THE EXPERT'S VOICE® IN NETWORKING



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®

WIZ, Inc. EXHIBIT - 1009
WIZ, Inc. v. Orca Security LTD.

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 Winquist
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** 35

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 471

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 495

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!

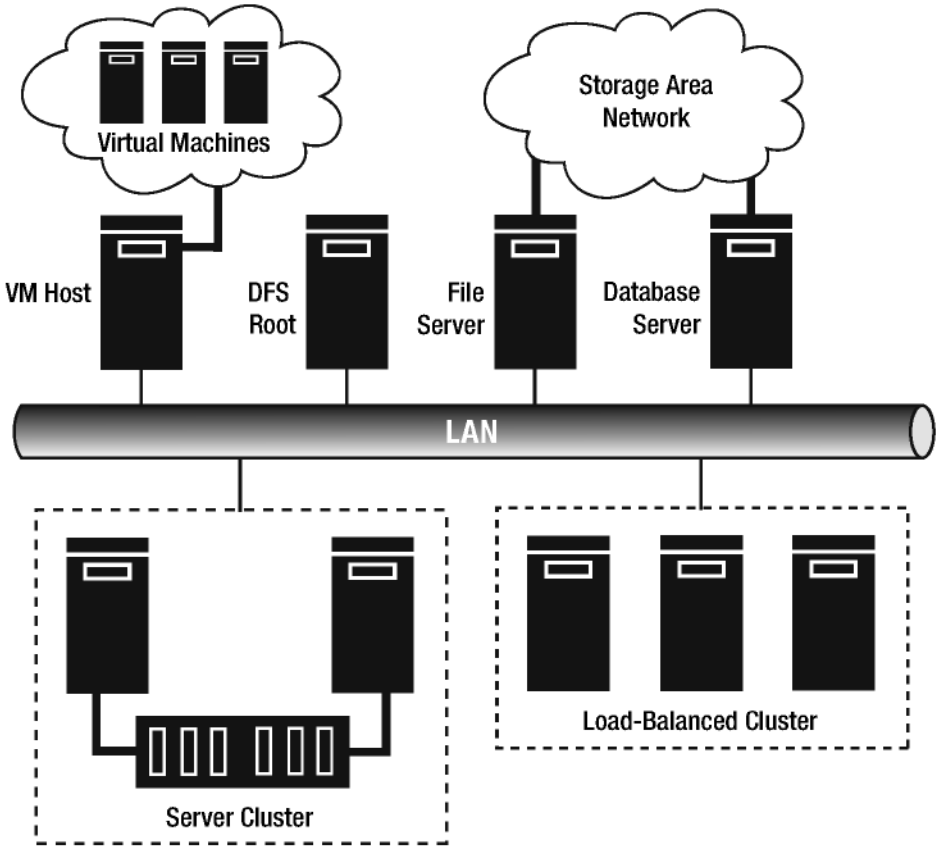


Figure 1. *A virtualized information system*

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 as 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.

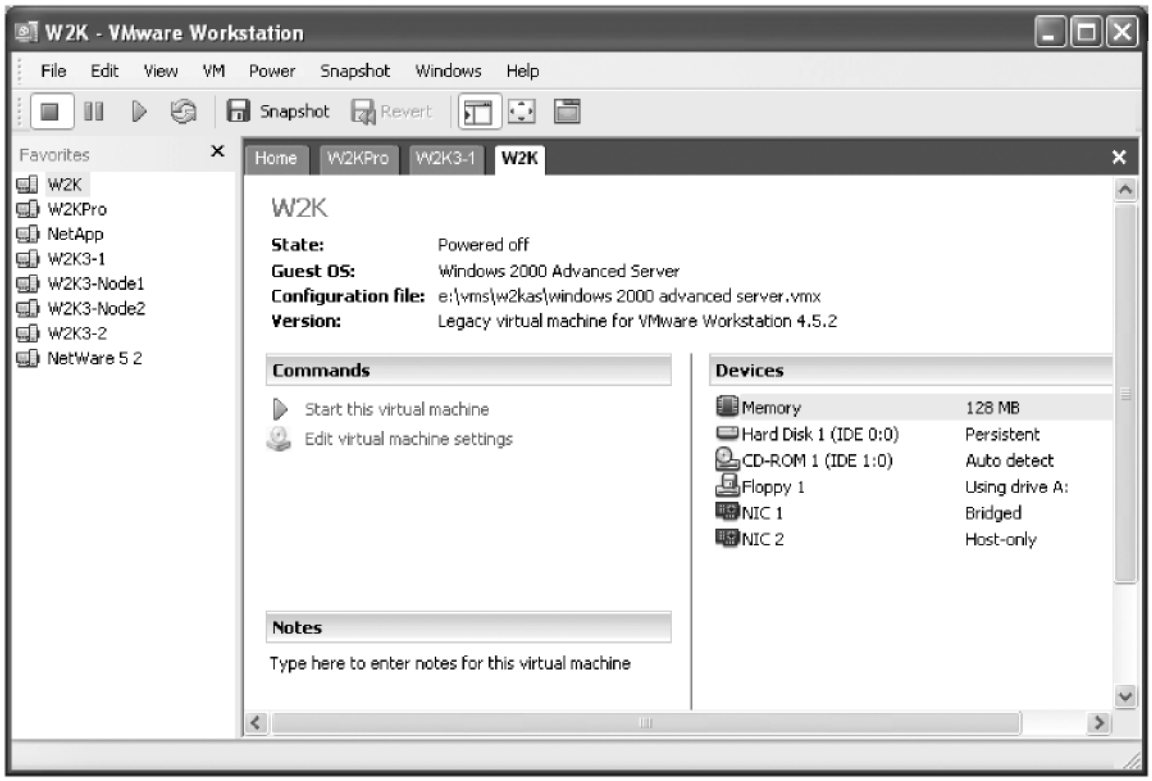


Figure 2. *VMware Workstation UI*

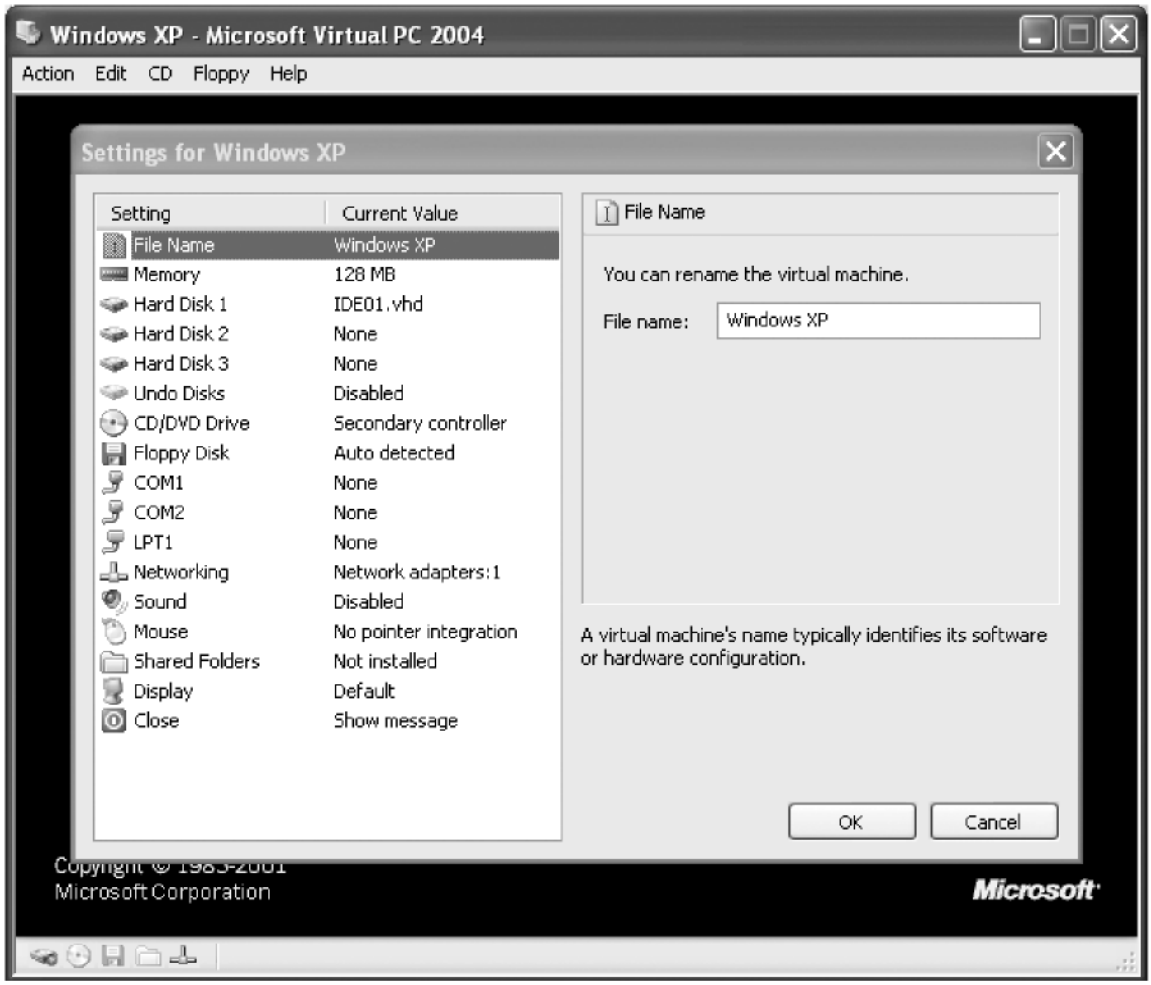


Figure 3. *Virtual PC UI*

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.

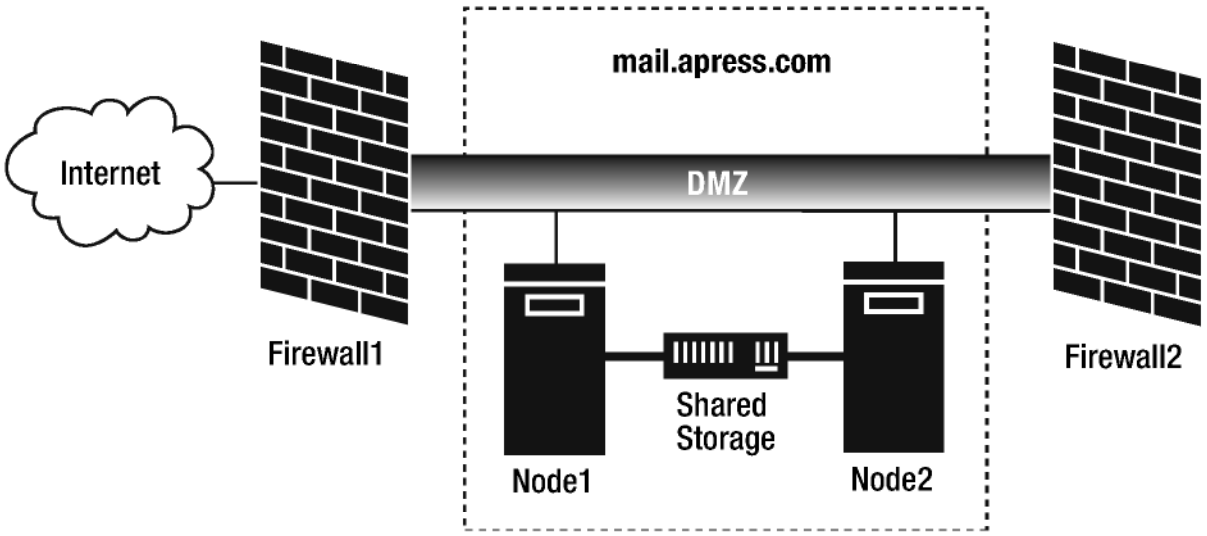


Figure 4. *Two-node mail server cluster*

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.

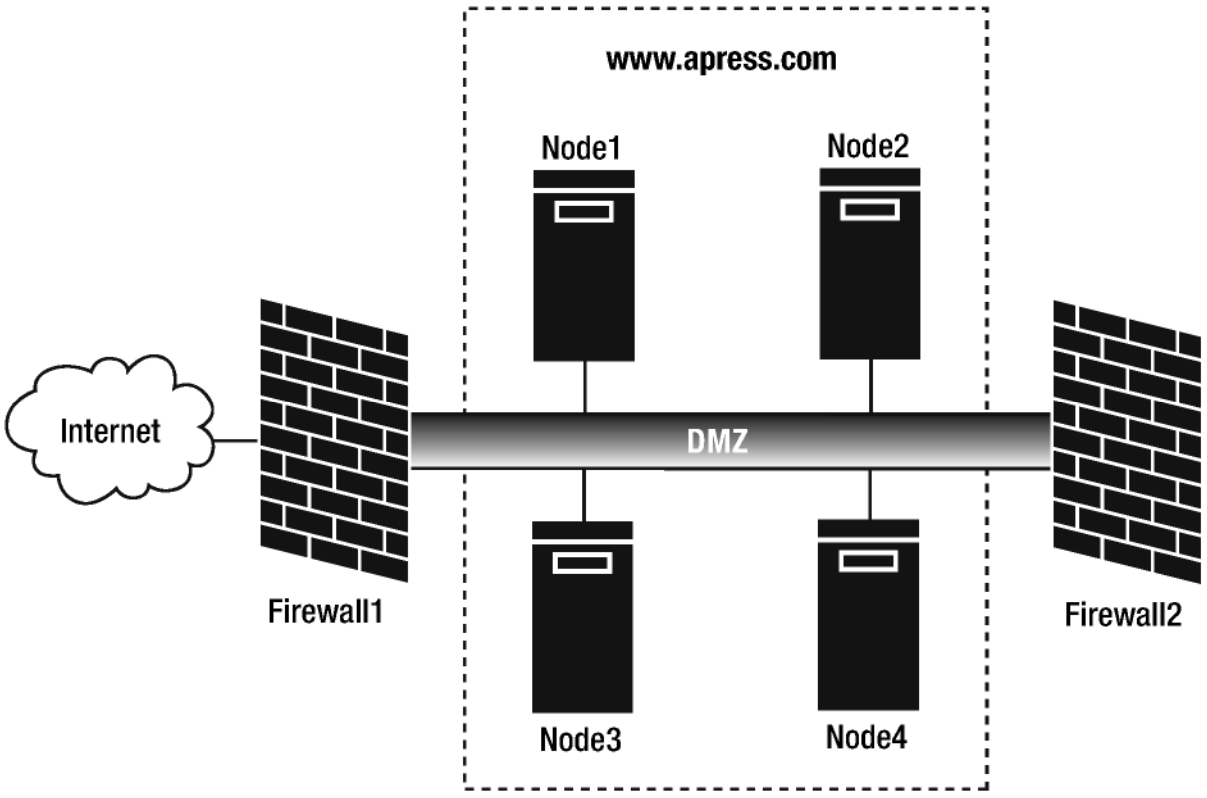


Figure 5. *Four-node load-balanced Web server cluster*

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.

CHAPTER 1

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/SRG/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 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 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.

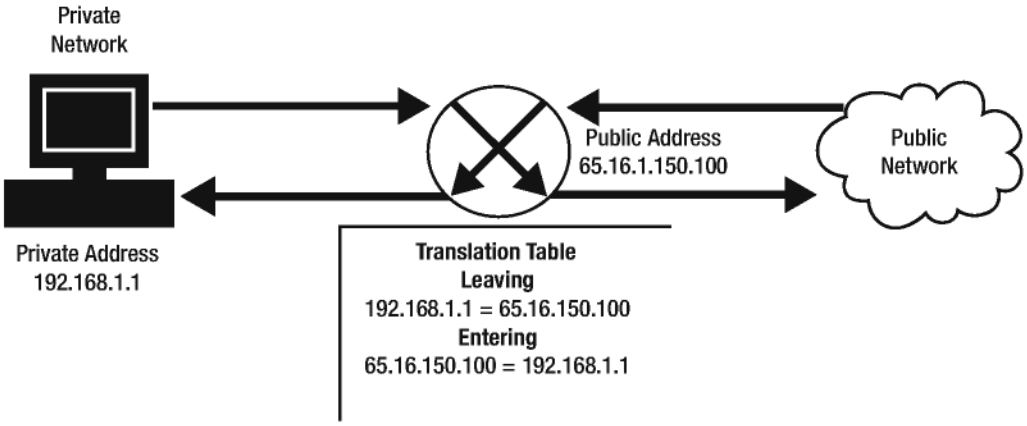


Figure 1-1. *The NATing process*

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.

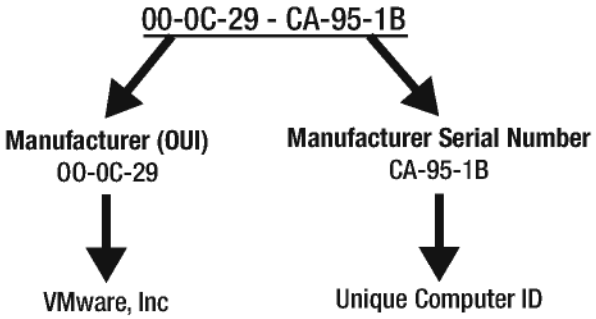


Figure 1-2. *MAC address breakdown*

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 9*x*, 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 you 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 multiprocessor (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 8*i*, 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 polices 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 polices 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.

CHAPTER 6

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 9*x* 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 `umount` the CD-ROM, like so:

```
mount /dev/cdrom /mnt/cdrom
cd /tmp
tar zxf /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/winxpvirtualcdcontrol➥panel_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.presen t= 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 `vmkpcidivy` command with the `s` option to specify the sharing of devices. You should execute the `vmkpcidivy` 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/vmkpcidivy [-i] [-l name] [-m memsize] [-q] [-v bus:slot:fcn]
```

Table 6-1. *`vmkpcidivy` 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: `vmkmod`, `vmkdump_part`, or `vmkdump_dev labels vmfs_part`
`-v`	Assigns device to VMs

To get an idea of how the command works, run `vmkpcidivy -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 `vmkpcidivy` 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 *shbang* 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.mspx`. 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.mspx`.
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,resets=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=ffffffff
   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_Soruce_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 2000000000. 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 netwware. 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
autoRevert	On power down, reverts to original state of snapshot
autoCommit	Automatically updates snapshot on power down
prompt	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 VMs 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_vmvkfstools.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 vmhbax:x:x:x:<virtual_disk_name>.dsk`.
- To import a virtual disk into a VMFS partition, use `vmkfstools -i <virtual_disk_source>.dsk vmhbax:x:x:x:<virtual_disk_destination>.dsk`.
- To export a virtual disk from VMFS partitions for use on GSX Server or Workstation, use `vmkfstools -e <destination_disk_name>.vmdk vmhbax:x:x:x:<source_disk_name>.dsk`.
- To list VMFS virtual disks, use `vmkfstools -l vmhbax: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>.dsk
```

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 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
`/forceshutdown`	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.

CHAPTER 7

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.

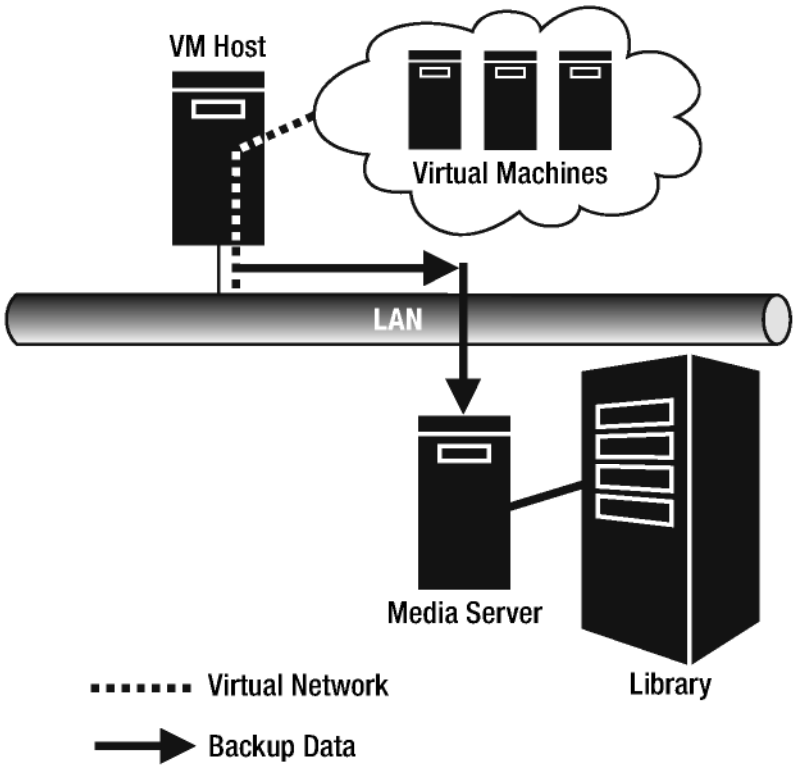


Figure 7-1. *Backing up a VM over the LAN*

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 (1111111) 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 (100Mb ÷ 8) of bandwidth, which translates to 45GB per hour (12.5MB × 60 seconds × 60 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.

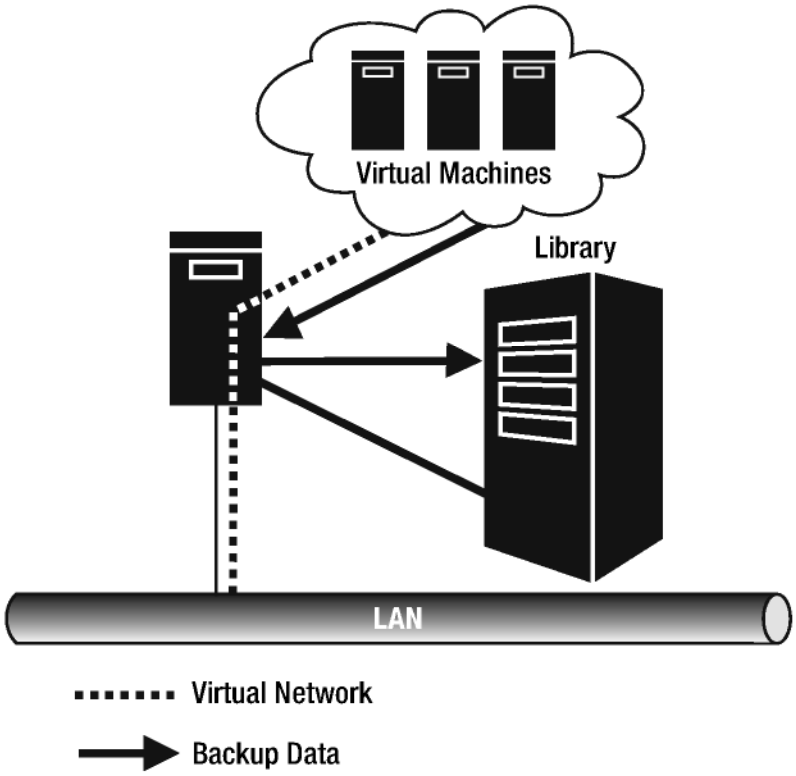


Figure 7-2. *Backing up a VM through the VM host*

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.

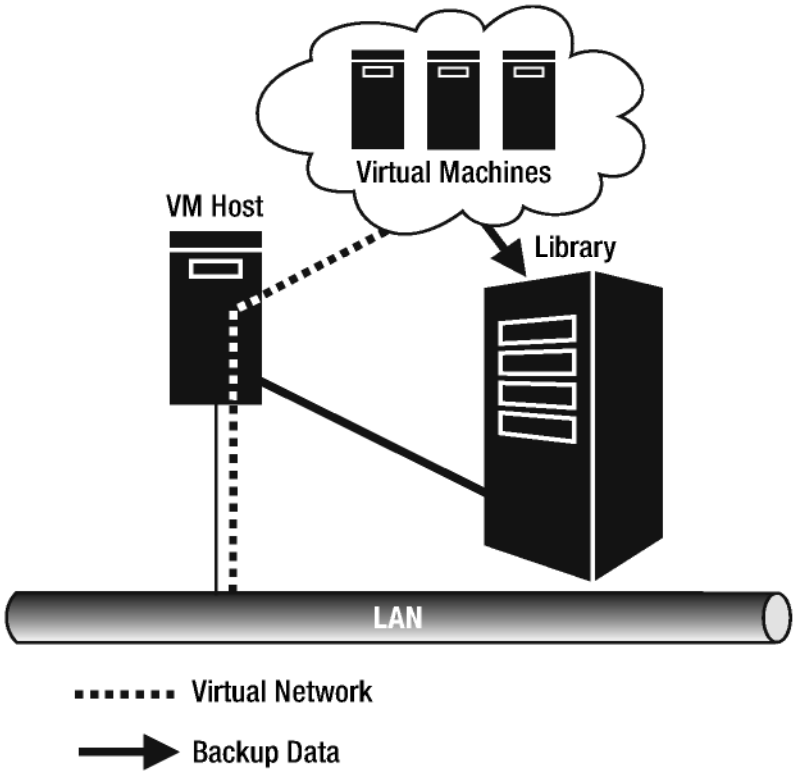


Figure 7-3. *Backing up a VM directly to a storage device*

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.

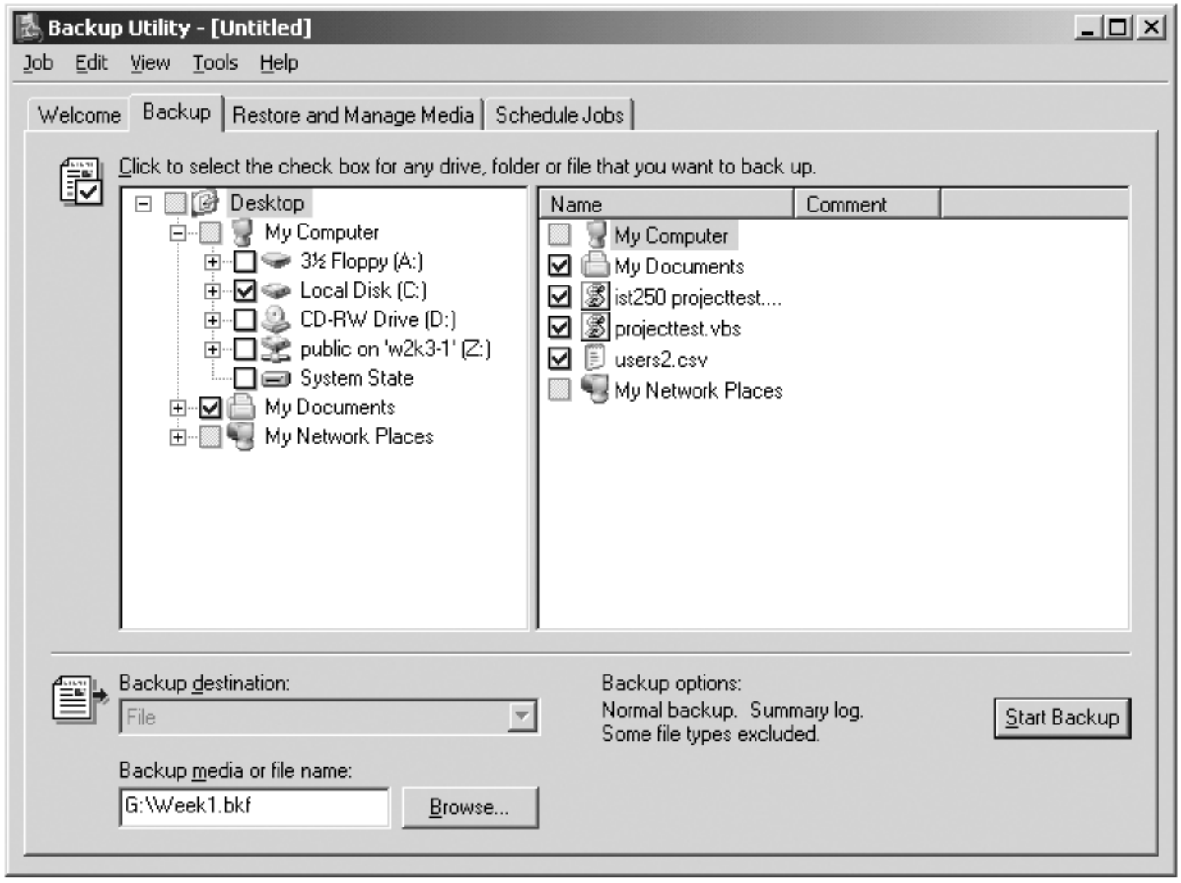


Figure 7-4. *Windows Backup*

■ **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.

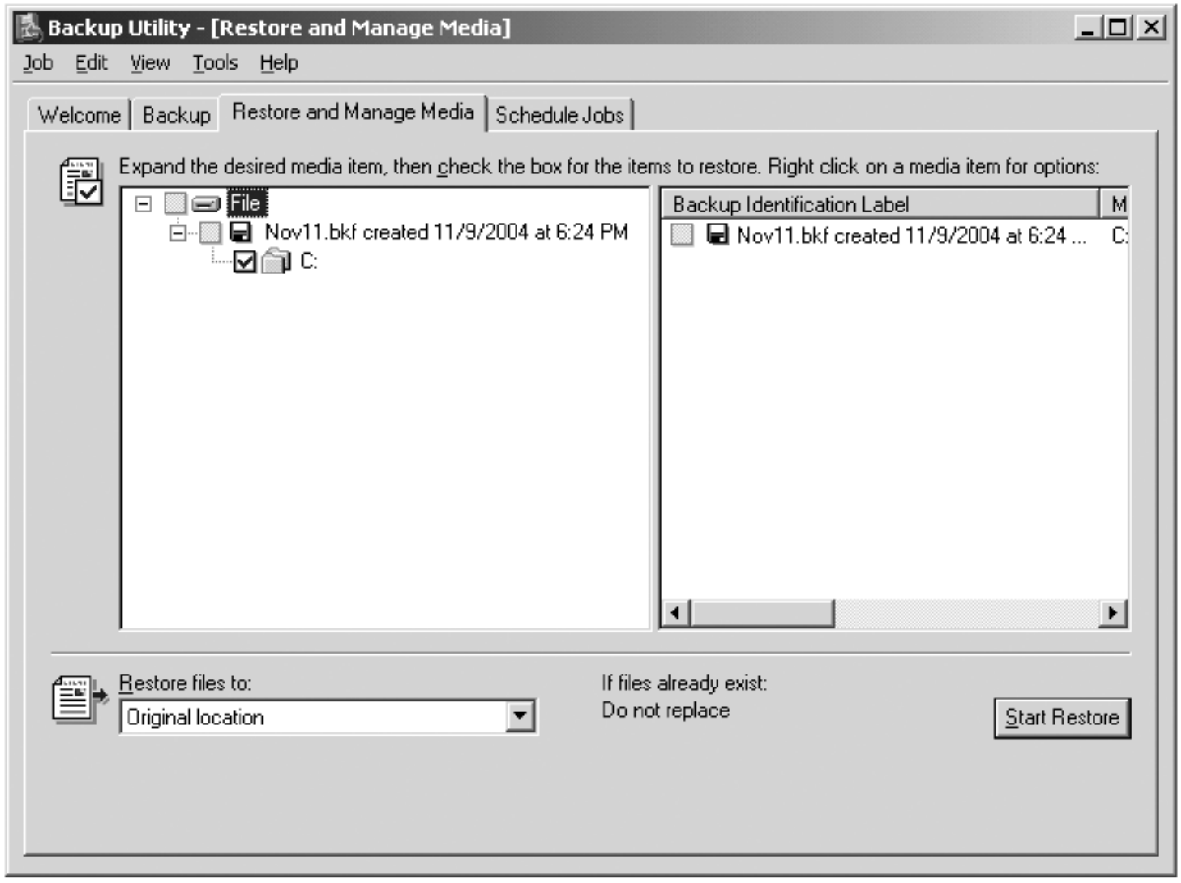


Figure 7-5. *Selecting data to restore using Windows Backup*

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.

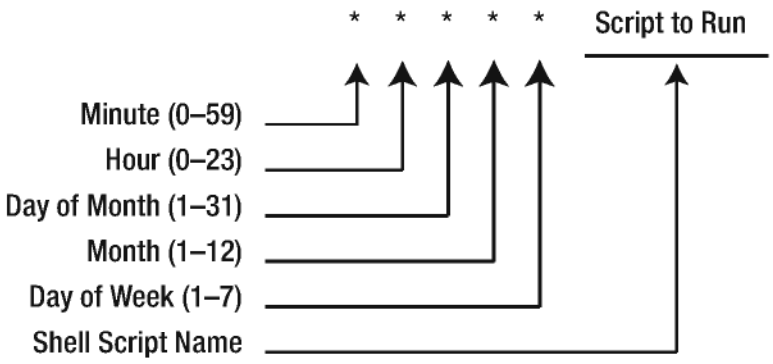


Figure 7-6. *crontab file text format*

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.

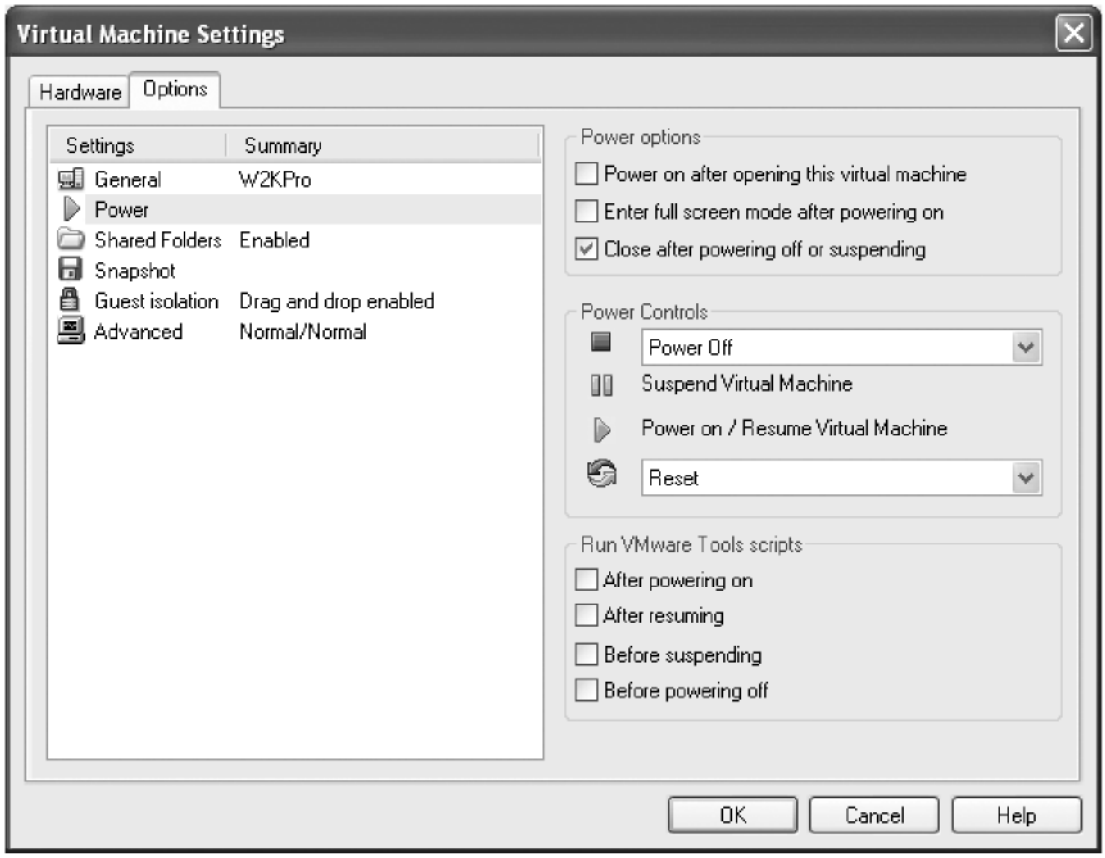


Figure 7-7. *Selecting for the VM to automatically close when powered down*

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 PenguiNet shareware software to perform the remote ssh connection and shutdown command. You can download this tool from `http://www.download.com`. With PenguiNet, 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 PenguiNet, you can create connection profiles using the connection profiles toolbar in the PenguiNet 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.

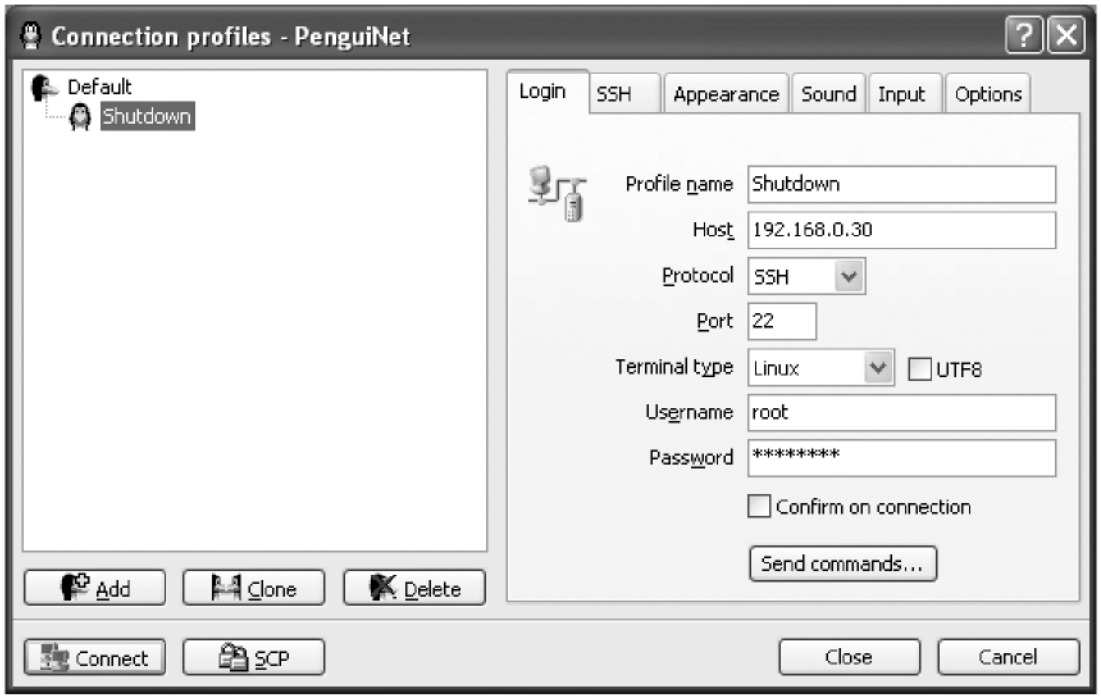


Figure 7-8. *PenguiNet shutdown profile settings*

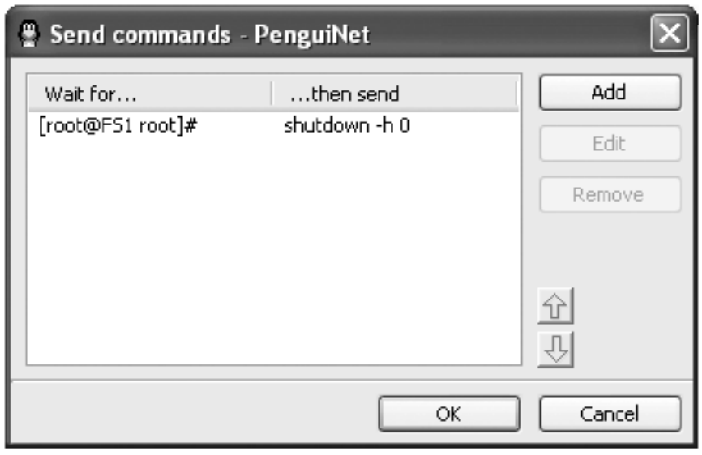


Figure 7-9. *PenguiNet shutdown profile Send Commands settings*

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 PenguiNet 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%-%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 PenguiNet.exe to issue shutdown command
"D:\Program Files\PenguiNet\PenguiNet.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 PenguiNet 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://sshwindows.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.

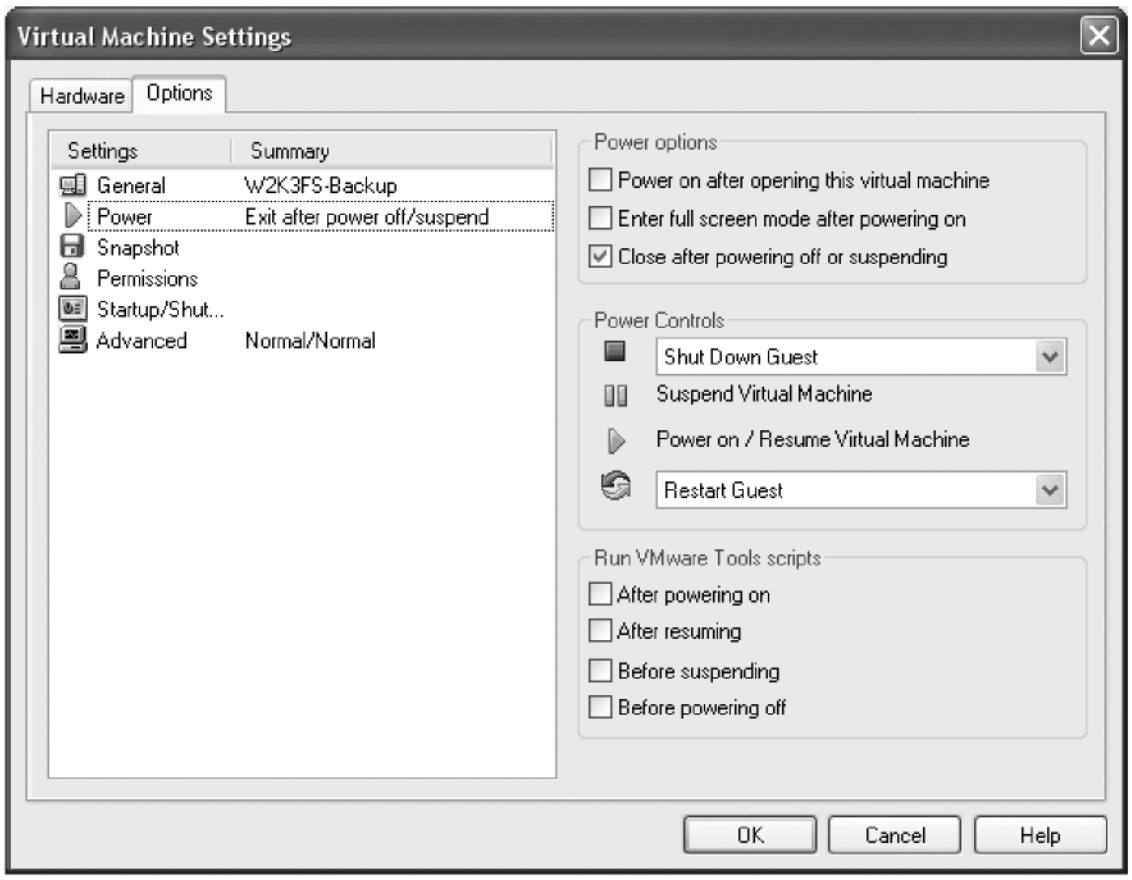


Figure 7-10. *Selecting GSX VM power-down options*

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%-%Date:~7,2%-%Date:~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 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.

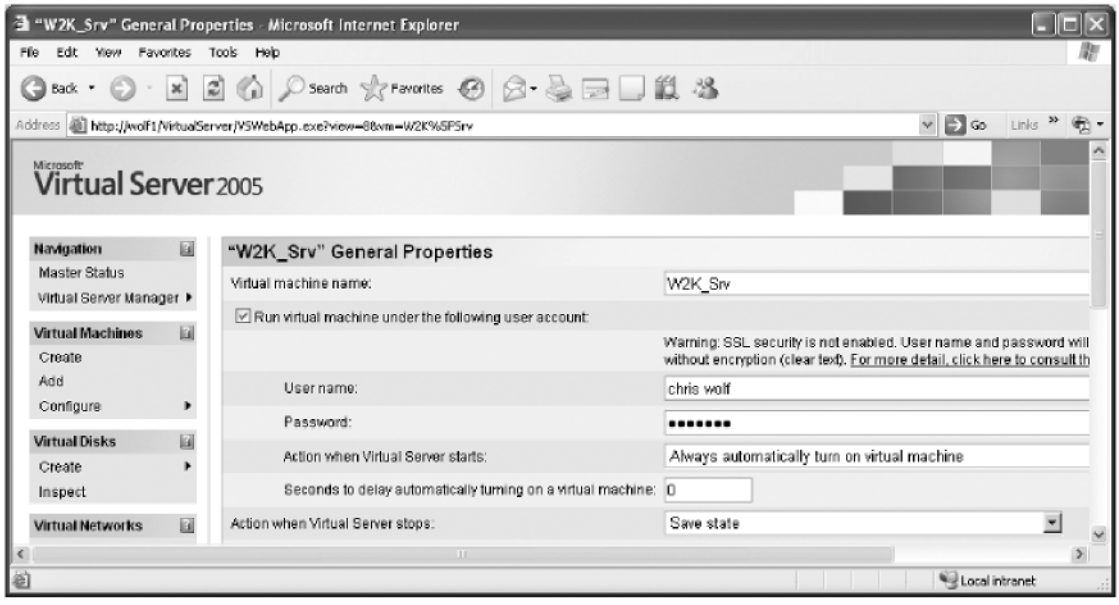


Figure 7-11. *Setting Virtual Server VM start and stop actions*

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: VS2005AllVMBackup.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: VS2005VMBackup.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.

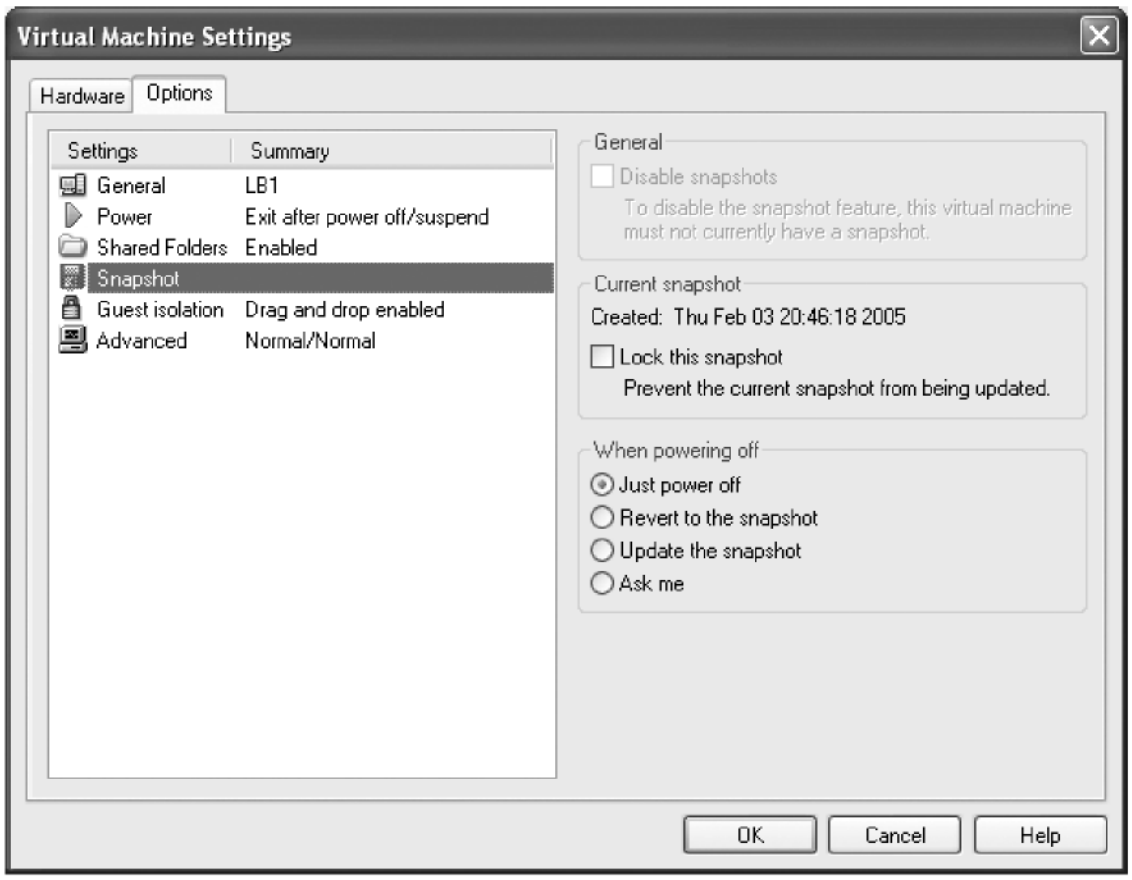


Figure 7-12. *Configuring VM snapshot options*

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.

APPENDIX A

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).

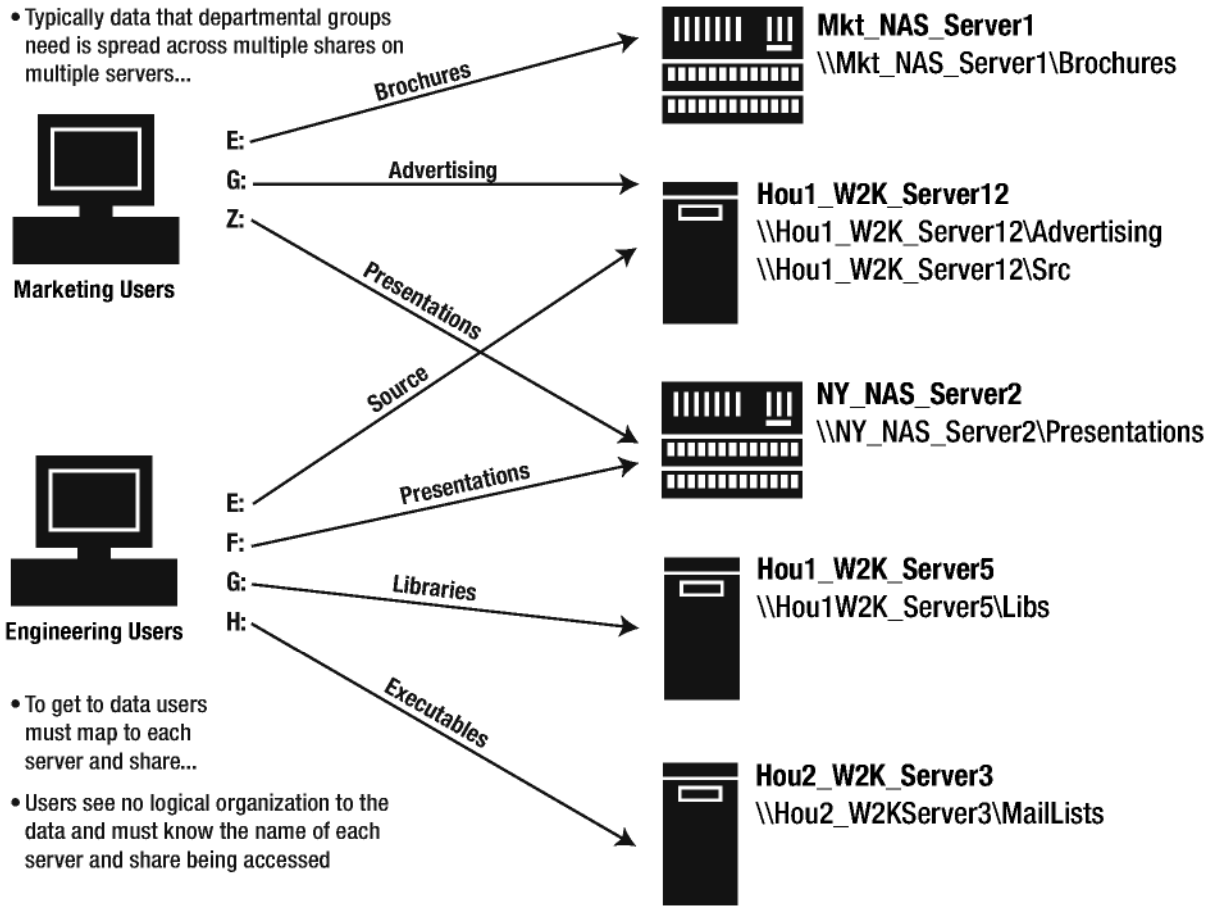


Figure A-1. *Current file system environment*

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.

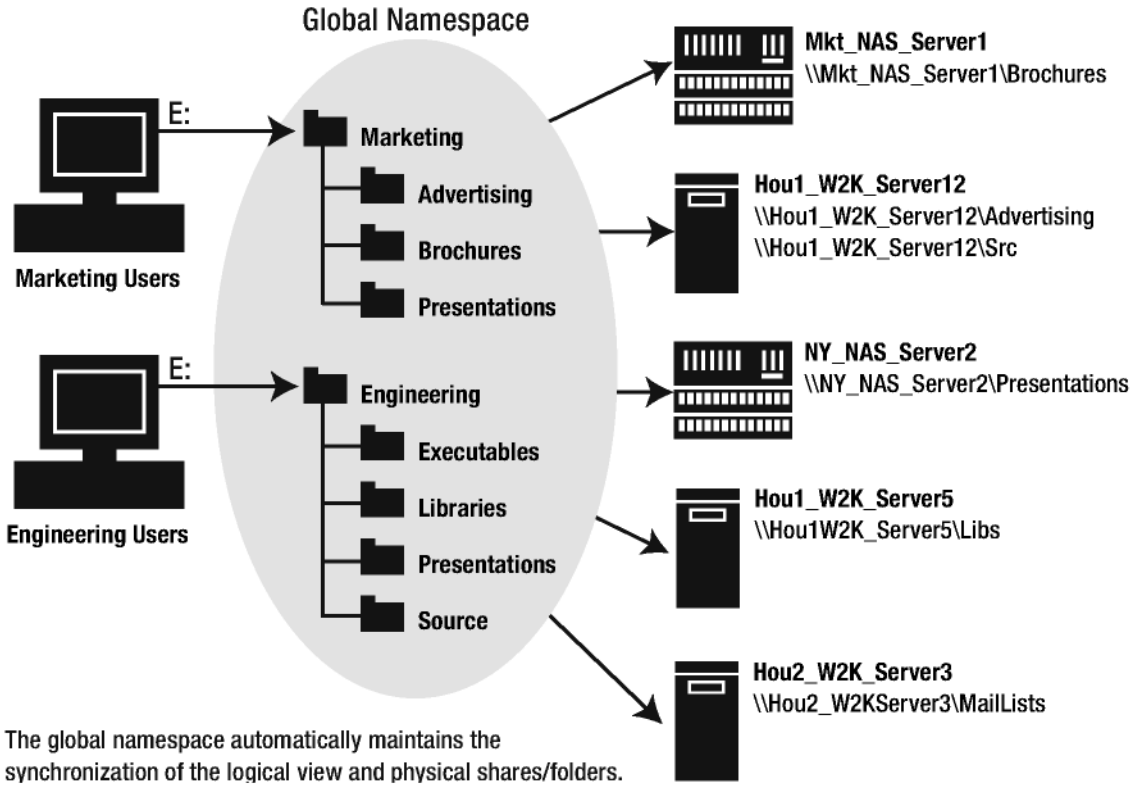


Figure A-2. *File system environment with a global namespace*

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.

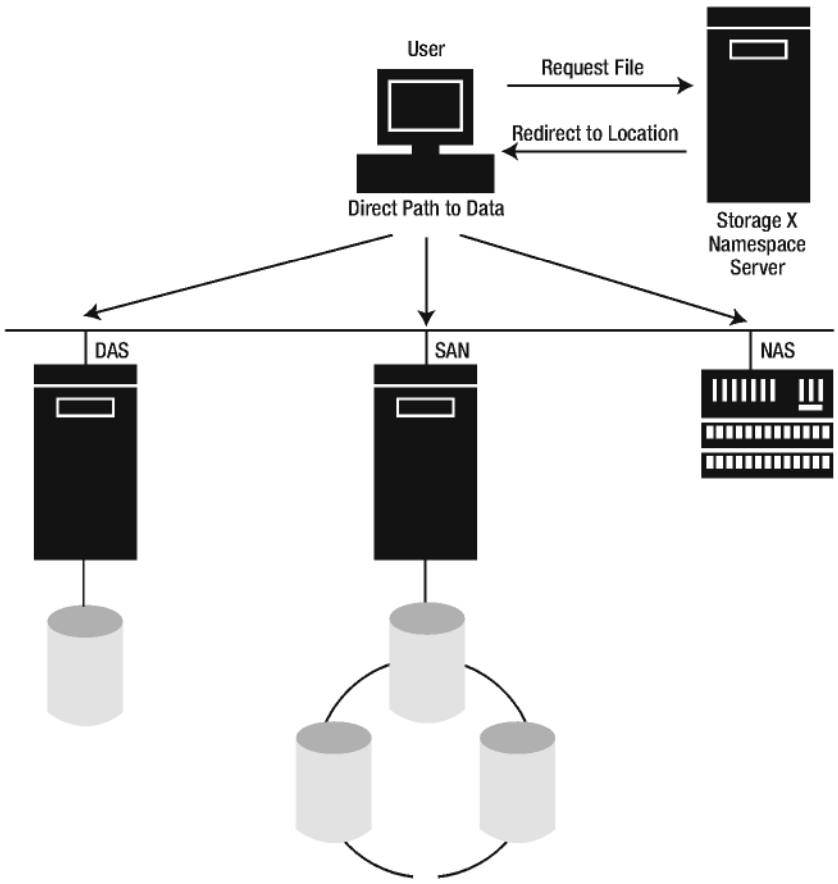


Figure A-3. *Global namespace architecture*

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).

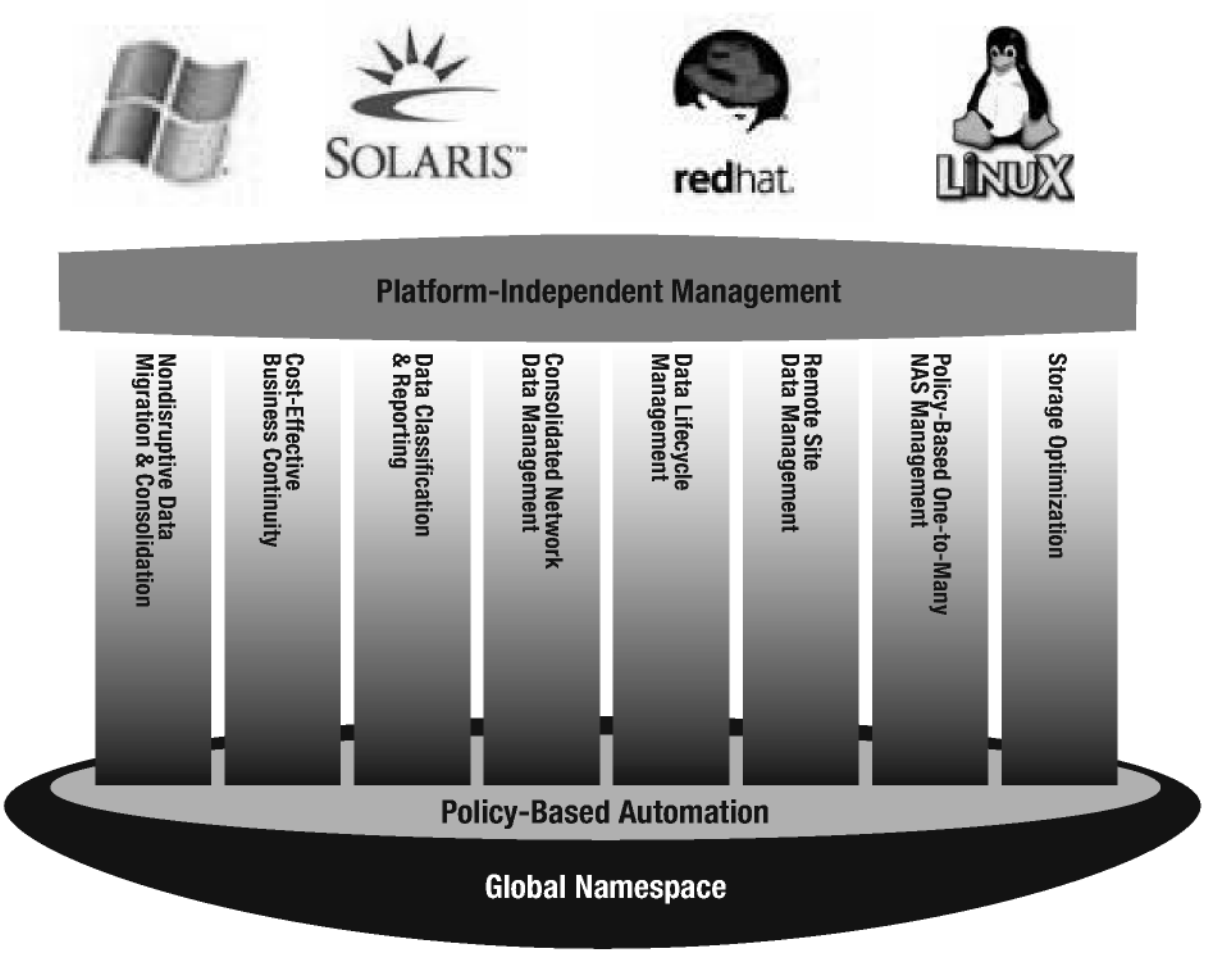


Figure A-4. *StorageX Global Namespace*

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).

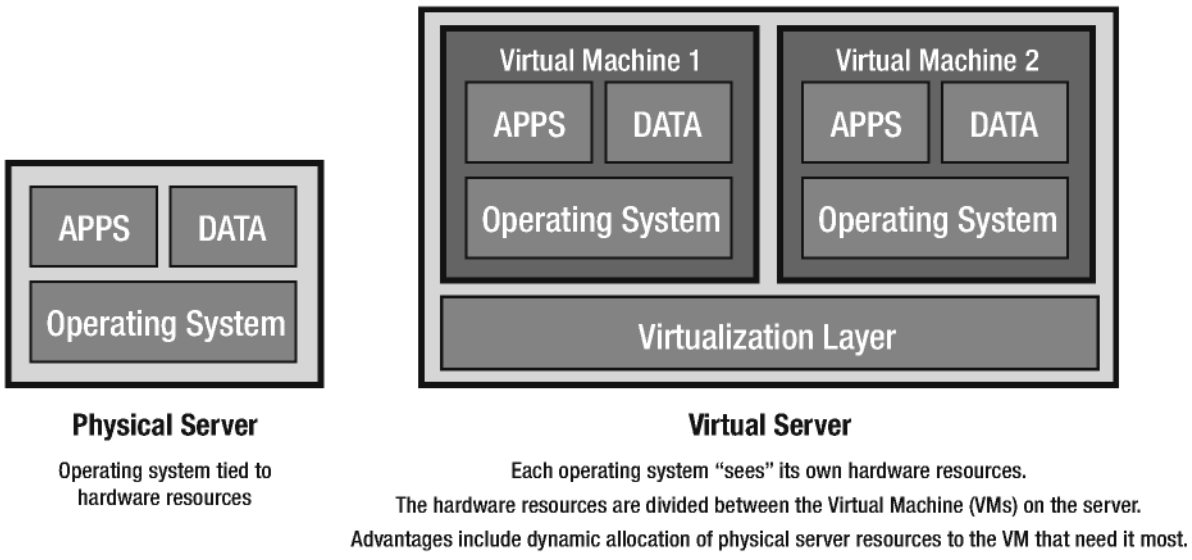


Figure A-5. *Virtual server architecture*

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.

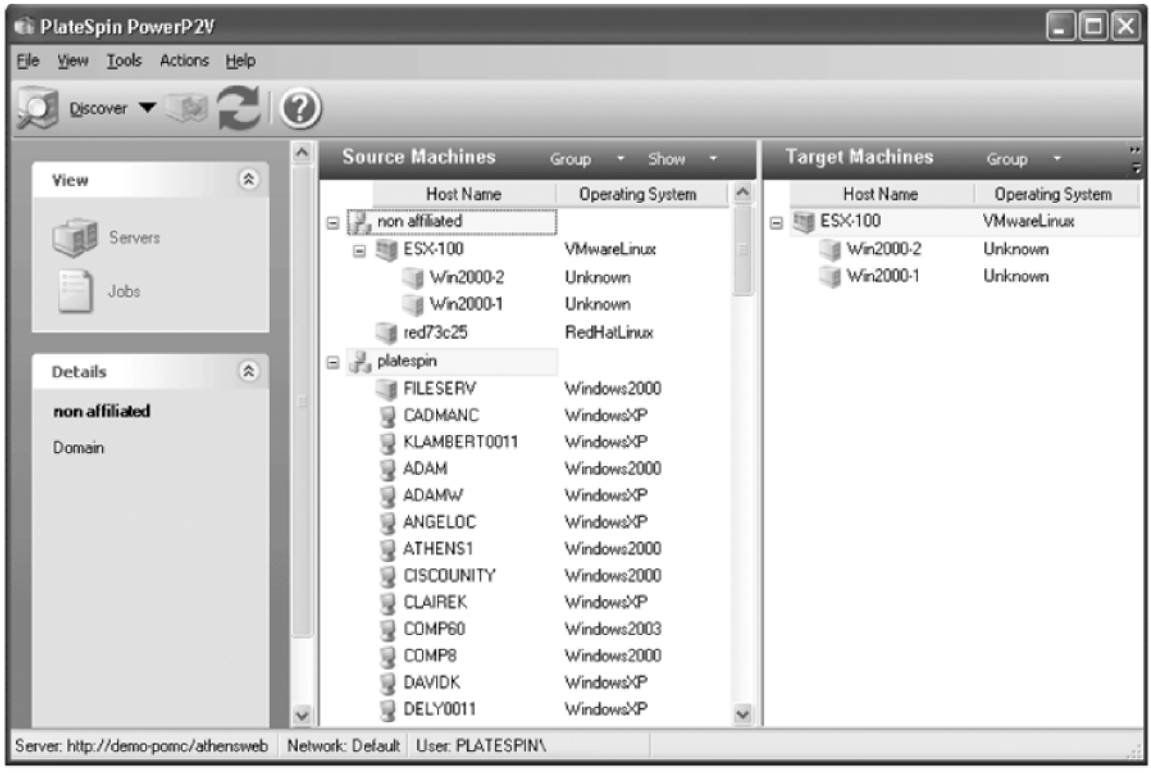


Figure A-6. *Selecting the source machine*

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.

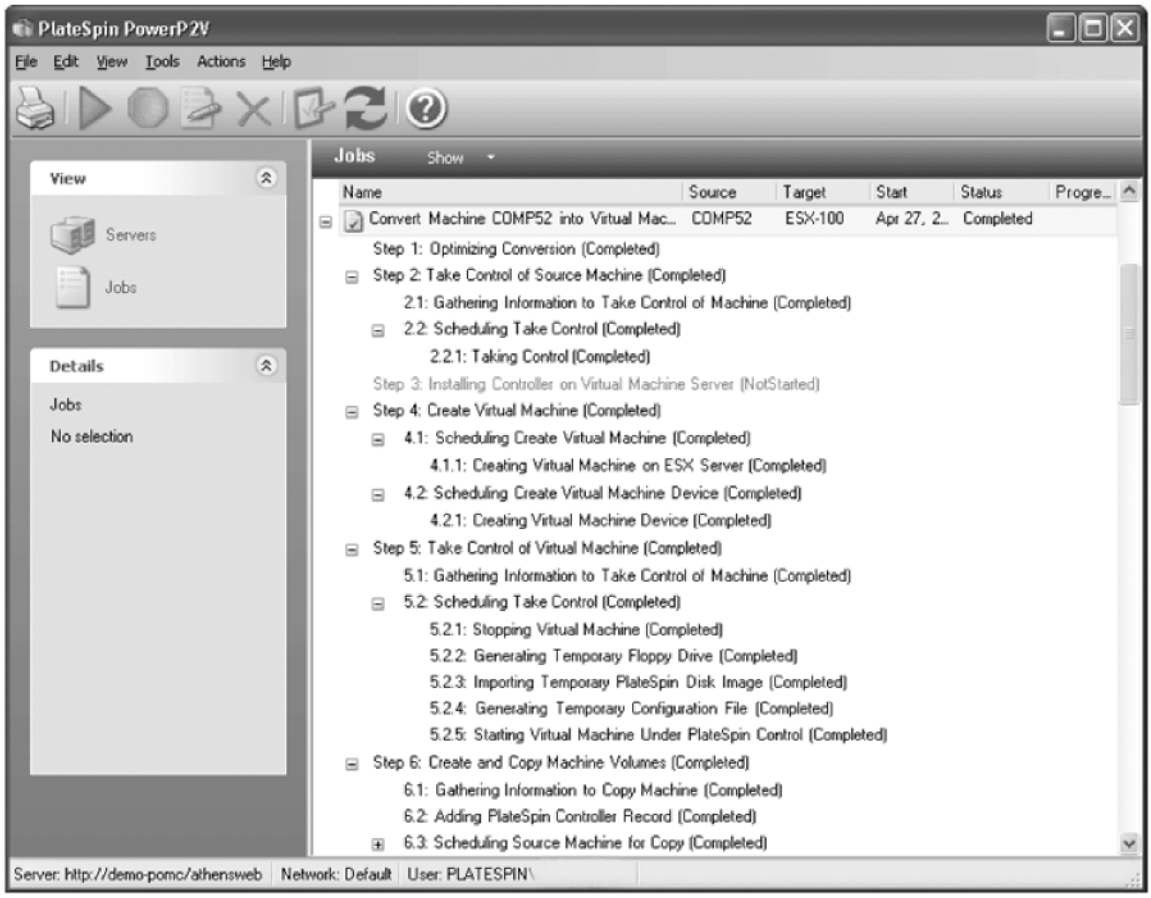


Figure A-8. *Monitoring the conversion jobs*

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).

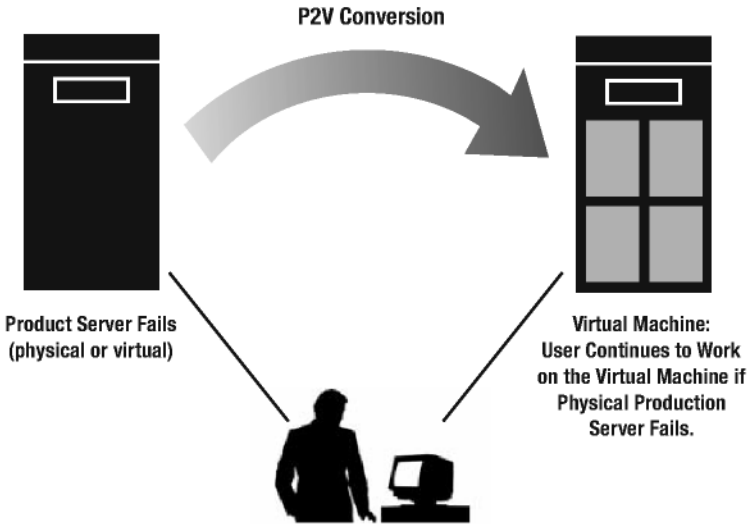


Figure A-9. *P2V conversion*

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).

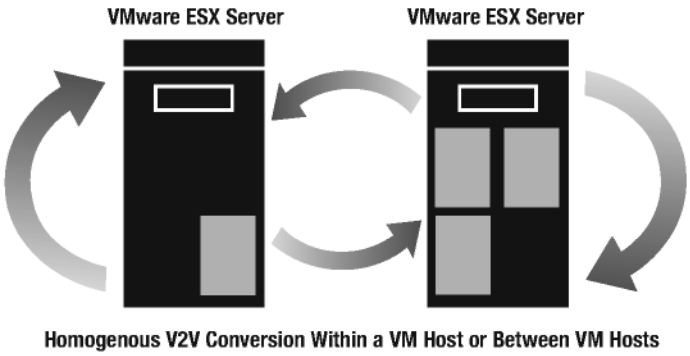


Figure A-10. *Homogeneous V2V conversion*

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).

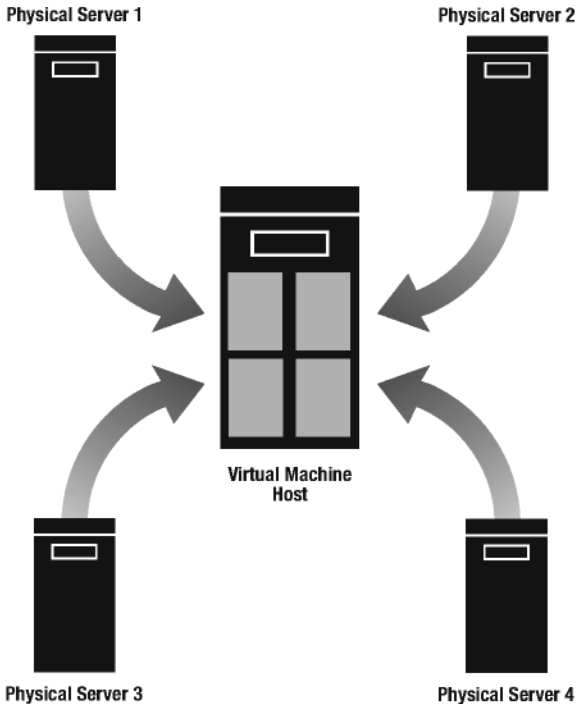


Figure A-11. *Consolidating many underutilized physical servers into one VM host*

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).

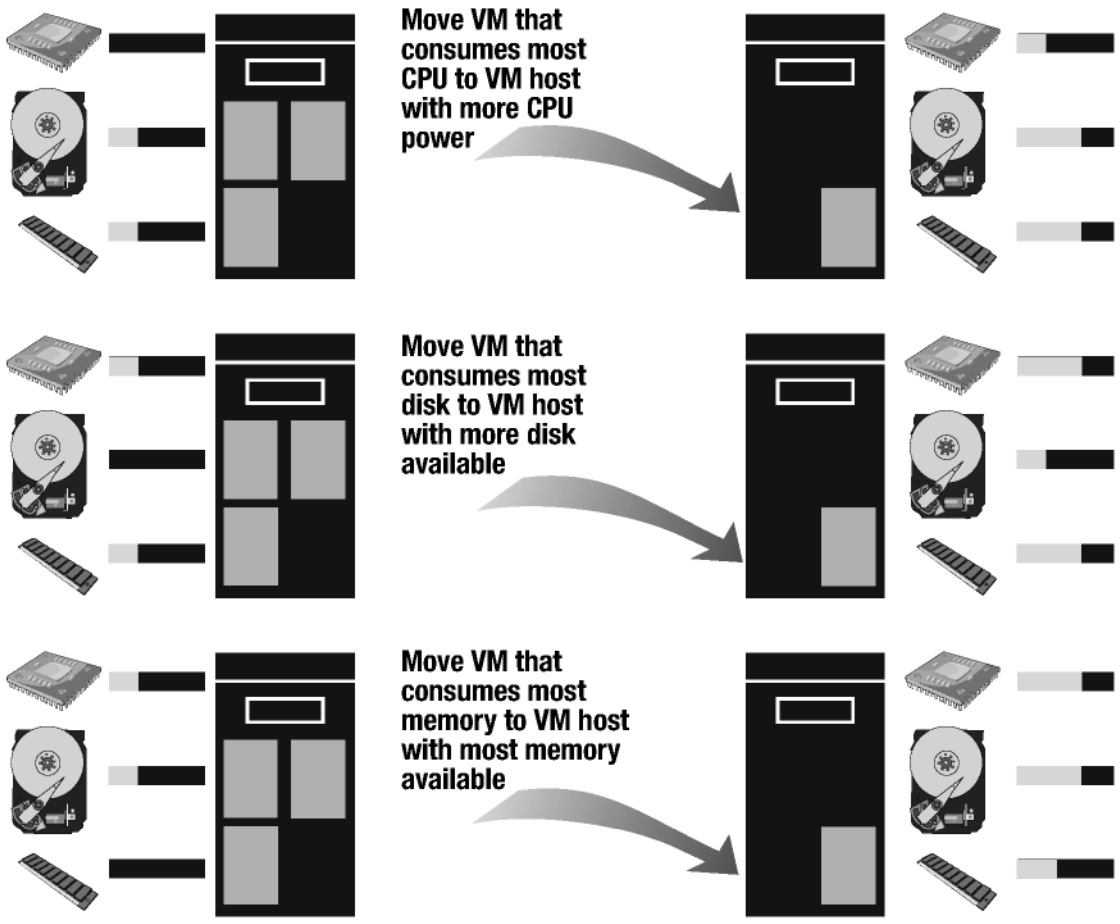


Figure A-12. *Having the ability to move VMs across virtual hosts*

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.

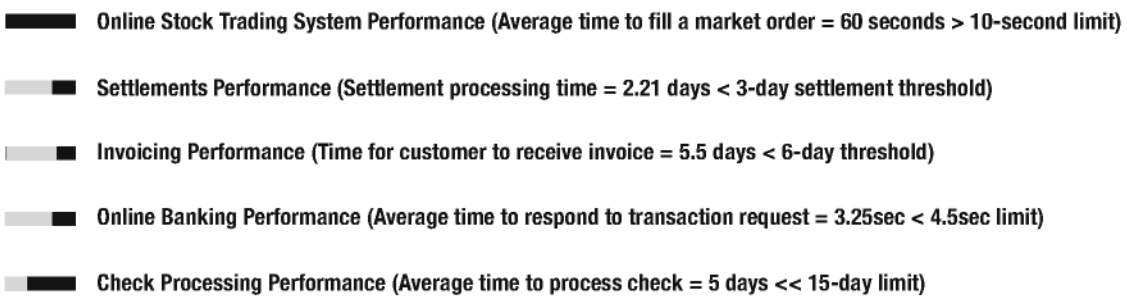


Figure A-13. *Five business services*

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.

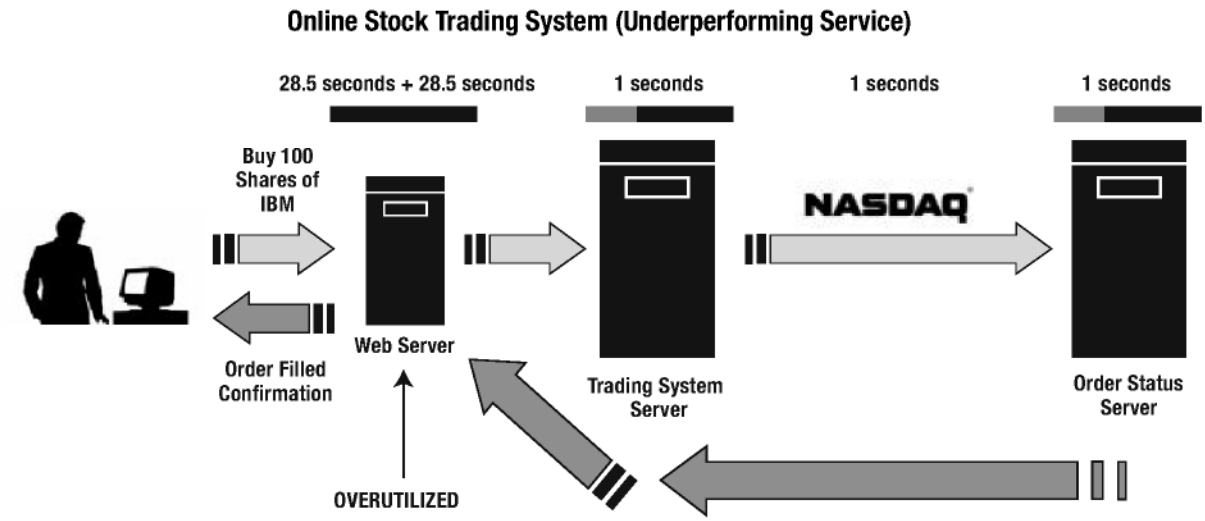


Figure A-14. *Underperforming service*

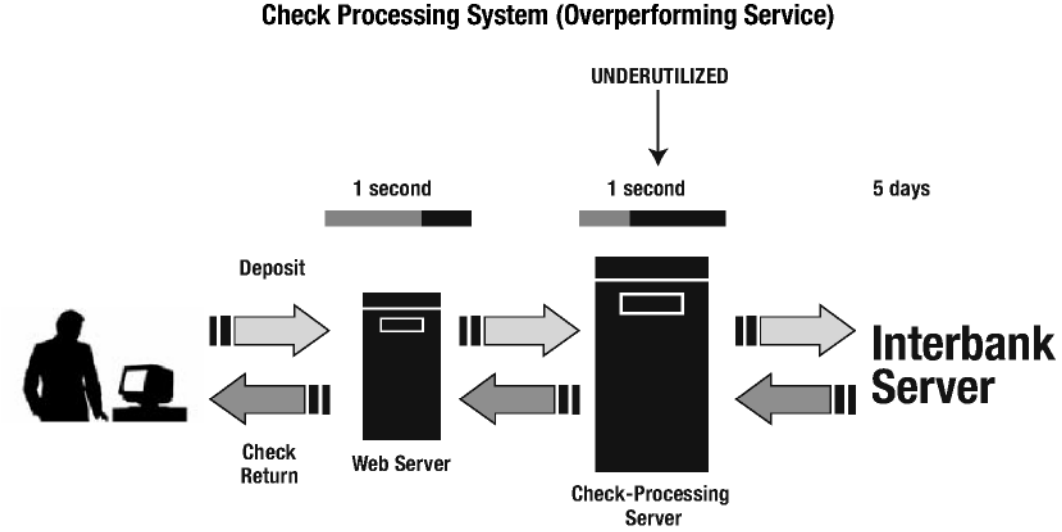


Figure A-15. *Overperforming service*

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).

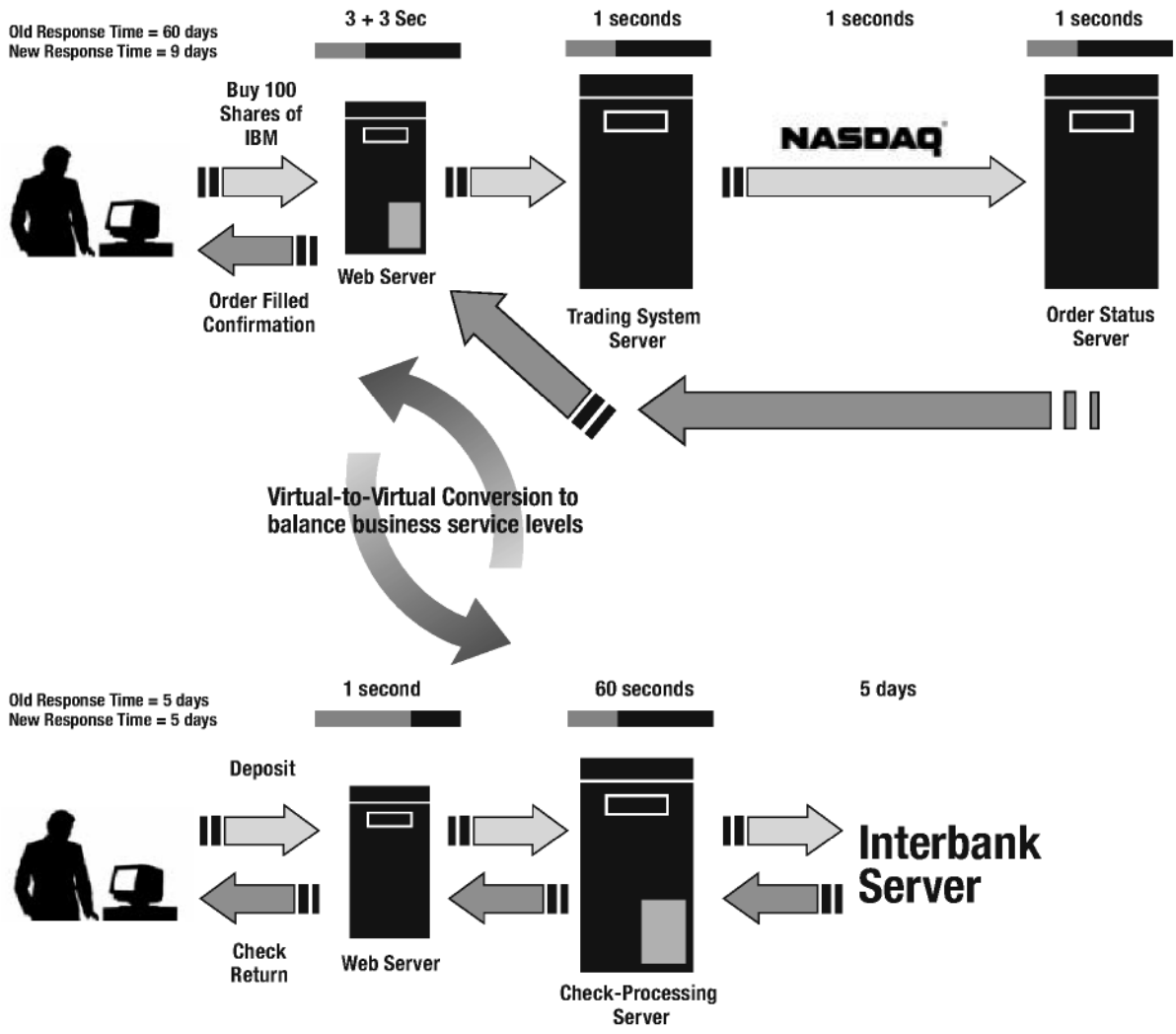


Figure A-16. *V2V conversion causes improvement in business service performance.*

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).
