling for virtual machines, 117
- Event Log Manager, in Microsoft Cluster Service (MCS), 322
- Event Processor, in Microsoft Cluster Service (MCS), 322
- Exchange-SQL active-active cluster, example of, 313
- expansion slots, considerations for when researching motherboards, 42–43
- external networking, function of, 52
- external transfer rate, of disk drives, 58

F

- /f switch, for forcibly dismount a virtual disk, 113
- Failover, defined for failover clusters, 311
- failover cluster architecture, introduction to, 312–314
- failover clustering, introduction to, 310–315
- failover cluster products, for configuring failover clusters, 315
- failover clusters
 - avoiding split brain problems in, 319
 - choosing the right model for, 316–319
 - configuring hardware, 319–320
 - dedicated secondary node (active-passive) model, 318
 - dedicated secondary node (active-passive) split model for, 318
 - defining essential terms, 311
 - as enterprise virtualization elements, 472–473
 - example of a simple two node, 310
 - fundamental hardware requirements for, 315
 - high availability with static load balancing (active-active) model for, 318–319
 - hybrid model, 319
 - implementing, 309–345
 - planning for, 315–320
 - points of failure you can eliminate by using, 309
 - resource and group configuration for MCS, 323–325
 - setting up on Linux operating systems, 331–345
 - setting up Red Hat Cluster Suite, 331–333
 - shared storage choices for, 472–473
 - single-node virtual mail server cluster, 317
 - typical reasons for implementing, 309
 - working with clustering solution products for, 314–315
- failover domains, Red Hat Cluster Suite support for, 333
- Failover Manager, in Microsoft Cluster Service (MCS), 322
- FalconStor IPStor Virtual Table Library (VTL), library emulation provided by, 468–469
- Fallback, defined for failover clusters, 311
- fault tolerance, adding to SANs, 466
- fault-monitoring and fault tolerance, performing, 221
- FCP. *See* Fibre Channel Protocol (FCP)
- Fiber vs. fibre, 423
- fiber-optic cable modes, 423
- fibre vs. fiber, 423
- Fibre Channel
 - advantages of for connecting to external storage, 421
 - vs. iSCSI for virtual ISs, 477–478
 - reasons not to use, 422
- Fibre Channel bridge/router
 - function of, 426
 - web site address for information about, 427
- Fibre Channel cables, introduction to, 423–424
- Fibre Channel hardware, configuring, 429
- Fibre Channel hardware devices, introduction to, 424–428
- Fibre Channel HBAs. *See* HBA (Host Bus Adapter)
- Fibre Channel hubs, 426
- Fibre Channel Industry Association, web site address for, 424
- Fibre Channel networks
 - arbitrated-loop topology, 427–428
 - point-to-point topology, 427
 - switched-fabric topology, 427
- Fibre Channel network topologies, 427–428
- Fibre Channel over IP (FCIP), for bridging two Fibre Channel SANs over an IP network, 430
- Fibre Channel Protocol (FCP), used by storage area networks (SANs), 63
- Fibre Channel SAN, hardware devices used in, 424–428
- Fibre Channel storage solution, for failover clusters, 473
- Fibre Channel switches, for connecting devices on Fibre Channel networks, 425–426

- Fibre Channel switches and routers,
 - methods for connecting and configuring, 429
 - File Name option, for renaming a virtual machine, 106
 - File Replication Services (FRS). *See* FRS (File Replication Services)
 - file system environment
 - changes when global namespace introduced, 498
 - with a global namespace, 497–498
 - file systems
 - implementation of in file/record layer, 451
 - tweaking to improve performance, 83
 - file transfers, increasing I/O performance for large files, 83
 - file-based storage architectures, combining with block-based storage architectures, 456–457
 - file/record layer
 - common implementations seen at this level, 451
 - function of in SNIA Shared Storage Model, 451
 - places functions can be implemented, 452
 - files, renaming using the Manage Files option in the MUI, 201
 - Filtering Mode, setting for NLB clusters, 367
 - fixed and preallocated disks, 13
 - fixed disks vs. dynamic disks for VMs, 89–90
 - flat-file backups
 - performing, 239–259
 - restoring for VMs, 261
 - running VMware Workstation, 241–247
 - running VMware Workstation on Windows, 242–245
 - running with Virtual Server 2005, 252–257
 - script for backing up a VM on a GSX Server host, 249
 - steps for performing, 240
 - variables for running on Virtual PC 2004, 251
 - Floppy Disk option, default setting for VMs, 107
 - floppy drive
 - creating virtual, 99, 182
 - *.filename value is mount point for device, 208
 - forced perfect termination. *See* FPT (forced perfect termination)
 - foreign cells
 - defined, 289
 - making visible in a Linux VM local cell's file tree, 306–307
 - FPT (forced perfect termination), for single-ended (SE) SCSI devices, 420
 - Free Software Foundation, definition, 4
 - FRS (File Replication Services)
 - configuring for DFS, 270
 - used by domain-based DFS root, 266
 - used for domain-based DFS namespaces, 270
 - full system recovery, performing, 259–261
 - fully-qualified domain name (FQDN)
 - finding configuration files relating to, 162–163
 - finding for server, 142
 - using, 159
- ## G
- gaming emulators, 4
 - GBIC (Gigabit Interface Converter), built-in adapter for Fibre Channel HBAs, 424–425
 - GBICs (Gigabit Interface Converter),
 - advantages of using for SAN topology, 425
 - generic SCSI devices
 - adding to your Linux guest VMs, 101
 - configuring, 184–185
 - eliminating reasons for failure, 54
 - importance of to VMs, 25
 - Linux configuration options, 101
 - for Linux VMs, 55–56
 - supporting, 53–56
 - GFI Network Server Monitor, web site
 - address, 489
 - Gigabit Interface Converter (GBIC). *See* GBIC (Gigabit Interface Converter)
 - global namespace
 - architecture example, 499
 - defined, 495
 - file system management with, 497
 - how it works, 498
 - for lifecycle management of reference data, 501
 - managing user home directories with, 500
 - managing Web site data with, 501
 - the new paradigm in distributed data management, 495–504
 - for optimizing exchange usage, 501
 - requirements for creating and deploying, 499
 - for sharing data across multiple departments, 500
 - using in the enterprise, 500–501
 - global namespace requirements, how StorageX and VFM meet them, 499
 - “Global Namespace: The New Paradigm in Distributed Data Management”, by NuView (2004), 495–504

- Global Update Manager, in Microsoft Cluster Service (MCS), 322
- GNU Privacy Guard (GnuPG), using to import the Samba public key, 273
- graph option, using with `vmkusage1` command, 216
- group assignments
 - making final for MCS, 325
 - making primary for MCS, 325
- groups
 - adding resources to, 329–330
 - moving to another node, 329
 - server resources place into, 320–321
- Groups folder, function of in Cluster Administrator, 329
- GSX Server
 - Add Hardware Wizard for installing virtual devices, 180
 - adding DVD/CD-ROM drives to guest VMs, 182
 - adding parallel port to guest VMs, 184
 - configuring a USB controller for, 185
 - copying and moving VMs to other hosts, 199–200
 - counters needed to identify system bottlenecks, 219–220
 - creating a serial port for, 183
 - hard disk options, 181
 - limitations of virtual switches, 23
 - physical hard drive creation for, 181
 - removing/adding VM hardware devices, 180
 - renaming guest VMs, 198
 - support for Broadcom-based network adapters, 50
- GSX Server for Windows. *See also* VMware GSX Server for Windows
 - installing, 143–145
 - silently installing, 144
- GSX Server VM startup, scripting, 492
- guest disk sizing
 - IDE device controller limitations in, 62–63
 - preallocating space upfront, 90
 - using virtual SCSI disks in, 62–63
- guest operating systems
 - Microsoft virtual hardware specifications for, 6
 - requirements for installing in ESX Server, 156
- guest VMs
 - adding USB support, 99
 - copying and moving in VMware Workstation, 119–120
 - OS licenses needed for, 70
 - preliminary tasks before moving, 119
- H**
 - ha.cf file configuration, for Linux H-A clusters, 337–341
 - handling priority, setting for NLB clusters, 367
 - hard disk. *See also* hard drive
 - adding to VMs using Add Hardware Wizard, 97–98
 - Hard Disk (1-3) option, for enabling or disabling VM hard disks, 106
 - Hard Disks Properties option, for Microsoft Virtual Server VMs, 195
 - hard drive
 - implementation situations table, 59
 - specifications for physical, 57–59
 - for virtual machines, 10–11
 - hardware. *See* computer components (hardware)
 - hardware compatibility lists (HCLs)
 - importance of when choosing VM hardware, 5–8
 - purpose of, 7
 - hardware emulators, function of, 2
 - hardware RAID
 - implementing, 442–443
 - vs. software RAID, 59–60
 - Hardware Type selection dialog box, for adding virtual hardware to VMs, 97
 - hardware updates, downloading driver and adapter updates, 79
 - hardware virtualization, benefits of, 1
 - haresources file configuration, for Linux H-A clusters, 341–345
 - hb_check.pl script, for monitoring the heartbeat of a VM, 221
 - HBA (Host Bus Adapter)
 - for connecting a server to a Fibre Channel network, 424–425
 - web site address for information about, 425
 - HCLs. *See* hardware compatibility lists (HCLs)
 - heartbeat
 - defined, 311
 - defined for failover clusters, 311
 - Heartbeat, as key to operating Linux-HA clusters, 334
 - heartbeat service, starting for Linux H-A clusters, 345
 - Help window, closing in VMware ESX Server, 155
 - hierarchical storage management (HSM)
 - common reasons for implementing, 467
 - introduction to, 466–467
 - major product vendors for, 467

- home cell. *See* local cell (or home cell)
 - host, defined, 16
 - Host Bus Adapter (HBA). *See* HBA (Host Bus Adapter)
 - host computer supported by SNIA Shared Storage Model, 449
 - host disk sizing, determining needs for your environment, 61–62
 - host storage, prioritizing, 56
 - host systems, running backup agents on, 230
 - host-based architecture, understanding in SNIA Shared Storage Model, 461–462
 - host-based virtualization, used for RAID configurations, 461
 - host-only networking
 - advantages of, 46
 - supported by Microsoft and VMware, 20–21
 - of virtual machines, 52
 - HP's Virtual TapeServer, web site address for information about, 468
 - HTTP port rule, configuring, 368
 - hubs. *See* Ethernet hubs; Fibre Channel hubs
 - HVD SCSI buses. *See* differential (HVD) SCSI buses
 - HVD termination, 421
 - hybrid model, failover clusters, 319
 - hybrid networking, supported by Microsoft and VMware, 20–21
 - hyper option, using with vmkusaget1 command, 216
 - hyperthreading, disabling for Microsoft Virtual Server hosts, 140
-
- IBM TotalStorage SAN Volume Controller, features of, 464–465
 - IDE devices, checking that controllers use direct memory access, 72
 - IDE drives, performance of vs. SCSI drives, 58
 - IDE RW decimal values, compared to Octal values and disk sizes in GBs, 207
 - IDE virtual disk advanced options, 91
 - iFCP, function of, 430
 - IGMP Multicast setting, using to correct switch flooding, 356
 - IIS (Internet Information Services), installing before Virtual Server, 141
 - IIS 6.0 Resource Kit Tools, self-signing certificate utility included in, 142
 - in-band network-based virtualization, function of, 464–465
 - independent disks, persistent and nonpersistent, 14–15
 - initiators. *See* iSCSI initiators
 - installing
 - and deploying VMs on enterprise servers, 139–171
 - IIS before Virtual Server, 141
 - Kerberos, 275–278
 - Microsoft Virtual Server on Windows 2003, 140–143
 - RPM for GSX Server for Linux, 147–149
 - TAR for GSX Server for Linux, 149–150
 - Virtual Machine Additions, 108–109
 - Virtual PC 2004 SP1, 71
 - Virtual PC on a Windows XP host computer, 69–71
 - VM applications on desktops, 69–84
 - VMware ESX Server, 156–160
 - VMware GSX Server for Linux, 146–150
 - VMware GSX Server for Windows, 143–145
 - VMware Management Interface, 150–151
 - VMware Tools, 92–101
 - VMware Virtual Machine Console, 152–153
 - VMware Workstation for Linux, 74–78
 - VMware Workstation for Linux RPM, 75–76
 - VMware Workstation for Linux TAR, 76–77
 - VMware Workstation for Windows, 71–74
 - Windows Server 2003 Cluster Service, 326–330
 - Instant Provisioning Deployment Wizard, for quickly deploying new servers, 221
 - instsrv.exe tool, for running VMs as services, 123
 - integrated devices, considerations for when researching motherboards, 43–44
 - integrated NICs, desirability of on motherboards, 43
 - Intel, web site for CPUs and required bus speeds information, 41
 - intercabinet configuration, defined, 423
 - interconnection network component, supported by SNIA Shared Storage Model, 448
 - internal transfer rate, of disk drives, 58
 - International Standard Organization Open System Interconnect (ISO OSI) model, 17
 - Internet Protocol (TCP/IP) Properties dialog box, adding NLB cluster nodes IP address in, 363–364
 - Internet SCSI (iSCSI). *See also* iSCSI Windows Server clusters
 - used by storage area networks (SANs), 63
 - intracabinet configuration, defined, 423
 - I/O Adapter Types selection screen, for choosing adapter type for VMs, 175

I/O devices, for virtual machines, 25–26

IP address

- controlled by Cluster Service, 320
- defined, 16

IPv4 classes, table of, 16

iSCSI. *See also* Internet SCSI (iSCSI)

- vs. Fibre Channel for virtual ISs, 477–478
- securing, 431
- web site address for information about, 431

iSCSI (Internet SCSI), introduction to, 430–431

iSCSI architecture

- example of, 431
- understanding, 430–431

iSCSI Initiator Properties window, properly configured iSCSI target disks in, 403

iSCSI initiators

- in iSCSI architecture, 430–431
- iSCSI target methods to authenticate, 431

iSCSI initiator software

- configuring to connect nodes to shared storage, 401–403
- steps for installing, 401
- web site address, 400

iSCSI shared storage solution, for failover clusters, 473

iSCSI target

- code for ping, 393
- configuring, 394
- installing Active Directory on, 394
- installing the StarWind software, 394–395
- methods to authenticate iSCSI initiators, 431
- preparing the cluster nodes as clients, 400–404

iSCSI target portal, adding, 401

iSCSI VM cluster, 392

iSCSI Windows Server clusters

- configuring drive letters for shared storage, 404
- network settings table, 393
- preparing for the VM configuration, 392–393
- setting up, 391–404

ISO image-making software, downloading a trial copy of, 89, 175

ISO images

- mounting, 177–178
- mounting with Microsoft OSs, 182

ISO OSI model, 17

K

Kerberos

- adding principals for network hosts after installing, 277
- adding remote access, 276

- configuring administrative access, 276
- creating the realms database, 276
- installing, 275–278
- starting the daemons, 276
- test commands table, 277
- verifying Samba compatibility with, 275

Kerberos 5 server, configuring a XP user and host machines to use, 277–278

“Kerberos Interoperability”, web site address for, 278

“Kerberos V5 Installation Guide”, web site address for, 276

kernel RPM, downloading and installing upgrade, 410

keyboard shortcuts

- and administration for VMware Workstation CLI, 129–131
- creating in your VM service shortcut folder, 127
- included with VMware Workstation, 129–131
- for Virtual PC, 134

kill, sources for downloading, 82

KITS (keep IT simple), concept for VM networks, 48

L

LAN-based backups, configuration on a private host-only network, 225–227

LAN-free backups, performing for SANs, 432

LANs, backing up VMs over, 225–227

laptop computers, using as virtual machine hosts, 36

layers, in SNIA Shared Storage Model, 450

Lee, James

- Beginning Perl*, Second Edition (Apress, 2004) by, 192

legacy devices, configuring, 183–184

legacy operating systems, running in a virtual environment, 26–27

library emulation, provided by FalconStor IPStor Virtual Table Library (VTL), 468–469

licensing

- verifying for load-balanced clusters, 351
- verifying for MCS configurations, 324

Lightweight Directory Access Protocol (LDAP) (UDP/TCP 389), required to support DFS, 265

link referral, defined for DFS, 264

link target, defined for DFS, 264

linked and physical disks, 13

Linux applications, Red Hat Cluster Suite support for all major, 333

Linux computer system accounts, creating with parts that Samba requires, 283

Linux DFS, implementing, 271–284

- Linux file systems, backing up, 234–239
- Linux guest VMs, configuring XWindows for, 105–106
- Linux H-A clusters. *See also* Linux High Availability (Linux H-A) clusters
 - example of two node, 409
 - installing the remaining configuration and management packages, 335–336
 - starting the heartbeat service, 345
 - two-node failover cluster, 334
 - UltraMonkey implementation of, 334–336
 - upgrading and installing the kernel, 335
- Linux H-A failover clusters, web site address for information about, 409
- Linux High Availability (Linux H-A) clusters, using, 333–345
- Linux High Availability Project, web site address for, 333
- Linux iSCSI project, web site address for iSCSI information, 431
- Linux operating system, case sensitivity of, 161
- Linux OSs
 - table of survival commands, 164
 - Vi editor included with most, 162–163
- Linux reference book, *Beginning SUSE Linux: From Novice to Professional* (Apress, 2005), 154
- Linux server, configuring, 274
- Linux survival commands, using at the Service Console, 163–164
- Linux Virtual Server (LVS) clusters. *See also* LVS clusters
 - building, 374–377
- Linux VM clusters, building, 409–411
- Linux VMs
 - backing up on a Windows host, 245
 - backing up with tar, 236–238
 - building with Microsoft Virtual PC, 105–109
 - configuring basic cell security, 303
 - directories created during installation, 303
 - dump command for performing backups, 234–236
 - generic SCSI devices for, 55–56
 - installing VMware Tools for on guest VMs, 95
 - monitoring performance of, 135–136
 - restoring a tar backup file, 237
 - restoring data backed up with dump command, 235–236
 - scripting a flat-file backup on a Windows host, 243–245
 - steps for installing AFS for, 302–307
 - using the vmware command on a host, 131
- load balancing with clustering and network adapter teaming, 49
- load weight, setting for NLB clusters, 367
- load-balanced clusters
 - analyzing risks, 352
 - creating, 347–377
 - estimating server capacity, 352–353
 - example of a typical three-node cluster, 348
 - fault tolerance provided by for virtual ISs, 473–474
 - implementation options for, 473–474
 - key phases for planning, 351
 - planning for, 351–353
 - selecting applications to run on, 351
 - typical implementations of, 347
 - verifying licensing for, 351
- local cell (or home cell), defined, 289
- Log Manager, in Microsoft Cluster Service (MCS), 322
- logical storage resource, supported by SNIA Shared Storage Model, 449
- logical units (LUs), support for in SCSI storage devices, 453
- low-voltage differential (LVD) SCSI buses, function of, 416
- LPT ports for Virtual Server VMs, 197
- LPT1 option for allowing VM connection to parallel ports, 107
- LSI Logic driver, downloading for VMware guest VMs, 89, 175
- LUN
 - defined, 64
 - management, 64–66
 - masking, 66
 - vs. SCSI ID, 415
- LVD SCSI buses. *See* low-voltage differential (LVD) SCSI buses
- LVD termination, 421
- LVS architecture
 - advantages of virtual server via IP tunneling, 376–377
 - understanding, 375–377
 - virtual server via direct routing, 377
 - virtual server via NAT, 376
- LVS clusters
 - example of fault-tolerant four node, 375
 - example of four node with one load balancer, 374
 - implementing, 377
 - web site address for information about, 409
- LVS Project, Linux load-balanced clusters designed around, 374

M

- MAC address. *See* media access control (MAC) address
- MAC addresses, changing for Ethernet adapters in *.vmc configuration files, 118
- MagicISO, web site address for downloading, 89, 175
- magnetic disk, writing to, 468
- mainframe virtual machine (MVM), function of, 3
- make command, creating/installing Samba binaries with, 278
- makecert.exe
 - steps for placing in an easy-to-find location, 142–143
 - web site address for downloading, 142
- MAME. *See also* Multiple Arcade Machine Emulator (MAME)
 - web site address, 4
- Management User Interface (MUI). *See also* VMware Management Interface
 - working with in VMware ESX Server, 165
- Map Drive Letter screen, 301
- master key distribution center (KDC), configuring, 275–276
- McDATA, web site address, 426
- MCS. *See* Microsoft Cluster Service (MCS)
- media access control (MAC) address, for network interface card, 22
- media server, configuring static routes on, 226
- Membership Manager in Microsoft Cluster Service (MCS), 322
- memory allocation for VMware virtual machines, 88
- Memory option
 - for increasing or decreasing available RAM, 106
 - for Microsoft Virtual Server VMs, 195
- Memory selection dialog box, selecting RAM requirements for Linux VM, 105
- memory types, performance data for, 41
- mesh networks for achieving redundant switching and routing, 48
- micron, 423
- Microsoft
 - acquisition of Virtual PC from Correctix, 26–27
 - BIOS packaged with, 24–25
 - dynamic expanding disks, 12–13
 - fixed disks, 13
 - linked disks, 13
 - supported I/O devices for VMs, 25–26
 - undo disk, 13–14
 - virtual disks, 11–16
 - virtual network types supported by, 20–21
 - VM network interface card specifications, 20
- Microsoft backup domain controller, inability of Samba to communicate with, 281
- Microsoft Cluster Service (MCS)
 - assignment of physical disk resources, 324
 - Checkpoint Manager in, 321
 - Cluster Network Driver in, 321
 - Communications Manager in, 321
 - Configuration Database Manager in, 321
 - documenting dependencies for each resource, 324
 - examples of cluster resources, 320–321
 - listing all nonapplication resources where failover is desired, 324
 - looking under the hood, 320–323
 - making primary group assignments, 325
 - Node Manager in, 321
 - other components in, 322
 - resource dynamic link libraries (DLLs) in, 322
 - resource monitor in, 322
 - setting up, 320–330
- Microsoft Installer, msixexec.exe, 213
- Microsoft licensing brief, web site address for downloading, 31
- Microsoft Loopback adapter, function of, 53
- Microsoft operating systems, and RAM configurations, 10
- Microsoft Performance console, removing counters from or adding counters to, 136
- Microsoft Remote Storage Service (RSS), web site address, 467
- Microsoft System State. *See* System State
- Microsoft virtual hardware specifications, for guest operating systems, 6
- Microsoft Virtual Machine Additions. *See* Virtual Machine Additions
- Microsoft Virtual Machine clusters. *See also* Microsoft Windows Virtual Machine clusters; VM clusters
- Microsoft Virtual PC. *See also* Virtual PC
 - building a Linux VM with, 105–109
 - building Windows VMs with, 101–104
 - default VM configuration for VM, 102
 - deploying virtual machines (VMs) with, 69–71
 - minimum supported host specifications, 27
 - similarities to VMware Workstation, 28
 - using the New Virtual Machine Wizard, 101–104
- Microsoft Virtual PC for Mac, function of, 2

- Microsoft Virtual Server 2005. *See also* Virtual Server 2005
 - adding a virtual SCSI adapter to VMs, 389
 - adding or removing network adapters for VMs, 196
 - adding scripts in the Scripts Properties dialog box, 196
 - best-practice minimum hardware configuration, 140
 - COM ports for the VMs, 197
 - configuring clusters on, 388–391
 - configuring SCSI adapters for VMs, 196
 - configuring SSL for, 141
 - contents of General properties page, 194
 - copying and moving VMs to other hosts, 205–206
 - deploying using group policies, 213
 - disabling autorun before install from CD-ROM, 140
 - function of Virtual Machine Additions package, 194–195
 - guest operating systems supported by, 29
 - Hard Disks Properties option for VMs, 195
 - importance of having VM backups before moving, 205
 - importance of installing IIS before installing, 141
 - LPT ports for VMs, 197
 - minimum specification for guest operating systems, 29
 - minimum supported host specifications, 30
 - performing command-line management for, 213
 - steps for installing, 140–141
 - steps for renaming a VM, 204–205
 - using Memory option add/reduce available RAM, 195
 - virtual floppy drives for the VMs, 196
 - VM initial configuration, 388
 - web site address for checking for patches and updates, 143
- Microsoft Virtual Server 2005 Enterprise. *See* Virtual Server 2005 Enterprise
- Microsoft Virtual Server Toolkit (MVST), function of, 30
- Microsoft Virtual Server VMs
 - building, 193–197
 - steps for creating, 193–194
- Microsoft VM clusters, building, 379–409
- Microsoft VMs and SANs, 63
- Microsoft Windows 2003 Resource Kit
 - srsvany.exe and instrsv.exe tools in, 123
 - web site address for downloading tools, 123
- Microsoft Windows Update, using to ensure your computer is up-to-date, 79
- Microsoft Windows Virtual Machine clusters. *See also* Microsoft VM clusters; VM clusters
 - building, 379–411
- Microsoft's HCL, web site address for, 8
- Microsoft's Script Center Script Repository, web site address for, 196
- Mirrordir tool, web site address for, 333
- mirrored stripes. *See* RAID 0+1 arrays
- mkinitrd, using RPM for installing latest package, 334, 410
- MMC, changing default port GSX Server uses for, 145
- /mnt/iso directory
 - listing contents of, 177
- "Monitor Disk Space on Multiple SQL Servers" article, web site address, 220
- monthly option, using with vmkusaget1 command, 216
- Morrill, Daniel L.
 - Tuning and Customizing a Linux System* (Apress, 2002) by, 74
- motherboard
 - bus types, 42–43
 - features that indicate good quality, 45
 - importance of board form factor when selecting, 44
 - importance of controller chipset when researching, 40–41
 - importance of CPU speed and quantity when researching, 40
 - importance of easy CMOS battery replacement, 45
 - importance of overall quality of, 44–45
 - memory requirements for host and guest VMs, 41–42
 - selecting for your virtual machine host, 38–45
 - things to look at when selecting, 39
- mount point, function of, 290
- mouse, releasing focus from guest VM to host system, 94
- Mouse Pointer Integration option, available after installation of Virtual Machine Additions, 107
- MUI. *See* VMware Management Interface
- MUI and SSL, understanding, 165–166
- multicast mode
 - disadvantages of, 355
 - vs. unicast mode, 354
- multicast NLB cluster, using a hub to connect to the network, 355–356
- multicast NLB cluster node, performing the initial configuration for, 361–364

multimode fiber-optic cable, two forms of, 423

multipathing of SANs, 466

Multiple Arcade Machine Emulator (MAME), open-source virtualization application, 4

MVM. *See also* mainframe virtual machine (MVM)

■ N

namespace root, publishing in Active Directory, 269

naming conventions, for naming VMware VMs, 86–87

NAS head (file server controller), function of in combined block and file/record layers, 457

NAS server, function of in combined block and file/record layers, 457

NAT (Network Address Translation), 16

function of, 18–19

NAT networking

- disadvantages of, 46–47
- supported by Microsoft and VMware, 20–21

NATing process, example of, 19

NBL clusters, example of three-node unicast cluster, 365

net time command, for synchronizing clocks on Windows 9x and NT systems, 272

NetBIOS (UDP 138, TCP 139), required to support DFS, 265

network adapter, offloading processor tasks to, 83

network adapters, need for two for ESX Server, 157

network adapter teaming

- defined, 49
- function of, 49–51

Network Address Translation (NAT). *See* NAT (Network Address Translation)

Network Appliance, web site address for virtual tape server information, 468

network configuration, in ESX Server installation process, 168–169

Network Configuration screen in ESX Server installation process, 159

Network Instruments' Observer

- "Case Study: Jack in the Box," by Network Instruments (2004), 523–526
- network analyzer, 271

network interface card specifications

- for virtual machines, 20
- for VMs, 22

network load balancing cluster, function of, 49

Network Load Balancing Manager

- adding a node to an existing cluster in, 370
- changing configuration parameters of an existing cluster in, 370
- configuring NLB VM clusters with, 407–409
- creating a new cluster in, 371
- managing NLB clusters with, 369–371
- monitoring status of an existing cluster in, 369
- removing a node from a cluster in, 371
- using, 358–359

Network Load Balancing Properties dialog box

- configuring a multicast NLB cluster node in, 361
- setting the NLB cluster node priority in, 362

network name, controlled by Cluster Service, 320

Network Time Protocol. *See* NTP (Network Time Protocol)

Network Type Selection screen, for choosing networking characteristics for VMs, 175

network-based architecture, understanding in SNIA Shared Storage Model, 463–465

networking, introduction to, 16–19

Networking option, for setting network adapters guest VMs can use, 107

networking virtual machines (VMs), introduction to, 20–21

networks, storing and executing virtual disks over, 12

New Link dialog box, for creating new DFS links, 269

New Resource dialog box, for adding resources to a group, 330

New Root Wizard

- choosing DFS root name in, 267
- completion summary screen, 268
- selecting DFS root type in, 267

New Server Cluster Wizard, installing Windows Server 2003 Cluster Service with, 405

New Virtual Machine Wizard

- building a Windows VM with, 101–104
- completing, 104
- launching for VMware Workstation, 85
- Specify Disk Capacity screen, 89

NIC (network interface cards). *See* network interface card specifications

NLB cluster node, steps for completing the configuration of, 365

- NLB clusters
 - adding a node to an existing, 370
 - adding the IP address to the network
 - interfaces IP addresses, 363–364
 - best practices for implementations, 359
 - building, 353–374
 - changing configuration parameters of an existing, 370
 - configuring and managing, 360–374
 - configuring port rules for, 366–369
 - configuring with the Network Load Balancing Manager, 407–409
 - creating a new in Network Load Balancing Manager, 371
 - function of priority numbers in, 357
 - managing with the `wlbs.exe` command, 371–373
 - monitoring status of an existing, 369
 - performing initial configuration for unicast nodes, 364–365
 - performing the initial configuration for a multicast node, 361–364
 - performing the initial node configuration, 360–361
 - remotely administering nodes from the command line, 358
 - removing a node from, 371
 - setting node priority numbers in, 357
 - setting port rules for, 358
 - testing, 366
 - using the registry to modify parameters, 373–374
 - NLB network service
 - enabling, 353–354
 - steps for installing and enabling, 360
 - NLB registry
 - modifying to improve NLB cluster performance, 373–374
 - table of common values, 373
 - NLB VM clusters, setting up, 406–409
 - noauth option, disabling authentication
 - checking with, 302–303
 - Node Manager, in Microsoft Cluster Service (MCS), 321
 - Nodes, defined, 310
 - nograph option, using with `vmkusaget1` command, 216
 - non-agent-based backups
 - most popular types, 231
 - performing, 230–239
 - using Windows Backup utility for, 231–234
 - nonpersistent independent disks, function of, 14–15
 - Norton Ghost, use-imaging software, 15
 - Norton PartitionMagic, web site address for information about, 98
 - N-plus-1 architecture, function of, 313–314
 - N-tier clustering, introduction to, 313–314
 - N-to-1 architecture
 - example of three-node cluster, 314
 - function of, 313
 - NTP (Network Time Protocol), configuring for Linux guest VM, 106
 - NTP daemon, disabling, 272
 - `ntpdate` command, for synchronizing local clock with W32Time service, 272
 - NuView (2004)
 - “Global Namespace: The New Paradigm in Distributed Data Management” by, 495–504
 - NuView StorageX, for implementing out-of-band virtualization, 465
 - NuView StorageX Global Namespace
 - complete global namespace manageability, 503
 - data directory services, 503
 - enterprise DFS implementation using, 480
 - nonproprietary file system, 503
 - simplicity of installation and usage, 504
 - as standards-based program, 503–504
 - unique features of, 502–504
 - unlimited scalability of, 504
 - use of to deliver complete network data management platform, 502
- ## O
- Object Manager, in Microsoft Cluster Service (MCS), 322
 - Obtain New AFS Tokens screen, for obtaining new AFS tokens, 301
 - “Official Samba-3 HOWTO and Reference Guide”, web site address for, 278
 - online snapshots
 - reasons for taking, 257
 - setting options for, 258
 - steps for recovering, 258
 - steps to follow for taking, 257
 - Open GFS home page, web site address, 333
 - Open Source Initiative, definition, 4
 - OpenAFS. *See also* AFS (Andrew File System)
 - Choose Components dialog box, 293
 - compiling into Samba, 288
 - component options, 293
 - configuring for your Windows system server, 294–301
 - in public domain for development and maintenance, 288
 - steps providing basic overview of Windows install, 292–293
 - web site address for downloading, 292
 - open-source virtual machines, defined, 4

- OpenSSH ssh server for Windows, web site address, 246
- OpenSSH ssh server for Windows Services for Unix, web site address, 246
- Operating System dialog box
 - in New Virtual Machine Wizard, 103
 - selection for Linux VM in Virtual PC, 105
- operating system virtual machines (OSVMs), function of, 3
- out-of-band network-based virtualization, common implementation of, 465
- P**
 - p option, for VMware Workstation CLI, 129
 - /p option, with vmware-mount for finding what partitions are available, 113
 - P2V Assistant, function of, 33
 - package program, managing client configuration files with, 292
 - Paging Executive, disabling to improve performance, 82
 - paging file, steps for optimizing to improve performance, 83
 - parallel port
 - adding to GSX Server guest VM, 184
 - adding to your Linux guest VMs, 100
 - enabling support for ESX Server, 184
 - parallel processing vs. parallel virtual machines, 3
 - parallel virtual machines (PVMs), function of, 3
 - passive termination, for single-ended (SE) SCSI devices, 420
 - passwords, creating strong, 159
 - PC Additions for Linux OSs, dropped by Microsoft after Virtual PC purchase, 105
 - PCI-X bus, speed of, 42–43
 - PenguinNet shareware
 - shutdown profile Send Commands settings, 244
 - shutdown profile settings, 244
 - web site address for downloading, 243
 - performance counters
 - using in Windows XP, 136
 - VMware, 135–136
 - performance tuning. *See* tuning tips
 - Perl (Practical Extraction and Reporting Language), created by Larry Wall in 1987, 191
 - Perl scripts, executing by calling at the CLI, 192
 - Permissions, setting on Perl scripts, 192
 - persistent binding, available for ESX Server VMs, 66
 - persistent independent disks, function of, 14–15
 - physical disks. *See* physical hard drive
 - controlled by Cluster Service, 320
 - physical hard drive
 - creation for ESX Server, 181
 - creation for GSX Server, 181
 - repartitioning, 98
 - specifications for, 57–59
 - using for VMs, 97–98
 - physical libraries, dividing, 468
 - physical storage resource, supported by SNIA Shared Storage Model, 449
 - ping command, for testing network connectivity between two systems, 226
 - .pl extension for Perl scripts, 191
 - plain disk files, using to build Windows VM cluster, 381
 - PlateSpin Power P2V
 - comparing other methods of converting between physical and VM infrastructure with, 507–510
 - enhancing data center flexibility through, 506
 - monitoring the conversion jobs, 510
 - physical-to-virtual migration comparison, 507–510
 - summary of features, 506
 - point-to-point topology, for Fibre Channel networks, 427
 - policy-based data management, developed for performing backups, 466
 - port range, setting for NLB clusters, 366
 - port rules
 - configuring for NLB clusters, 366–369
 - setting for NLB clusters, 358
 - setting to be aware of, 366–367
 - steps for configuring for NLB clusters, 368–369
 - pound sign (#), using for Perl script comment lines, 192
 - power-on-self-test (POST), as part of computer boot process, 25
 - power-up or power-down characteristics, determining for a guest VM, 174
 - presentation environments, determining disk size needs for, 61
 - print spooler
 - controlled by Cluster Service, 320
 - sample resource dependence tree, 325
 - private disk, defined for failover clusters, 311
 - private network vs. public network, 19, 45–47
 - production environments, determining disk size needs for, 62
 - production server virtualization, 513–514
 - production VMs, deploying and managing on enterprise servers, 173–222
 - Project RC5, web site address, 3

Protocols, setting for NLB clusters, 366
 public network vs. private network, 19, 45–47
 pulist, web site address for downloading, 82
 Purple Screen of Death, in ESX Server, 160
 PuTTY, Telnet and ssh client for Windows
 OSs, 157
 PVMs. *See also* parallel virtual machines
 (PVMs)
 examples of, 3

Q

Q key, to advance to next prompt if GSX
 Server for Linux hangs, 147
 -q option, for VMware Workstation CLI, 129
 Qlogic, web site address, 425
 quorum resource, defined for failover
 clusters, 311

R

RAID (Redundant Array of Inexpensive
 Disks)
 configured hard drives for VM hosts, 36
 hardware vs. software, 59–60
 implementing, 442–443
 introduction to common levels, 435–442
 the root of storage virtualization, 435–443
 for virtual machines, 59–61
 RAID 0
 function of, 60, 436–437
 reasons to avoid, 437
 RAID 0+1 arrays
 configuring, 440
 example of, 439–440
 vs. RAID 1+0 arrays, 440
 RAID 1 (disk mirroring)
 disadvantages of, 438
 forms available, 437–438
 function of, 60
 RAID 1+0 (RAID 10), function of, 440–441
 RAID 1+0 arrays vs. RAID 0+1 arrays, 440
 RAID 1 array, example of, 437
 RAID 5 (block-level striping with distributed
 parity)
 function of, 61, 438–439
 function of vs. RAID 5+0 arrays, 442
 vs. RAID 0, 438
 reasons to think twice about, 439
 RAID 5+0 arrays
 example of, 441
 function of vs. RAID 5, 442
 RAID types, benefits and drawbacks, 60–61
 RAM (computer memory), for virtual
 machines (VMs), 9–10
 RAM configurations, Microsoft operating
 systems and, 10
 Ready to Install the Program screen, for
 VMware Workstation for Windows, 72

real servers, defined, 310
 Recycle Bin, resetting size to reduce disk
 space needed, 80
 Red Hat Cluster Configuration tool, simple
 configuration and administration
 with, 332
 Red Hat Cluster Suite
 SCSI and Fibre Channel shared storage,
 331
 setting up, 331–333
 simple configuration and administration,
 332
 support for all major Linux applications,
 333
 support for failover domains, 333
 Red Hat Enterprise Advanced Server systems,
 deploying and configuring, 332
 Red Hat Enterprise Linux System
 Administration Guide, web site
 address to download, 239
 Red Hat failover clusters, use of a shared
 quorum drive to maintain updates,
 332–333
 Red Hat Linux, installation navigation keys,
 155
 Red Hat's HCL, web site address for, 8
 redo logs, using to share virtual disk base
 images, 209
 redundancy, as function of block
 aggregation, 454
 Redundant Array of Inexpensive Disks
 (RAID). *See* RAID (Redundant Array
 of Inexpensive Disks)
 -registervm option, to make new workstation
 appear in console, 132
 Registry, using to modify NLB cluster
 parameters, 373–374
 regroove option, using with vmkusagect1
 command, 216
 Remote Assistance, disabling, 80
 remote command-line access, gaining to ESX
 Server or GSX Server Linux host
 machine, 157
 Remote Console
 changing GSX Server's port number for,
 153–154
 configuring to use a specific port
 automatically, 154
 remote control, understanding for NLB
 cluster nodes, 358
 Remote Desktop Sharing, disabling, 80
 removable media, storing virtual disks on, 12
 REMOVE and ADDLOCAL options,
 qualifiers, 144
 renaming
 a Microsoft Virtual Server VM, 204–205
 VMware ESX Server VMs, 201–202

- reparse point of stub file, 466
- repartitioning physical hard drives, 98
- replication, using to increase AFS performance, 290
- resource and group configuration
 - creating a list of all server-based applications for MCS, 323
 - listing all nonapplication resources where failover is desired, 324
 - planning for MCS, 323–325
 - sorting the application list for MCS, 323–324
 - verifying licensing for MCS, 324
- resource dynamic link libraries (DLLs), in Microsoft Cluster Service (MCS), 322
- Resource Manager, in Microsoft Cluster Service (MCS), 322
- resource monitor, in Microsoft Cluster Service (MCS), 322
- Restore and Manage Media dialog box, in Windows Backup utility, 234
- Robocopy, replicating data for stand-alone DFS deployment with, 270
- Rocket Division Software
 - market focus, 523
 - mission and technologies, 522–523
 - overview of iSCSI target and initiator software, 519–523
 - software-based iSCSI concentrator by, 477
 - web site address for StarWind application, 394
- root, defined for DFS, 264
- root referral, defined for DFS, 264
- Root Share selection screen, creating a DFS directory to be shared, 268
- root targets
 - adding/removing, 268
 - defined for DFS, 264
- root user account, for complete administration of ESX Server, 159–160
- round-robin DNS
 - the beginning, 349–350
 - flaws in, 350
 - vs. load-balanced clustering, 349
- route add command, for configuring static routes on the media server, 226
- RPC (TCP 135), required to support DFS, 265
- RPM, installing for GSX Server for Linux, 147–149
- RPM database, querying to verify installation of RPM prerequisites, 271
- RPM distributions, packages needed for, 271
- Rsync tool, web site address for, 333
- “running on the metal”, defined, 1–2

S

- s NAME=VALUE option, for VMware Workstation CLI, 129
- Samba
 - adding client VMs to the domain, 283
 - adding to Active Directory, 285–286
 - checking if it is installed on your Linux system, 271
 - configuration file variables table, 282
 - configuring as a domain controller, 280–284
 - configuring prior to using, 279–284
 - creating a backup domain controller, 284
 - creating shared directory and configuring permissions, 279
 - creating symbolic links within the DFS root directory, 287
 - creating user account for root with legacy authentication, 279
 - editing the smb.conf file to enable DFS, 287
 - enumerating shares, share type, and comments, 279
 - installing additional programs for, 274
 - installing and configuring on your Linux server, 271–278
 - packages needed for RPM distributions, 271
 - setting up to act as a Kerberized domain controller, 271
 - using a Kerberos server for authentication, 284–285
 - verifying integrity, 273
 - web site address for downloading, 273
- Samba daemons, table of, 273
- Samba DFS shares
 - adding subdirectories in, 287
 - setting up, 286–288
- Samba primary domain controller, providing redundancy to with a BDC, 284
- Samba public key, importing into your system, 273
- Samba service account, creating, 283
- Samba Web Administration Tool (SWAT), for managing Samba, 280
- SAN hardware, elements to consider when planning SAN deployment, 478
- SAN integration, with virtual machine hosts, 474–475
- SAN LUN, maximum number supported, 64
- SAN switches, major vendors for, 464
- SANs. *See also* storage area networks (SANs)
 - extending with FCIP and iFCP, 430
 - performing server-free backups for, 433
 - performing serverless backups for, 434

- pros and cons of using for virtual ISs, 476–477
- using backup and recovery techniques, 432–434
 - using LAN-free backups, 432
- SATA RAID, 60
- Scheduled Tasks system tool, for scheduling
 - Windows batch scripts to run, 240
- Scripting API. *See* VMware's Scripting API
- SCSI
 - ID vs. LUN, 415
 - introduction to, 414–421
- SCSI adapter, creating with a shared SCSI bus, 389
- SCSI attached shared storage solution, for failover clusters, 473
- SCSI buses
 - cable selection for, 421
 - deciding where to terminate, 418–420
 - types and specifications of, 415
 - understanding termination for, 418–421
 - using, 415–417
- SCSI devices. *See also* generic SCSI devices
 - configuring generic, 184–185
 - maximum cable lengths table, 416
 - performance of vs. IDE drives, 58
 - reasons for not properly installing for
 - Windows guests, 54
 - symbols for, 417
 - troubleshooting installation of new, 54–55
 - using ohmmeter or multimeter to check type, 417
- SCSI drives. *See* generic SCSI devices; SCSI devices
- SCSI hard drive, external with terminator attached, 419
- SCSI ID vs. LUN, 415
- SCSI parallel interface, categories of, 416
- SCSI termination
 - differentiating between active and passive, 420
 - rules for, 420
 - understanding, 418–421
- SCSI terminology, 414
- SCSI virtual disk advanced options, 91
- SE SCSI devices. *See* single-ended (SE) SCSI devices
- search directory, adding to a path in Linux, 150
- search path, configuring in Linux, 150
- secure shell (ssh), for remote command-line access to Linux host machines, 157
- self-signed certificate, creating and installing, 142–143
- SelfSSL utility, included in IIS 6.0 Resource Kit Tools, 142
- semicolon (;), ending for all Perl statements, 192
- Send Commands dialog box, for PenguiNet, 244
- serial port
 - creating for a VM, 100
 - creating for GSX Server, 183
- serial port support, enabling for ESX Server, 183
- server capacity, estimating for load-balanced clusters, 352–353
- server clustering, Microsoft implementation of, 320–330
- server consolidation, enhancing VM infrastructure through automation, 505–519
 - “Server Consolidation and Beyond: Enhancing Virtual Machine Infrastructure Through Automation”
 - written by PlateSpin (2004), 519
- server migrations across geographic regions, 514
- server-free backups, performing for SANs, 433
- serverless backups, performing for SANs, 434
- servers
 - migration of across geographic regions, 514
 - using as virtual machine hosts, 36
- Service Console, managing ESX Server through, 157
- services subsystem, SNIA Shared Storage Model, 460–461
- Seti@Home project, web site address, 3
- Setup.exe command
 - for installing Virtual PC, 131–134
 - table of options for, 131
- shared cluster quorum drives, use of by Red Hat failover clusters, 332–333
- shared controller interrupts, VMkernel management of, 186
- Shared Folders option, for allowing guest to access data on the host, 108
- shared networking, function of, 52
- shared nothing architecture, of active-passive failover clusters, 312
- shared storage
 - maintaining a standby VM server with, 483–484
 - standby server configuration, 484
- shared storage configuration, considerations when planning, 316
- shared storage environments, security concerns, 459–460

Shared Storage Model: A Framework for Describing Storage Architectures (SNIA, 2003)

- about SNIA shared storage architecture, 443–444
- shared storage resources
 - configuring for VMware GSX Server, 385–386
 - configuring for VMware Workstation cluster, 382–384
 - providing security for, 459–460
- shbang line, as first line of Perl scripts, 191
- single-ended (SE) SCSI devices
 - function of, 416
 - termination types for, 420
- single-mode fiber-optic cable, 423
- single-node model,
 - advantages/disadvantages of for failover clusters, 316–317
- SLA violations, using virtual machines to prevent, 515–516
- SMB (UDP/TCP 445), required to support DFS, 265
- SMB/CIFS protocol, installing, 273–284
- SMBIOS. *See also* System Management BIOS (SMBIOS)
 - web site address for specification, 200
- SMP licensing, for running ESX Server guest VMs, 154
- Snapshot, in reference to virtual machines, 14–15
- SNIA Shared Storage Model
 - access control, 459–460
 - applying, 461–465
 - benefits of, 445
 - common access paths, 457
 - data caching in, 458
 - example of, 447
 - function of block layer in, 453–456
 - introduction to, 443–461
 - the layering scheme of, 450
 - main components within its scope, 448
 - a note on the graphical conventions used in, 445
 - services subsystem, 460–461
 - storage system components, 448–449
 - understanding host-based architecture, 461–462
 - why to have a model for shared storage, 444
- software RAID
 - vs. hardware RAID, 59–60
 - implementing, 443
- sound adapters
 - adding to your GSX Server guest VMs, 183
 - adding to your Linux guest VMs, 99
- Sound option, enabled by default, 107

- SourceForge.net, Bochs project supported by, 4
- space management, as function of block aggregation, 453
- Specify Disk Capacity screen, for specifying maximum space used by guest VM, 175
- Specify Named Pipe dialog box, in Add Hardware Wizard, 100
- spindle speed of physical hard drives, 57
- split brain problem, avoiding for failover clusters, 319
- srvany.exe tool, for running VMs as services, 123
- ssh. *See* secure shell (ssh)
- SSL, configuring for you Virtual Server, 141
- SSL and MUI, understanding, 165–166
- stand-alone DFS root
 - DFS link limitations, 266
 - vs. domain-based root considerations, 266
- standard naming conventions, for first AFS cell, 290
- standby VM server
 - advantages of, 481
 - automating startup for, 489–493
 - configuring how monitoring check responds to failure, 490–491
 - configuring monitoring criteria for, 489–490
 - example of, 482
 - maintaining, 480–493
 - maintaining with disk mirroring, 485–489
 - maintaining with scheduled backups and restores, 483
 - maintaining with shared storage, 483–484
 - setting up, 482–483
 - shared storage configuration, 484
 - specifying VM startup script, 491
 - synchronized via software RAID 1, 485
 - testing startup of, 493
 - third-party applications for automating startup, 489
- StarPort iSCSI initiator, RAM disk, and virtual DVD emulator, product overview, 521–522
- startup applications, removing unnecessary to recover disk space, 80
- Startup/Shutdown options, managing for new guest VMs, 174
- “StarWind (iSCSI Target) and StarPort (iSCSI Initiator)”
 - white paper by Rocket Division Software (2004), 523
- StarWind iSCSI Service
 - configuring physical drives to be shared, 400

- configuring the iSCSI client connection
 - security settings, 398–399
 - connecting to the StarWind management UI, 396–398
 - StarWind iSCSI target, product overview, 519–521
 - StarWind software
 - configuring, 395–400
 - installing, 394–395
 - setup screen, 395
 - static load balancing, with round-robin DNS, 349
 - static load balancing (active-active) mode, high availability with for failover clusters, 318–319
 - status.pl script, for getting the power status of a VM, 221
 - Stephens Company
 - the problem: network data management and movement, 496–497
 - the solution: global namespace, 497–501
 - storage area networks (SANs)
 - adding fault tolerance to, 466
 - advantages of over other storage types, 422
 - defined, 63
 - example of simple, 422
 - and Microsoft VMs, 64
 - providing security for, 459–460
 - sharing physical storage devices with, 462
 - storing and executing virtual disks over, 12
 - understanding zoning for, 428–429
 - in virtual ISs, 476–478
 - and VMware VMs, 64
 - storage devices
 - backing up VMs to, 228–229
 - supported by SNIA Shared Storage Model, 449
 - storage directory, naming conventions for VMware VMs, 87
 - storage fabric, defined, 63
 - storage management functions, supported by SNIA Shared Storage Model, 449
 - storage model, classic, 446–447
 - Storage Network Industry Association, web site address, 434
 - storage networking
 - understanding, 413–434
 - web site address for technology and concepts information, 434
 - storage options, considering for VMs, 56–66
 - storage point, naming for your VMs, 174
 - storage virtualization, 435–469
 - storage-based architecture, understanding in SNIA Shared Storage Model, 463
 - storage-network aggregation devices,
 - function of in block-based storage architectures, 456
 - storage-network attach devices, function of in block-based storage architectures, 456
 - storage-network attached devices, function of in combined block and file/record layers, 456
 - StorageTek, web site address, 468
 - StorageX. *See* NuView StorageX
 - stripe set, use of in RAID level 0, 436–437
 - striping, as function of block aggregation, 453–454
 - stub file, embedded pointer contained in, 466
 - Swap Configuration screens, in ESX Server installation process, 168
 - SWAT. *See also* Samba Web Administration Tool (SWAT)
 - using to configure your Samba server, 280
 - SWAT RPM, installing, 280
 - switch flooding, using IGMP Multicast setting to correct, 356
 - switched-fabric topology, for Fibre Channel networks, 427
 - switches
 - for connecting multiple networking devices, 22–24
 - popular port types for Fibre Channel networks, 426
 - Sysprep tool
 - command-line help options, 215
 - web site address for downloading, 214
 - Sysprepped VMs, using, 214–215
 - system backups
 - performing, 466
 - performing full system recovery with, 259–261
 - restoring flat-file VM backups, 261
 - restoring online VM backups, 260–261
 - System Event log, convergence event stored in, 357
 - System Management BIOS (SMBIOS), UUIDs based on, 121
 - System State, problems associated with system backups, 481–482
- ## T
- Tar, backing up a VM's directories to an NFS mount point, 236–237
 - TCP/IP
 - function of, 17
 - used by network computers to communicate, 17

- teaming and load balancing, with clustering and network adapter teaming, 49
- teaming software, availability of, 51
- termination, understanding for SCSI buses, 418–421
- terminator types, for SCSI termination, 420–421
- test bench, determining disk size needs for, 62
- text mode, entering from Vi editor command mode, 162
- Thomas, Keir
 - Beginning SUSE Linux: From Novice to Professional* (Apress, 2005) by, 154
- time synchronization, synchronizing for all computer networks, 271–272
- Time Zone Selection screen, in ESX Server installation process, 159
- Tip of the Day pop-up, disabling in VMware Workstation, 72
- Token Ring network, using NAT to connect virtual networks to, 19
- transaction-based servers, virtualization of, 513–514
- transfer rate of disk drives, 58
- Transitive's QuickTransit emulator, function of, 2
- Transmission Control Protocol/Internet Protocol (TCP/IP). *See* TCP/IP
- Tuning and Customizing a Linux System* (Apress, 2002)
 - by Daniel L. Morrill, 74
- tuning tips
 - disabling extraneous services, 81–82
 - memory tweaking for improving performance, 82–83
 - resolving problems with hanging processes, 82
 - for VM hosts, 78–84

U

- UltraMonkey software
 - installing configuration and management packages, 335–336, 410–411
 - using to build Linux VM clusters, 409
 - web site address for downloading, 334, 409
- Undisker, web site address for downloading, 89, 175
- Undo Disks option, for committing or discarding changes to a VM, 107
- undo log, location of in VM's disk redo statement, 209
- undopoint.action statement, function of and options list, 209
- unicast and multicast mode, setting the NLB cluster service to run in, 354–356
- unicast mode vs. multicast mode, 354
- unicast nodes, performing initial configuration for each in NLB cluster, 364–365
- Universally Unique Identifiers (UUIDs). *See* VMware Universally Unique Identifiers (UUIDs)
- update sequence number (USN) journal record, using to maintain replication integrity, 270
- USB CD-ROM drive, steps for mounting, 187–189
- USB connectivity, scripting for ESX Server, 191–193
- USB controller
 - adding support for, 99
 - configuring for GSX Server, 185
- USB drives
 - mounting a USB CD-ROM drive, 187–189
 - mounting each time the system is rebooted, 188
 - using to run virtual machines for ESX Server, 186–187
- USB external hard drives, making available to ESX Server console, 189–191
- USB thumb drives
 - making available to ESX Server console, 189–191
 - script for mounting, 192
 - script for unmounting, 192–193
- USB-to-IP technology, for making USB devices available to guests, 187
- use-imaging software, Norton Ghost, 15
- user accounts, creating for your AFS server, 297–298
- USN journal record. *See* update sequence number (USN) journal record
- UUID.BIOS, comparison of
 - UUID.LOCATION to on each boot, 200
- UUID.LOCATION
 - comparison to UUID.BIOS on each boot, 200
 - used to identify servers, 121
- UUIDs. *See* VMware Universally Unique Identifiers (UUIDs)

V

- v option, for VMware Workstation CLI, 129
- /v option, with vmware-mount for mounting a specific partition, 113
- verbose option, using with vmkusagectl command, 216
- Veritas, for policy-based data management software, 466

- Veritas NetBackup Storage Migrator, web site address, 467
- *.vhd configuration file, changing settings in, 210
- .vhd filename extension, for Microsoft VM disks, 12
- Vi editor for Linux, 162–163
- VIN. *See* virtual infrastructure node (VIN)
- Virtual CDrom utility, for mounting ISO images with Microsoft OSs, 182
- virtual devices
 - changing behavior of by changing device's value, 208
 - controlling behavior of, 116
- virtual disk device node, selecting in Special Advanced Options dialog, 90–91
- virtual disk mode, selecting in Special Advanced Options dialog, 90–91
- virtual disk partitions, mounting certain types of, 113
- virtual disks, 12
 - configuring to be independent, 98
 - creating new for VMware virtual machines, 89–92
 - managing, modifying, and creating from the command line, 211–212
 - naming and selection storage location for, 90
 - restoring, 113
 - running vmware-mount command to list, 112
 - storing and executing over a network, 12
- virtual disk types
 - functionally equivalent, 12
 - table of, 11
- virtual file systems, using, 263–307
- virtual floppy drive, creating, 99, 182
- virtual hard disk (VHD), 12
 - adding to configuration of each cluster-node VM, 390–391
 - assigning a shared to a VM, 391
 - creating and configuring shared for Virtual Server VMs, 390
 - determining disk size needs for, 175–176
 - specifying name and storage location for, 176
- Virtual Hard Disk Location dialog box
 - for choosing a storage point in Windows VM, 103
 - for choosing a storage point of Linux VM, 105
- Virtual Hard Disk Options dialog box
 - for choosing a Linux VM disk option in Virtual PC, 105
 - for choosing a Windows VM disk option in Virtual PC, 104
- virtual hardware options
 - configuring for VMware GSX Server and ESX Server, 180–193
 - VMware for Windows and Linux, 95–101
- virtual hardware specifications, VMware's, 7
- virtual host, defined, 310
- virtual IDE devices, controlling for virtual machines, 116
- virtual information system (virtual IS). *See also* virtual IS
 - consolidation of servers for, 476
 - pros and cons of using a SAN for, 476–477
 - putting it all together, 471–494
 - using DFS in, 479–480
 - using standby VM hosts, 476
- virtual infrastructure node (VIN)
 - contents of, 33
 - creating, 33
- Virtual Infrastructure SDK, 221
- virtual IS, reviewing the elements of, 471–480
- Virtual Machine Additions
 - command-line options for unattended install, 109
 - downloading from Microsoft's web site, 195
 - installing, 108–109
 - installing on Virtual Server, 194–195
 - table of command-line options, 195
 - web site address for downloading, 108
- Virtual Machine Console
 - changing port used by GSX Server for Linux, 147
 - working with, 152–153
- Virtual Machine Editor
 - locating *.vmx configuration files with, 207
 - locating the *.vmx configuration file in, 115
- virtual machine host
 - avoiding bottlenecks by investing in major systems, 36
 - evaluating requirements for, 36–38
 - function of in virtualized information systems, 474–476
 - memory requirements for, 41–42
 - preparing, 35–67
 - SAN integration, 474–475
 - selecting the motherboard for, 38–45
 - system priorities, 37
- virtual machines (VMs). *See also* VMs
 - adding a parallel port to Linux guest, 100
 - adding DVD/CD-ROM drives to, 98
 - adding Ethernet adapters to, 99
 - adding sound adapters to, 99
 - backing up and recovering, 223–261
 - backing up directly to storage devices, 228–229

virtual machines (*continued*)

- backing up over the LAN, 225–227
 - backing up through the VM host, 227–228
 - basics for keeping host and guests
 - running optimally, 135
 - BIOS for, 24–25
 - building in VMware ESX Server, 176–177
 - building Windows VMs vs. Linux VMs, 173–178
 - considering storage options for, 56–66
 - controlling Ethernet adapters for, 117
 - controlling IDE devices for, 116
 - copying and moving GSX Server VMs to other hosts, 199–200
 - creating a serial port for, 100
 - deploying and managing on the desktop, 85–138
 - deploying with Microsoft Virtual PC, 69–71
 - deploying with VMware GSX Server and ESX Server, 173–178
 - determining amount of RAM needed for, 174
 - determining power-up or power-down characteristics for, 174
 - emulating, 2
 - evaluating host requirements for, 36–38
 - examining the anatomy of, 1–34
 - example of newly created with virtual devices, 92
 - general recovery process after crashes, 260–261
 - importance of generic SCSI to, 25
 - importance of purchasing quality memory for, 42
 - installing and deploying on enterprise servers, 139–171
 - installing VMware Tools for Windows on guest, 93–94
 - making Network Type selection for, 175
 - managing in a production or test environment, 109–128
 - managing server class, 197–211
 - managing Startup/Shutdown options for, 174
 - maximum CPU and memory specifications, 9
 - Microsoft and VMware supported I/O devices for, 25–26
 - Microsoft network configurations, 52–53
 - migrating between, 34
 - modifying configuration files, 197–200
 - monitoring and configuring performance, 134–138
 - monitoring performance of, 215–221
 - monitoring performance of for Virtual Server, 220–221
 - moving from one VM host to another, 512–513
 - naming your storage point for, 174
 - network interface card specifications for, 22
 - network interface card specifications table, 20
 - open source, 4
 - planning for simplicity of networks, 48
 - RAM (computer memory) for, 9–10
 - removing devices from, 95–97
 - renaming GSX Server guests, 198
 - replicating entire test lab environments using, 510–511
 - restoring online backups, 260–261
 - running as services, 123–128
 - running backup agents on, 224–229
 - setting Access Rights for, 174
 - setting the memory size for, 116
 - setting up as standby server, 482–483
 - space needed for virtual hard disks, 175
 - specifying maximum space used by guests, 175
 - steps for configuring VM power-off settings, 247–248
 - steps for taking online snapshots of, 257
 - system requirements for good performance, 5
 - using as hot backup servers for downtime, 511–512
 - using DHCP with, 17–18
 - using hb_check.pl script to monitor, 221
 - using status.pl script to get power status of, 221
 - using Sysprepped, 214–215
 - using Windows Backup utility for backups of, 231–234
 - virtual disks for, 10–11
 - VMware network configurations, 51–53
- Virtual Machine Settings dialog box
- for adding/removing devices from VMs, 96
 - for configuring VM power-off settings, 247–248
 - configuring VM snapshot options in, 259
 - configuring VMs to close after powering off in, 241
 - for VMware Workstation, 120
- virtual machine-to-host networking, function of, 53
- virtual network card. *See* network interface card specifications
- Virtual PC
- best-practice system minimums for installing, 70
 - building Windows VMs with, 101–104
 - configuration options, 69
 - differencing disks, 14

- dynamic disk setting as default for, 13
 - installing on a Windows XP host
 - computer, 69–71
 - introduction to, 26–27
 - limitations of virtual switches, 23
 - memory overcommitment prohibited by, 10
 - Microsoft Installer options table, 213
 - Performance options, 137–138
 - preliminary tasks before moving guest VMs, 122
 - Setup.exe command for installing, 131–134
 - steps for moving guest VMs, 123
 - steps for renaming VMs, 110–111
 - table of keyboard shortcuts, 134
 - verifying version number, 71
 - virtual hardware options, 106–108
 - vs. Virtual Server, 27
 - VM support for network interfaces, 52
 - VMware *.vmc configuration files, 117–118
- Virtual PC 2004
- running flat-file backups, 251–252
 - variables for running flat-file backups, 251
 - VM network interface card specifications, 20
- Virtual PC 2004 InstallShield Wizard, for installing Virtual PC software, 70
- Virtual PC 2004 SP1
- installing, 71
 - web site address for downloading, 70
- Virtual PC 2004 SP1 Readme files, web site address for downloading, 70
- Virtual PC CLI administration, 131–134
- Virtual PC.exe command
- for launching VMs in a specific mode, 131–134
 - table of options for, 133
- Virtual PC Performance options
- considerations when configuring
 - background performance, 138
 - minimal considerations for selecting, 138
 - table of, 137
- Virtual PC Performance options dialog box, 137
- Virtual PC service, modifying the shortcut target, 128
- Virtual PC VMs, copying and moving to other hosts, 121–123
- virtual SCSI adapter, adding to Virtual Server VMs, 389
- virtual SCSI devices, controlling for virtual machines, 117
- virtual server
- defined, 310
 - defined for failover clusters, 311
- Virtual Server
- differencing disks, 14
 - identifying system bottlenecks in guest VMs, 220–221
 - limitations of virtual switches, 23
 - memory overcommitment prohibited by, 10
 - monitoring performance of guest VMs, 220–221
 - vs. Virtual PC, 27
 - vs. VMware GSX Server, 31
- Virtual Server *.vmc configuration files, working with, 210–211
- Virtual Server 2005
- editions, 29
 - running flat-file backups, 252–257
 - VM network interface card specifications, 20
- Virtual Server 2005 Enterprise. *See also* Microsoft Virtual Server 2005 Enterprise
- VM network interface card specifications, 20
- Virtual Server 2005 VM startup, scripting, 492–493
- Virtual Server Administration Web site
- configuring, 140–141
 - steps for creating VMs through, 193–194
- virtual server architecture, example of, 505
- Virtual Server VM's
- changing the MAC address, 210–211
 - setting start and stop actions, 253
 - steps for backing up all, 253
 - variables for backing up all on a Virtual Server host, 254
 - variables for backing up a single, 255
- virtual switches, types of based on protocols employed, 24
- virtual tape libraries, using to create logical tape libraries, 467–469
- virtual tape server appliances, vendors for, 468
- VirtualCenter
- function of, 33
 - for performing fault monitoring, 221
- virtualization appliance, 463
- virtualization layer, duties of, 2
- virtualization product roundup, 495–526
- virtualization software, function of, 1–2
- virtualized (synthetic) motherboard, impact of BIOS on, 24–25
- virtualizing storage, 435–469
- VM applications, installing on desktops, 69–84
- VM CLI administration and keyboard shortcuts, 128–134

- VM cluster nodes, setting up for iSCSI access, 400–404
- VM clusters
 - configuring to support SCSI and ICSI server clusters, 379–409
 - folder structure for VMware Workstation cluster, 381
- VM configurations
 - adding for VMware Workstation service, 125
 - backing up and modifying, 109–114
- VM configuration selection type, selecting OS you will be using for guest VM, 86
- VM hardware, choosing, 5–8
- VM host
 - backing up a VM through, 227–228
 - comparisons using multiple controllers and disks in, 58
 - running backup agents on, 230
 - tuning tips, 78–84
- VM Memory selection dialog box, in New Virtual Machine Wizard, 103
- VM monitor (VMM), for safely processing privileged instructions, 9
- VM networking configurations, 51–53
- VM networking protocols, TCP/IP vs. DHCP, 17
- VM networks
 - considerations for building and deploying, 45–53
 - deciding on public vs. private, 45–47
 - importance of high availability and performance of, 47–48
 - planning for simplicity of, 48
- VM performance, monitoring and configuring, 134–138
- VM products, introduction to, 26–34
- VM server, maintaining a warm standby, 480–493
- VM service
 - adding the path to the VM configuration file, 125
 - creating, 124–128
 - labeling, 124
 - modifying the shortcut target, 127–128
 - troubleshooting problems with, 128
- VM service shortcut folder, creating a shortcut in, 127
- VM startup, scripting, 491–493
- VM types, introduction to, 2–4
- VM virtual switches, supported by Microsoft and VMware, 22–24
- *.vmdk file, typical entries regarding the virtual disk, 206–207
- .vmdk filename extension, for VMware VM disks, 12
- VMFS volume, making writable, 212
- VM-hosted Samba server, connecting to from a Windows VM, 279
- vmkfstools command
 - function of for ESX Server, 212
 - option to display virtual disk geometry, 212
- vmkpcidiv command
 - items you can configure with, 186
 - usage of and options for ESX Server, 185
- vmkusage, for monitoring ESX Server performance, 215–216
- vmkusagct1 command options, table and function of, 216
- vmkusagct1 install command, for activating vmkusage, 215–216
- Vmotion, function of, 33
- VMs. *See also* virtual machines (VMs)
 - adding DVD/CD-ROM drives to, 98
 - adding Ethernet adapters to, 99
 - adding hardware to, 96–97
 - adding sound adapters to, 99
 - backing up and modifying, 109–114
 - backing up directly to storage devices, 228–229
 - configuring to automatically close for flat-file backups, 241
 - creating a serial port for, 100
 - deploying, 5
 - deploying and managing on the desktop, 85–138
 - managing in a production or test environment, 109–128
 - migrating between, 34
 - monitoring performance of, 215–221
 - pros and cons of backing up in a suspended mode, 122
 - running as services, 123–128
 - running backup agents on, 224–230
 - selecting a name and a storage point for, 86–87
 - selecting network type for, 88–89
- vm-support script, for backing up VMware configuration files, 114
- _VMTest1 service, enabling to interact with the desktop, 125
- VMware
 - adapter teaming support for GSX Server on Windows OSs, 50
 - additional tools and management packages, 32
 - BIOS packaged with, 24–25
 - download web site address, 72
 - dynamic disks, 12–13
 - installing the DiskMount utility, 73
 - network configurations, 51–53

- physical disks, 13
- preallocated disks, 13
- removing devices from VMs, 95–97
- setting and configuring snapshot options, 258
- support for generic SCSI for Linux VMs, 55–56
- supported I/O devices for VMs, 25–26
- taking online snapshots, 257–259
- undoable disk, 13–14
- virtual disk drive geometry, 206–207
- virtual disks, 11–16
- virtual hardware options for Windows and Linux, 95–101
- virtual hardware specifications, 7
- virtual network types supported by, 20–21
- VirtualCenter and VMotion, 33
- vmware-vdiskmanager utility for Windows hosts, 211–212
- web site address for white papers, etc., 184
- VMware *.vmc configuration files, for Virtual PC, 117–118
- VMware *.vmx configuration files
 - Ethernet setting, 210
 - example of, 115
 - function of, 115–117
 - lines added to after snapshot taken of a VM, 209
 - working with for GSX Server and ESX Server, 207–210
- VMware ACE
 - downloading a demo copy of, 85
 - function of, 34
- VMware configuration files, vm-support script for backing up, 114
- VMware ESX Server. *See also* ESX Server
 - backing up directly to storage devices, 228–229
 - best-practice system minimums for deployment, 154–155
 - booting from the CD-ROM for installation, 156
 - building VMs in, 176–177
 - checking basic configuration, 161
 - checking IP address information, 161
 - closing the Help window, 155
 - configuration of VMs virtual disk, 177
 - disabling USB support in, 185
 - importing Workstation and GSX Server VMs into, 202–207
 - installation modes, 155
 - installing, 154–160
 - location of setup log files, 160
 - minimum supported host specifications, 32
 - mounting ISO images, 177–178
 - moving or copying VMs to other ESX hosts, 203–204
 - need for two network adapters, 157
 - partitions of the autopartitioning process, 158
 - resolving interrupt conflicts during installation, 156
 - running Terminal Services for Citrix Metaframe, 176
 - SAN-attached deployment, 474–475
 - selecting virtual disk for VMs, 176
 - steps for adding a new virtual machine, 176–177
 - steps for installing, 156–160
 - steps for renaming guest VMs, 201–202
 - steps for verifying amount of free space, 202
 - verifying configuration information, 161
 - VM network interface card specifications, 20
 - working with the management interface, 165
- VMware forums, web site address for accessing, 116
- VMware GSX Server
 - adding first node disks to second node configuration, 387
 - backing up directly to storage devices, 228–229
 - changing the Remote Console port number, 153–154
 - configuring clusters on, 385–387
 - editing the VM configuration file for, 387
 - minimum supported host specifications, 31
 - monitoring performance of, 219–220
 - running flat-file backups for, 247–251
 - standby VM server configuration, 488–489
 - steps to configure shared storage resources, 385–386
 - variables for running on Linux flat-file backups, 250
 - variables for running on Windows flat-file backups, 248–249
 - vs. Virtual Server, 31
 - VM network interface card specifications, 20
- VMware GSX Server cluster, VM hardware configuration, 385–386
- VMware GSX Server for Linux
 - best-practice system minimums for installing, 146
 - changing port used for Virtual Machine Console, 146
 - firewall problems associated with installations, 146

- VMware GSX Server for Linux (*continued*)
 - installing, 146–150
 - installing VMware Tools package built into, 179–180
 - packages available for installation, 146
 - steps for installing the RPM, 147–149
 - steps for installing the TAR, 149–150
- VMware GSX Server for Windows
 - best-practice system minimums for installing, 144
 - installing, 143–145
 - steps for installing, 145
 - web site address for checking for patches and security updates, 145
- VMware GSX Server VMs, building Windows VMs vs. Linux VMs, 173–178
- VMware hotkeys, controlling keyboard shortcuts from, 129–131
- VMware Management Interface
 - installing, 150–151
 - steps for installing, 150–151
 - testing it's functionality, 151
- VMware P2V Assistant. *See* P2V Assistant
- VMware performance counters, function of, 135–136
- VMware Tools
 - Custom Setup dialog box, 93
 - installing, 92–101
 - installing for GSX Server and ESX Server VMs, 178–180
- VMware Tools for Linux
 - installing, 95, 179–180
 - using, 179–180
- VMware Tools for Windows
 - installing on a guest VM, 93–94
 - options in Custom Setup configuration screen, 178
 - using, 178–179
 - web site address for installation information, 94
- VMware Universally Unique Identifiers (UUIDs)
 - for generating a MAC address for a VM, 121
 - problems with when moving location of guest VMs, 200
 - using, 200–202
- VMware VirtualCenter. *See* VirtualCenter
- VMware Virtual Machine Console
 - steps for installing, 152–153
 - working with, 152–153
- VMware VMotion. *See* VMotion
- VMware VMs and SANs, 64
- VMware Workstation. *See also* VMWare Workstation for Windows; Workstation
 - backing up, 111–114
 - backing up directly to storage devices, 228–229
 - best-practice system minimums for installing, 71
 - configuring clusters on, 380–384
 - copying and moving guest VMs in, 119–120
 - deploying VMs with, 85–92
 - disabling Tip of the Day pop-up in, 72
 - edited VM configuration file, 384
 - hardware configuration screen, 383
 - keyboard shortcuts table, 130
 - minimum host specifications, 28
 - overview of variables for flat-file backups, 242
 - running flat-file backups for, 241–247
 - running on Linux flat-file backups, 246–247
 - selecting OS you will be using for guest VM, 86
 - steps for moving guest VMs, 119–120
 - steps for renaming VMs, 110
 - typical path directories for configuration program, 77–78
 - variables used for running Linux flat-file backups on, 246
 - VM network interface card specifications, 20
- VMware Workstation CLI
 - administration and keyboard shortcuts, 129–131
 - checking full range of options available for, 129
- VMware Workstation cluster, configuring shared storage resources for, 382–384
- VMware Workstation for Linux
 - installing, 74–78
 - launching the New Virtual Machine Wizard, 85
 - steps for installing the RPM, 75–76
 - steps for installing the TAR, 76–77
- VMware Workstation for Windows
 - installing, 71–74
 - launching the New Virtual Machine Wizard, 85
- VMware Workstation service
 - creating, 124–128
 - example of VM service error, 126
 - modifying the shortcut target, 127–128
 - secondary approach to get it to work, 126–128
 - what registry might look like after editing, 126
- vmware-config.pl configuration program, resolving hanging issue with, 74

- vmware.exe /? Command, checking range of options for VMware Workstation CLI with, 129
- vmware-mount command options, 111
- VMware's HCL, web site address for, 8
- VMware's Scripting API, web site address for, 221
- vmware-vdiskmanager, web site address for information about, 211
- vmware-vdiskmanager utility for Windows hosts, VMware, 211–212
- VMwareVirtual SMP, 32
- *.vmx configuration files. *See* VMware *.vmx configuration files
- Volume, as main administrative unit in AFS, 290
- volumes
 - creating for your AFS server, 298–299
 - naming restrictions, 291
 - space quota assigned to, 291

W

- w32tm command, for adding the time source, 271–272
- Wall, Larry
 - Perl created by in 1987, 191
- WAN, impact of running a DFS across, 263
- web site address
 - for accessing VMware forums, 116
 - for Advanced Digital Information Corporation, 427
 - for AMANDA, 238
 - for article on creating base image and redo log files, 209
 - for assistance with the diskpart command, 112
 - for binary numbering, routing and TCP/IP subnetting information, 226
 - for Bochs, 4
 - Brocade, 426
 - for checking for GSX Server patches and security updates, 145
 - for checking for Virtual Server patches and updates, 143
 - for checking Samba prerequisites, 271
 - for “Choosing an Availability Strategy for Business-Critical Data”, 270
 - Cisco Systems, 426
 - for CommVault, 466
 - for comprehensive resource regarding Samba, 278
 - for Coyote Point Systems, 474
 - for Crimson Editor color-sensitive editor, 117
 - Crossroads Systems, 427
 - for documentation about setting up DFS in Windows, 270
 - Emulex, 425
 - for ESX Server's manual, 212
 - for Fibre Channel Industry Association, 424
 - for free online SAN courses by EMC Legato, 478
 - for Free Software Foundation definition, 4
 - for IIS 6.0 Resource Kit Tools, 142
 - for information about ArlaAFS, 288
 - for information about cron daemon, 239
 - for information about Fibre Channel switches, 426
 - for information about HBAs, 425
 - for information about installing VMware Tools, 94
 - for information about iSCSI, 431
 - for information about Microsoft clustering, 330
 - for information about Norton PartitionMagic, 98
 - for information about Xen, 4
 - for Intel CPUs and required bus speeds, 41
 - for iSCSI initiator software, 400
 - for “Kerberos Interoperability”, 278
 - for “Kerberos V5 Installation Guide”, 276
 - for latest LVS software documentation, 377
 - for Linux High Availability Project, 333
 - Linux iSCSI project for iSCSI information, 431
 - for LVS cluster software, 377
 - for MagicISO, 89
 - McDATA, 426
 - for Microsoft licensing brief, 31
 - for Microsoft's HCL, 8
 - for Microsoft's Script Center Script Repository, 196
 - for Mirrordir tool, 333
 - for “Monitor Disk Space on Multiple SQL Servers” article, 220
 - for NuView StorageX, 465
 - for “Official Samba-3 HOWTO and Reference Guide”, 278
 - for Open GFS home page, 333
 - for Open Source Initiative definition, 4
 - for OpenAFS, 292
 - for OpenSSH ssh server for Windows, 246
 - for PenguiNet shareware software, 243
 - Project RC5, 3
 - for prominent vendors in the RAID market, 442–443
 - for pulist, 82
 - for PuTTY, 157
 - QLogic, 425
 - for Red Hat Enterprise Linux System Administration Guide, 239

- web site address (*continued*)
 - for Red Hat's HCL, 8
 - for Rocket Division Software, 394, 477
 - for Rsync tool for remote replication of file data, 333
 - for Samba, 273
 - Seti@Home project, 3
 - for SMBIOS specification, 200
 - for Storage Network Industry Association, 434
 - for Sysprep tool, 214
 - for UltraMonkey software, 334
 - for Undisker, 89
 - for the USB/IP Project, 187
 - for Veritas, 466
 - for Virtual CDrom utility, 182
 - for Virtual Machine Additions, 108, 195
 - for Virtual PC 2004 SP1, 70
 - for Virtual PC 2004 SP1 Readme files, 70
 - for VMware DiskMount Utility guide, 113
 - for VMware downloads, 72
 - for VMware enableSSL script, 165
 - for VMware Tools for Windows OS information, 179
 - for VMware white papers and other information, 184
 - for VMware's HCL, 8
 - for VMware's Scripting API, 221
 - for vmware-vdiskmanager information, 211
 - for Webmin for your Linux system, 280
 - for Windows 2003 Resource Kit, 123
- Webmin, downloading and installing for your Linux system, 280
- Wget, for downloading a Samba compatible Kerberos file, 275
- Windows 2003 CD-ROM, DFS Support Tools and other tools on, 264
- Windows 2003 Resource Kit. *See* Microsoft Windows 2003 Resource Kit
- Windows 2003 Server
 - disk space setup requirements, 38
 - RAM requirements, 37
- Windows Backup utility
 - adding a generic SCSI storage device with, 234
 - steps for performing a backup with, 232–233
 - steps for using to restore data, 233–234
 - using for non-agent-based backups, 231–234
- Windows computer account password
 - changes, disabling, 482–483
- Windows DFS, implementing, 265–270
- Windows driver alert, in Hardware Installation dialog box, 94
- Windows environments, managing user home directories with global namespaces, 500
- Windows firewall
 - pros and cons of disabling, 80–81
- Windows Installer, verifying version number of, 144
- Windows Network Load-Balanced (NLB) clusters. *See also* NLB clusters
 - building, 353–374
- Windows NLB clusters. *See also* NLB clusters
 - setting up, 406–409
- Windows NT primary domain controller (PDC), configuring Samba as, 281–284
- Windows Server 2003, maximum nodes for load-balanced cluster, 350
- Windows Server 2003 Cluster Service
 - adding a node to, 327–328
 - installing, 326–330
 - installing in a two-node VM cluster
 - configuration, 404–406
 - prerequisites for installing, 326
 - steps for installing, 326–327
- Windows Server 2003 Enterprise Edition VMs, configuring two for VMware Workstation cluster, 380–381
- Windows Server clusters, setting up in a VM environment, 380–404
- Windows System Preparation Tool (Sysprep), using, 214–215
- Windows VMs
 - backing up on a Linux host, 246–247
 - batch file for backing up on a Windows host, 242–243
 - building with Microsoft Virtual PC, 101–104
 - non-agent-based backups, 231–234
- Windows XP
 - disabling extraneous services, 81–82
 - memory tweaking for improving performance, 82–83
 - using performance counters in, 136
- Windows XP kernel, running in memory to improve performance, 82
- Windows XP Professional Resource Kit, “Kerberos Interoperability” in, 278
- Windows-hosted GSX Server
 - support for Broadcom-based network adapters, 50
 - support for Intel-based network adapters, 50
- wlbs.exe command
 - managing NLB clusters with, 371–373
 - for remotely administering NLB cluster nodes, 358

- table of options for, 372
- table of remote options for, 372
- Workstation Management User Interface (MUI), returning to after disk creation, 92
- Workstations
 - dynamic disk setting as default for, 13
 - limitations of virtual switches for, 23
 - using as virtual machine hosts, 36
- Workstation's Installation Wizard
 - for installing VMware Workstation for Windows, 72
- world-wide names (WWNs), using to
 - configure zoning for SANs, 428–429
- WWN zoning, function of, 428–429

X

- X option, for VMware Workstation CLI, 129
- x option, for VMware Workstation CLI, 129
- X Windows
 - configuring for Linux guest VM, 105–106
 - problems with installing on a production ESX Server system, 169
 - starting to test MUI functionality, 151
- Xen, open-source virtualization application, 4

Z

- zoning
 - configuring on a Brocade Silkstorm 2400 switch, 429
 - understanding use of for SANs, 428–429