

Programmer to Programmer™



Professional

VMware® Server

Eric Hammersley



Updates, source code, and Wrox technical support at www.wrox.com

Professional VMware® Server

Eric Hammersley



Wiley Publishing, Inc.

Professional VMware® Server

Eric Hammersley



Wiley Publishing, Inc.

Professional VMware® Server

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2007 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN-13: 978-0-470-07988-1

ISBN-10: 0-470-07988-6

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1B/SR/RR/QW/IN

Library of Congress Cataloging-in-Publication Data:

Hammersley, Eric, 1974-

Professional VMware server / Eric Hammersley.

p. cm.

Includes index.

ISBN-13: 978-0-470-07988-1 (paper/website)

ISBN-10: 0-470-07988-6 (paper/website)

1. VMware. 2. Operating systems (Computers) 3. Virtual computer systems. I. Title.

QA76.76.O63H344637 2007

005.4'3--dc22

2006033472

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, or online at <http://www.wiley.com/go/permissions>.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. VMware is a registered trademark of VMware, Inc. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

*To Sandy, my wonderful wife, and our two children, Chandler and Renae.
Without you this book would never have been possible. Your support and
encouragement helped make this book what it is today. I love you all dearly.*

About the Author

Eric Hammersley has been a technician, systems engineer, project manager, and developer in the IT industry for more than 15 years. He began his career in networking and support in the early '90s installing and supporting hundreds of networks during that time. He then moved into the government sector joining the United States Navy where he remained on active duty through the turn of the century. Although he specialized in shipboard weapons and radar systems, his focus did not shy away from the IT field. Taking charge of the shipboard computer systems he later transitioned into the Integrated Shipboard Network System (ISNS) project—a Navy initiative in the late '90s to integrate the data systems of units afloat to help improve Command, Control, and Communication across the fleet. It was during the ISNS project, just after VMware first released its Workstation product, that he realized the potential of software virtualization. Having to cram racks of servers into an already overcrowded shipboard environment was a wake-up call. A better way had to exist, and VMware was the frontrunner in that technology. From that point forward he became an avid proponent of software virtualization. Throughout the evolution of the VMware product line he has continued his quest to increase the awareness and use of software virtualization by the software developer. He is currently the CIO and Lead Software Architect for The SASSI Institute and resides in Bedford, Indiana. You can contact him at eric@vmcoder.com—a site that focuses on the use of virtualization throughout the software development cycle.

Acknowledgments

None of this could have been possible without the wonderful team at Wrox. I would like to thank Chris Webb, who took a chance and gave me a shot at this. I am forever grateful for the opportunity. A special thanks to Kevin Kent. Without your amazing skills this book would be little more than thoughts on a dinner napkin. You are truly a pleasure to work with. To everyone else at Wrox a heartfelt thanks. Each and every one of you involved helped make this book a success.

A special thanks to Scott Lowe. Your technical expertise and point of view were important parts of this process.

Again, I would like to thank my wonderful family. I would be lost without you. You sacrificed a great deal over the past several months to make this book a reality. Without your love and support this would not have been possible.

I would also like to thank my parents, Larry and Sue. You have always been there for me. Without hesitation you stepped in and helped over the past several months. I love you both.

Credits

Executive Editor

Chris Webb

Senior Development Editor

Kevin Kent

Technical Editor

Scott Lowe

Production Editor

Eric Charbonneau

Copy Editor

Kim Cofer

Editorial Manager

Mary Beth Wakefield

Production Manager

Tim Tate

Vice President and Executive Group Publisher

Richard Swadley

Vice President and Executive Publisher

Joseph B. Wikert

Graphics and Production Specialists

Carrie A. Foster

Barry Offringa

Rashell Smith

Alicia B. South

Quality Control Technicians

John Greenough

Christy Pingleton

Project Coordinator

Erin Smith

Proofreading and Indexing

Techbooks

Anniversary Logo Design

Richard Pacifico

Contents

Acknowledgments	iii
Introduction	xiii
Who This Book Is For	xiii
What This Book Covers	xiv
How This Book Is Structured	xiv
What You Need to Use This Book	xvi
Conventions	xvi
Source Code	xvii
Errata	xvii
p2p.wrox.com	xvii
Chapter 1: VMware Server: A Brief Introduction	1
What Is Virtualization?	1
Software Virtualization	2
Hosted vs. Hypervisor Architecture	3
Terminology, Host vs. Guest	6
So What Is VMware Server?	6
Rapid Deployment	6
Multiple Platforms Available on a Single Physical Machine	7
The Ability to Move Virtual Machines from One Server to Another	7
Increased Utilization of Your Hardware	8
Life without VMware Server	8
Summary	9
Chapter 2: Installing Your VMware Server	11
Host Hardware Requirements	11
Processor	12
Memory	12
Hard Disk	13
Network Controller	14
Host Operating System Requirements	14
Windows	15
Linux	15

Contents

Remote Client Requirements	17
Hardware	17
Operating System	18
Installing VMware Server	18
Installing VMware Server on Windows	19
Installing VMware Server on Linux	24
Install Overview	34
Installing the VMware Server Management Interface on Linux	34
Installing the Remote Client	38
Installing the Remote Client on Windows	38
Installing the Remote Client on Linux	43
Troubleshooting the Install	48
Having to Compile Your Own Kernel Modules	48
Verifying Port 902 Is Free	50
Installing xinetd	50
The Case of the Missing Library: libdb.so.3	51
Launching the Server Console Manually	52
Enabling the MUI and Console Ports in Your Firewall	54
Summary	56
 Chapter 3: VMware Configuration Nuts and Bolts	 57
Getting to Know the Server Console	57
Host Settings and Console Preferences	62
Host Settings	62
Console Preferences	66
Virtual Machine Settings	72
Virtual Hardware Specifications	73
Virtual Machine Options	75
Virtual Networking	84
Automatic Bridging	86
Host Virtual Network Mapping	87
Host Virtual Adapters	91
DHCP and NAT	92
Virtual Machine Security	92
Options for User Access Levels	92
Configuring User Access Levels	93
Summary	96
 Chapter 4: Creating Your First Base Image	 97
What Is a Base Image?	97
Creating Your First Virtual Machine	98

Installing a Guest Operating System	109
Troubleshooting Guest Operating System Installation	112
Installing VMware Tools on Windows Guests	113
Installing VMware Tools on Linux Guests	113
Freezing Your Windows Guest with Sysprep	116
Summary	123
Chapter 5: Where Does This Fit into Your Development Environment?	125
Virtual Machines for Software Development	126
Virtual Machines for Software Testing	128
Platform Testing with Snapshots	129
Platform Testing with Persistent and Nonpersistent Disks	130
Virtual Machines for Post-Release Maintenance	131
Creating Base Images for Your Development Environment	131
Planning	132
Collaborating	133
Creating	133
Testing	134
Deploying	135
Summary	136
Chapter 6: Bringing It All Together	137
Organizing Your Image Library	137
Proper Image Naming	137
Archiving Your Images	140
Preparing Base Images for Use	140
Recycling Your Testing Images	143
Creating an ISO Image Library	144
Creating an ISO Image	145
Using an ISO Image	146
Summary	148
Chapter 7: Introduction to the Scripting APIs	151
What Can You Automate?	152
Supported Platforms	153
VmPerl	153
VmCOM	153
Programming API	154
VmPerl vs. VmCOM vs. Vix	154
The vmware-cmd Utility	156

Contents

The vmrun Utility	157
The VMware SDK	157
Summary	157
Chapter 8: Using VmCOM	159
A Quick OOP Primer	159
Windows Scripting Host	160
Objects	161
VmConnectParams	161
VmServerCtl	162
VmCtl	164
VmCollection	171
VmQuestion	173
Enumerations	177
VmExecutionState	177
VmPowerOpMode	178
VmProdInfoType	179
VmProduct	179
VmPlatform	179
Summary	180
Chapter 9: Using VmPerl	183
Cross-Platform Perl Scripts	183
Modules	185
VMware::VmPerl::ConnectParams	185
VMware::VmPerl::Server	187
VMware::VmPerl::VM	191
VMware::VmPerl::Question	199
Constants	204
VM_EXECUTION_STATE	204
VM_POWEROP_MODE	204
Infotype	205
VM_PRODINFO_PRODUCT	206
VM_PRODINFO_PLATFORM	206
Summary	207
Chapter 10: Using the Programming API	209
Compiling Against Vix with Visual C++	209
Setting Up the Project	210
Including the include Statement	215

Pointing the Project to the Header Files	216
Pointing the Project to the Libraries Used	218
A Quick Vix Primer	220
Handles	220
Handle Types	222
Handle Properties	222
Error Codes	225
Asynchronous Operations and the Job Handle	226
Functions	230
Vix_ReleaseHandle	231
Vix_GetProperties	231
Vix_GetPropertyType	232
Vix_GetHandleType	233
Vix_GetErrorText	234
Vix_FreeBuffer	235
Vix_CreateTemporaryObject	236
Vix_PumpEvents	236
Host Machine Operations	237
VixHost_Connect	237
VixHost_Disconnect	239
VixHost_FindItems	240
VixHost_RegisterVM	243
VixHost_UnregisterVM	244
Virtual Machine Operations	245
VixVM_Open	245
VixVM_PowerOn	246
VixVM_PowerOff	247
VixVM_Reset	248
VixVM_Suspend	249
VixVM_Delete	250
VixVM_CreateSnapshot	251
VixVM_GetNumRootSnapshots	253
VixVM_GetRootSnapshot	254
VixVM_RemoveSnapshot	255
VixVm_RevertToSnapshot	256
VixVM_InstallTools	258
VixVM_WaitForToolsInGuest	259
VixVM_UpgradeVirtualHardware	261
VixVM_LoginInGuest	262
VixVM_RunProgramInGuest	263
VixVM_CopyFileFromHostToGuest	265
VixVM_CopyFileFromGuestToHost	266

Contents

Job Handle Operations	268
VixJob_Wait	268
VixJob_GetError	268
VixJob_CheckCompletion	269
Using Vix — A Few Brief Examples	269
Summary	275
Chapter 11: Integrating with Your Existing Environment	277
Continuous Integration	278
VMware Tools Scripting	279
Scripting VMware Tools with VBScript	279
Scripting VMware Tools with C# and VB.NET	283
CruiseControl.NET	285
Basic Configuration	285
VMware Server Integration	299
Visual Studio Team System	312
Creating a Team Project	312
Adding a Development Project to the Team Project	318
Configuring a Team Build	322
Testing the Build	328
VMware Server Integration	330
VMware Server Integration under Linux — A Brief Look	349
Summary	349
Chapter 12: A Real-World Scenario	351
Defining Your Environment	351
Target Platform	352
Supported Platforms	352
Tools	352
Source and Process Control	353
Defining Your Goals	353
Setting Up Your Development Images	354
Setting up the Virtual Development Machine	355
Deleting the Virtual Development Machine Floppy Drive	365
Renaming the Virtual Development Machine Configuration File	365
Setting up the Virtual Development Machine Software for Deployment	367
Setting Up Your Testing Images	374
Installing the VMware Tools Script to Retrieve the Guest IP Address	375
Configuring Your Testing Images as Nonpersistent	378

Contents

Configuring Build Integration	380
Project Setup	381
VMware Server Integration	394
The Final Scripts	403
Summary	408
Appendix A: VmCOM Error Codes	409
Appendix B: VmPerl Error Codes	413
Appendix C: Vix Error Codes	417
Index	423

Introduction

Software virtualization is not a new concept; however, it is only recently that it has begun to gain momentum by the masses. This movement can be attributed in large part to the innovations of VMware, a company that has specialized in virtualization technology since the late '90s.

Early on, the release of VMware Workstation was used mainly by advanced users and hobbyists. This was due in large part to the lack of understanding of the technology as well as the progression of the industry as a whole. This quickly changed, however, as the technology matured into not only a more stable solution but one that the masses could easily utilize. Now it is not uncommon to find even the most basic users dual-booting their workstations, wanting to try out Linux or some other operating system, or just having a clean machine to play with. This move triggered the recognition of VMware Workstation as more than just a specialist's tool. With it, even a beginner can configure and operate a virtual machine with little or no previous experience.

This boom also gave way to new and innovative ways for greater utilization in the data center, giving companies from small to large the ability to increase their return on the investment of hardware and flourish. It doesn't stop there, however. The technology offers great advantages to the software developer and tester alike. It's such a perfect fit you'd think it was its sole purpose in life. Sadly, this seems to be the tail runner in the virtualization boom. Developers are still writing code on their office workstations and testing in rooms that contain dozens of individual PCs. Why?

This book focuses on VMware Server, a now free server-based version of VMware's popular virtualization line. Throughout this book I discuss not only the benefits of software development and testing within virtual machines, but the advantages they offer in terms of automation, stability, disaster recovery, and overall code quality.

Who This Book Is For

This book is designed to not only teach you how to use VMware Server but how to automate its function through the use of its APIs.

A few specific categories of readers stand to gain the most from this book:

- ❑ **Software Developers**—The software developer is one of the main players here. Every day they sit down either as an individual or as a team and crank out the software we all use on a daily basis. Without them, the industry wouldn't exist. This book introduces the software developer to the benefits of virtual machines and what they can offer. For the developer that already has an understanding of their benefits, this book takes them into more advanced topics, such as base images, test banks, and automation.

Introduction

- ❑ **Lead Developers**—Though this title may not differ a whole lot from the software developer itself, the roles are greatly different. This book aims to provide the software development team leads out there with the information required to make the move into team development within virtual machines. It discusses development images, enforcing standards on your development team, and project management beyond the development cycle—specifically, how software development within virtual machines allows for archival and later reference of exact development environments long after a product is released.
- ❑ **Software Testing Engineers**—Some of the real meat of this book centers around the automation available to increase the exposure your applications receive during their testing phase. You will learn how to integrate VMware Server into your existing systems to offer automated builds and testing on banks of virtual machines.

What This Book Covers

This book covers VMware Server, a free server-based virtualization product from VMware. It does not, however, stop there. A serious amount of attention is also provided to the legacy programming APIs, VmPerl and VmCOM, as well as the newest API, the Programming API. This book guides you through creating a fleet of virtual machine images that can be used for development or testing purposes, as well as how to organize them. It also gives heavy focus on the automation of VMware Server and its integration into your existing process management tools like CruiseControl.Net and Visual Studio Team System.

The first few chapters encompass the basics of virtualization and its place within the development and testing environment. It then quickly moves into the APIs and provides a deep, example-rich reference to their function and purpose. The final chapters walk you through integrating VMware Server into your existing environment using CruiseControl.NET and Visual Studio Team System as examples, as well as provide a more “real-world” type example to help tighten the concept up.

Although this book is designed around VMware Server, because it is a free product there is nothing that keeps it from being equally applicable to other VMware products like ESX Server.

How This Book Is Structured

The goal is a front-to-back approach to introducing virtualization into your development process. The already familiar reader can easily skim the first chapter; however, starting right in at Chapter 2 I get down and dirty with VMware Server and start outlining just how you can set it up for your development process. The API chapters, Chapters 8, 9, and 10, offer an easy reference to the API functions and samples of their use. They should provide a handy desk reference for some time to come.

- ❑ **Chapter 1, “VMware Server: A Brief Introduction”**—This chapter introduces virtualization as a whole and provides some insight into the technology, its evolution, and its uses.
- ❑ **Chapter 2, “Installing Your VMware Server”**—This chapter walks you through installing VMware Server, its requirements, and its individual pieces. This chapter also reviews the hardware and software requirements at the time of writing as well as the virtual hardware specifications your virtual machines will utilize.

- ❑ **Chapter 3, “VMware Configuration Nuts and Bolts”**—This chapter really dives into the different settings available with the VMware Server product. From the host settings to virtual machines settings, this is an all-encompassing look into the product itself. It also discusses virtual networking and virtual machine security in a networked environment.
- ❑ **Chapter 4, “Creating Your First Base Image”**—This chapter covers the base image and its creation. A base image is essentially a virtual machine configured with an operating system installed from which you will base all other virtual machines. Topics include creating the virtual machine, installing the operating system, and freezing the install with Sysprep, a Windows-based installation tool.
- ❑ **Chapter 5, “Where Does This Fit into Your Development Environment?”**—This chapter begins the trek into virtual machines for software development and testing. Here I discuss their use in the development process as well as some good tips to follow when creating your development base images for your team. I also discuss post-release maintenance and methods to introduce virtual machine development to your team.
- ❑ **Chapter 6, “Bringing It All Together”**—This chapter brings Chapters 4 and 5 together and explains the methods in which you can organize your image libraries and prepare them for use, and organize your testing images and ISO image libraries.
- ❑ **Chapter 7, “Introduction to the Scripting APIs”**—This chapter introduces the APIs available for VMware Server. Here I discuss their supported platforms and the pros and cons of each. This chapter is a lead into the reference chapters ahead.
- ❑ **Chapter 8, “Using VmCOM”**—This is the first of three API reference chapters. Here I introduce the VmCOM API and its functionality. I review the specific objects and methods available as well as provide example code to help you leverage the VmCOM API in production.
- ❑ **Chapter 9, “Using VmPerl”**—This chapter introduces the VmPerl API. Here I discuss its use, functionality, and flexibility within the product line. Topics include a module reference complete with their corresponding objects and methods as well as example code to help you leverage it in real life.
- ❑ **Chapter 10, “Using the Programming API”**—This is a rather lengthy look into the newest API offering from VMware, the Programming API, or Vix. I set up a development environment for your scripts utilizing C++, which is currently the only language supported, and provide a primer of sorts into Vix and its methodology. The Vix API contains a great deal more functionality than the previous two API offerings and therefore is a bit more complex; however, once it matures, it will be the de facto standard in VMware automation.
- ❑ **Chapter 11, “Integrating with Your Existing Environment”**—This chapter takes on two common integration packages, CruiseControl.NET and Visual Studio Team System. Here I outline how to integrate VMware Server into those environments and automate its functionality. The main goal is to automate builds, test bank virtual machine allocation, and the pushing of test builds up to those machines.
- ❑ **Chapter 12, “A Real-World Scenario”**—This chapter brings everything together in a real-world scenario of sorts. Here I take CruiseControl.NET, a popular free continuous integration package, and integrate it into an existing development environment with specific needs and specifications. I’ll show how to leverage VMware Server to provide banks of testing machines on demand all through a nightly build script.
- ❑ **Appendixes A, B, and C, “Error Codes”**—To wrap things up, the appendixes outline the various error codes by API and provide a quick reference to their use, trapping, and identification.

What You Need to Use This Book

You will need at the very least a PC with an approved operating system for VMware Server as well as the VMware Server software itself. The fact that VMware Server is free makes it a lot easier.

Chapter 2 outlines the approved server-based operating systems as well as the installation procedure for VMware Server.

The coding examples and scenarios for the VmPerl and VmCOM APIs require no additional software support. They can be compiled and run either natively by the operating system itself or by the Perl interpreter provided by VMware Server. The Programming API examples are written in Visual C++ 2005. They can, however, be compiled using the command-line compiler provided by the .NET Framework.

Where appropriate, Linux examples are provided. Two distributions are utilized in this book; however, any can be used that fall within the specifications provided for VMware Server. Specifically, I use Red Hat Enterprise Linux and Fedora Core. The versions used are pointed out in their respective sections.

The final two chapters that deal with integration require either CruiseControl.NET, a free continuous integration product, or Visual Studio Team System (VSTS). Scenarios are provided utilizing both; however, if you do not have access to VSTS, there's still a great deal of value to be had. The final chapter, the real-world scenario, is written against CruiseControl.NET. This was done to keep all the readers, regardless of their access to VSTS, in the loop.

Conventions

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.

Boxes like this one hold important, not-to-be forgotten-information that is directly relevant to the surrounding text.

Tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.

As for styles in the text:

- ☐ We *highlight* new terms and important words when we introduce them.
- ☐ We show keyboard strokes like this: Ctrl+A.
- ☐ We show file names, URLs, and code within the text like so: `persistence.properties`.
- ☐ We present code in two different ways:

In code examples we highlight new and important code with a gray background.

The gray highlighting is not used for code that's less important in the present context, or has been shown before.

Please also note that in cases where a continuous code line has been broken due to the constraints of the margins of the book, you will see the ↵ symbol indicating that the code line continues onto the following line.

Source Code

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All of the source code used in this book is available for download at <http://www.wrox.com>. Once at the site, simply locate the book's title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book's detail page to obtain all the source code for the book.

Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 0-470-07988-6 (changing to 978-0-470-07988-1 as the new industry-wide 13-digit ISBN numbering system is phased in by January 2007).

Once you download the code, just decompress it with your favorite compression tool. Alternatively, you can go to the main Wrox code download page at <http://www.wrox.com/dynamic/books/download.aspx> to see the code available for this book and all other Wrox books.

Errata

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata you may save another reader hours of frustration and at the same time you will be helping us provide even higher quality information.

To find the errata page for this book, go to <http://www.wrox.com> and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors. A complete book list including links to each book's errata is also available at www.wrox.com/misc-pages/booklist.shtml.

If you don't spot "your" error on the Book Errata page, go to www.wrox.com/contact/techsupport.shtml and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

p2p.wrox.com

For author and peer discussion, join the P2P forums at p2p.wrox.com. The forums are a Web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

Introduction

At <http://p2p.wrox.com> you will find a number of different forums that will help you not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to p2p.wrox.com and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you wish to provide and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

You can read messages in the forums without joining P2P; but in order to post your own messages, you must join.

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.

1

VMware Server: A Brief Introduction

VMware has been on the cutting edge of virtualization since the release of VMware Workstation in 1999. In 2001 VMware released its first of a long line of server-based products. The VMware ESX and GSX Server products have changed the way we look at our server utilization to this day. There are now dozens of virtualization software packages on the market with price points that are all over the map. This increase in competition gave birth to VMware Server, the next iteration of its GSX product line. What makes VMware Server so special? As you will see over the course of this book, VMware Server is a robust and versatile tool for your data center and development process. Oh, did I mention that it was free?

This chapter introduces the basic concepts of hardware virtualization, defines some early terminology, introduces you to VMware Server, and gives you a look at what life was like before virtualization.

What Is Virtualization?

Virtualization itself is a rather broad term that most commonly refers to some sort of abstraction of resources. Throughout this book I will be focusing on one specific aspect of virtualization, software virtualization.

Software Virtualization

Software virtualization is a process in which the actual physical hardware of a machine is decoupled, or abstracted if you will, from the underlying operating system by means of software. This hardware abstraction allows multiple “virtual” environments to function side by side on top of a single physical machine. Through this abstraction process each virtual machine is provided with its own hardware (RAM, CPU, Network Card, and so on) that is independent and isolated from the other virtual machines running along side it.

Figure 1-1 shows how a normal x86 machine presents its hardware to the operating system for your use, and Figure 1-2 provides a visual representation of how software virtualization takes form and how the physical machine hardware is partitioned out into multiple virtual segments for your use.

All the magic happens in the virtualization layer seen in Figure 1-2. This layer decouples your hardware and handles all allocation and utilization of the physical hardware by your virtualized machines.

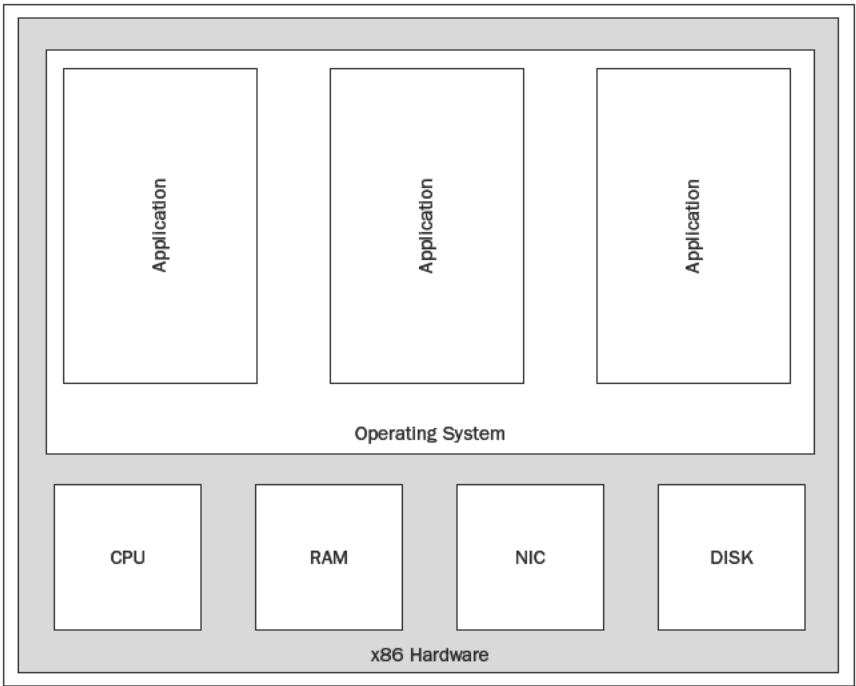


Figure 1-1

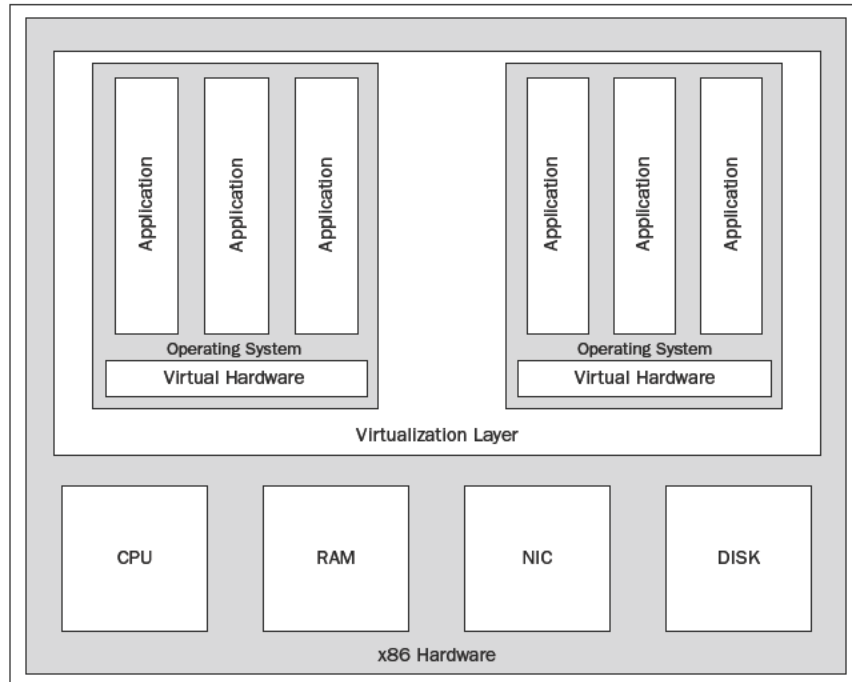


Figure 1-2

Hosted vs. Hypervisor Architecture

You can't really complete the overview of virtualization without a quick discussion of hosted versus hypervisor architecture.

- ❑ **Hosted architecture** is where your virtualization software is installed as an application onto the pre-existing host operating system. This means that your virtualization layer relies on your host operating system for device support and physical resource management. VMware Server, VMware Workstation, and Microsoft Virtual PC are good examples of a hosted architecture.
- ❑ **Hypervisor architecture** is more of a bare metal approach where there is no pre-existing operating system. The virtualization software is installed on a clean system, and it provides kernel and driver support for the raw physical hardware. VMware ESX server is a good example of virtualization utilizing hypervisor style architecture.

Chapter 1

Figure 1-3 and Figure 1-4 provide a visual representation of the architecture types and their relation to the physical hardware.

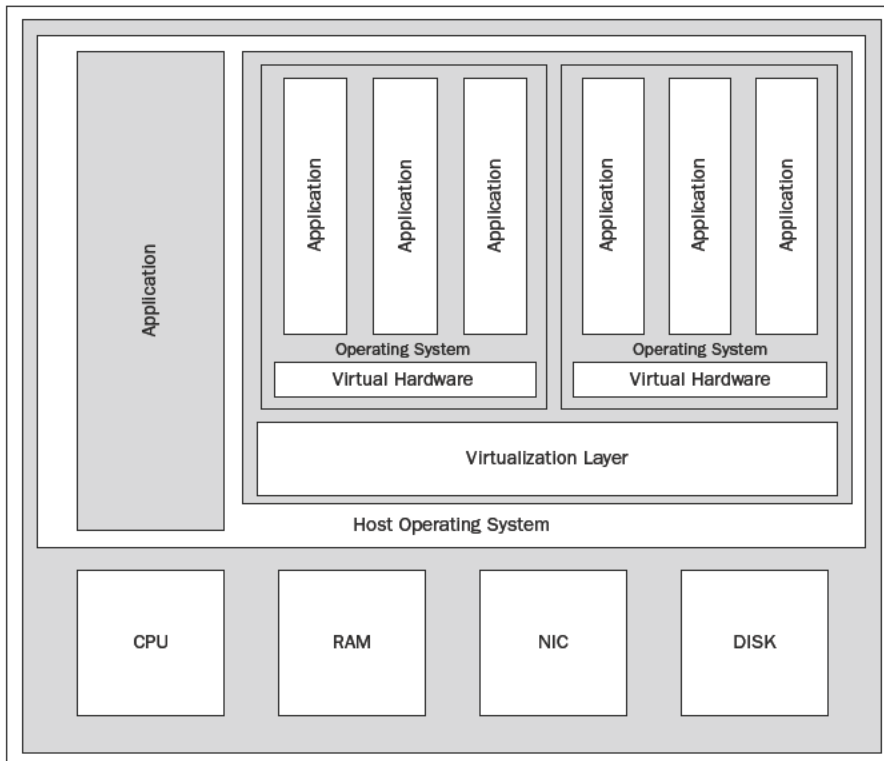


Figure 1-3

Hosted style architecture relies on the underlying operating system for hardware support. This means that theoretically as long as the underlying operating system supports the hardware it should be available to the virtualized machines. This allows for a greater variety of supported hardware. This comes at a price however, the added overhead of the physical machine's operating system. This is where the hypervisor architecture comes in. It doesn't require the physical machine to have a pre-existing operating system to be installed. In practice the virtualization software will install its own kernel and drivers for hardware. This allows for a dramatic decrease in overhead over the hosted architecture but it also seriously reduces the choices in compatible hardware. The reason is that companies, like VMware, have to choose a subset of hardware to write compatible drivers for, whereas drivers are readily available for hardware under Windows XP or Windows Server 2003 because the hardware manufacturers produce the drivers.

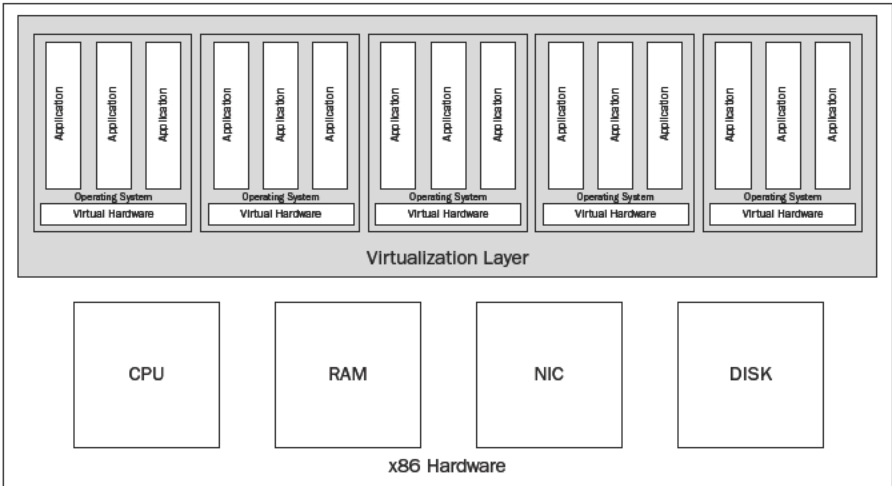


Figure 1-4

In conclusion, the following tables outline the advantages and disadvantages of each type of architecture.

Hosted Architecture	
Advantages	Disadvantages
Ease of deployment	Increased cost due to licensing of host software
Greater hardware support	Physical resources such as memory are shared between the host and virtual machines
Decrease cost for virtualization software	Less server utilization possible due to the additional overhead of the underlying host operating system

Hypervisor Architecture	
Advantages	Disadvantages
Greater resources for the virtual machines	Decreased hardware support
Decreased cost for licensing because no host operating system is required	Increased cost for virtualization software. Hypervisor software like VMware ESX server is considerably more expensive.
Increased utilization of server hardware	

Terminology, Host vs. Guest

If you remember the definition of hosted architecture you'll undoubtedly see where the term host comes from. The *host machine* is the physical machine itself, the machine where you will be installing VMware Server. The *host operating system* is the actual operating system installed on the host machine. Many times you will simply see them both referred to as just the host.

In the discussion of virtualization earlier in the chapter you learned that there are segments, or partitions, created in order to isolate and house your virtual machines. Once a new virtual machine is created, it is essentially a clean slate. Just like with a new PC you must install an operating system onto the newly created virtual machine. The *guest* refers to the operating system installed on the virtual machine. As with host, many times you'll see guest refer to both the virtual machine and its operating system.

I cover how to create virtual machines in Chapter 4.

I could spend the next several pages going over some of the other terminology used in virtualization and VMware Server; however, I will leave that for the chapters ahead. Nevertheless, I did want to stress the importance of understanding this difference between the host and guest. Though it is not a difficult concept, it is a very common one over the next several chapters.

So What Is VMware Server?

VMware Server is a multi-platform (Windows Server and Linux) application that provides host-based software virtualization. This gives you the ability to create and operate multiple x86-based virtual machines simultaneously on the same physical machine. Of course all of that is great, but what does it really do for you? Here are just a few of the advantages to running your server or development environment within virtual machines:

- ☐ Rapid deployment/decreased setup and tear down of new machines
- ☐ Multiple platforms available on a single physical machine
- ☐ Ability to move virtual machines from one server to another with no loss of information
- ☐ Increased utilization of your hardware/decreased costs

The next few sections go over each of these advantages in more detail.

Rapid Deployment

Especially in a development environment you'll find the ability to set up and tear down a set of virtual machines quickly is worth its weight in gold. This rapid setup and tear down can be accomplished by creating a library of base images. These images will contain a static install of, for example, Windows XP. Having a library of base images will allow you to quickly unpack and deploy virtual machines in a matter of minutes.

Chapter 4 covers creating and organizing a base image library.

As an example, say you have an obscure application that works right only on Windows 98. A customer calls with a serious problem to troubleshoot. In the past you would have to either dual boot a workstation into Windows 98 or have an old junk PC with Windows 98 on it for testing. Of course they aren't just lying around when you need them to be; you have to go find what door they are propping open, and what should be a 15-minute support call turns into a whole afternoon.

Now, say you receive the same phone call but this time you have a handy virtual machine already set up with Windows 98 installed. You simply connect to the VMware Server console and fire up your Windows 98 machine. You have a testing environment set up and ready within minutes, usually with the customer still on the phone. Once you finish the call you simply shut it down and forget it.

Multiple Platforms Available on a Single Physical Machine

If you're like me you spend most of your day on a Windows PC doing Windows development. The problem is that I like to tinker with new technology. I always want to try out that new Linux distro, but I never have a spare machine just lying around. This is where one of my favorite benefits of virtualization comes into the picture. If you remember earlier I talked about how the virtualization layer abstracts out the hardware from its implementation. This allows you to not just run Windows-based virtual machines, but you can run Linux, BSD, and so on. Almost anything that will run on an x86, you can run side by side with anything else in VMware Server, regardless of the host operating system. I just love that part. I would have to say the most common use I've found for this, from a developer's view, is setting up an Apache environment for web development. PHP and Perl just work so much better on an Apache installation in Linux.

The Ability to Move Virtual Machines from One Server to Another

As you will learn in Chapter 3, virtual machines in VMware Server are really nothing more than files on your hard disk. This coupled with the independent nature of the image with respect to the host hardware makes it extremely portable. In other words, there is nothing that keeps you from moving that Windows XP base image from your development VMware Server over to your VMware Workstation installation or over to a VMware ESX Server.

There are some minor incompatibilities between VMware Server, ESX, and Workstation images. The product you use to create the image will prompt you if a choice is made that will render the image incompatible with other versions.

As an example, suppose you have two highly critical servers running in virtual machines on one server. Now, and we've all had this happen, a fan goes out on that server. You can't buy the fan anywhere but direct, it always seems to be that way. You can rig the machine to come up but you don't have room in your data center for a machine cracked open with a box fan propped above it. All you have to do is get that server up long enough to copy those virtual machines over to another VMware Server and switch them on. Problem solved and your users are never the wiser. Suppose this server was simply a firewall or instant messaging server. Backup images of machines like that can be kept on another VMware Server machine for easy, quick fall over.

Chapter 1

The flexibility all the VMware products offer in this regard is outstanding. If care is taken to maintain compatibility when the virtual machines are created, you can move, mix, and match your virtual machines in any way you see fit. You can even use virtual machine images from other virtualization products like Microsoft Virtual PC and Microsoft Virtual Server. VMware Server can read and convert the images into a format compatible for use on your VMware products.

Increased Utilization of Your Hardware

Figuring the Return on Investment gives me a headache; however, it doesn't take too much to see that a \$3000 server that averages a CPU load of 25 percent has a lower return than a \$3000 server with an average load of 75 percent. The simple fact is your servers are probably underutilized. Take any server currently in your datacenter and monitor it for several hours. It may shock you to learn that an average server is rarely utilized over 10 to 20 percent. That leaves a bit to be desired in terms of your investment. VMware Server can keep that server doing what it was bought for, pushing bits. By consolidating your servers into a single host machine capable of multiple virtual machines, you increase the return on your investment. I haven't even figured in the decrease in power consumption, air conditioning, hardware failure, and other issues that plague a datacenter made up of several individual servers where each performs just a handful of tasks.

A perfect example that I've seen and heard from many people is that they have a piece of software that cannot be installed alongside a different piece of software; however, both are needed in the complete system. The product documentation probably tells you that you need two servers. Why? You just need virtualization. Create two virtual machines. Install the operating system and software needed on each and away you go. You save yourself the cost of an additional server and most likely still do not max out the hardware on your existing server.

Life without VMware Server

Looking at this in terms of your development process, what is life like without VMware Server? I may not be able to answer that question from your perspective, but it is certainly my goal to show you what your development process can be like with VMware Server as a part of it.

In terms of physical machines the common developer has three requirements, depending on the project of course:

- ☐ **Developer machine** — Usually the machine on your desk.
- ☐ **Servers** — Where you develop web-based code (Web Services, ASP.NET, and so on).
- ☐ **Test bed** — The bank of machines you use to test your code.

In a world where virtualization did not exist, the preceding requirements could result in a large number of physical machines. The test bed alone, if you were doing any serious stress or regression testing, could be comprised of several machines itself. Of course there can be many other types of machines added to the preceding list, but for the purposes here it covers the basic requirements.

Now, as you probably already know, I'm going to tell you the machines required for the preceding scenario would fit perfectly into virtual machines.

First, having your developers write code from within a development virtual machine has numerous advantages. In Chapter 4 I cover the advantages and disadvantages of placing your developers solely within a virtual machine. One of the biggest advantages, for the sake of discussion here, is a static environment. You can create a static development image complete with all the tools, IDE, and so on needed to function within your environment and then distribute that out to your development team. Now all of your developers are writing code on the same platform with the exact same settings and install. No more worrying about your developers installing the latest game or spyware on their work PC. Everything is static and isolated within your development image.

There are numerous advantages and disadvantages to developing within a virtual machine as well as a few instances where this approach might not produce the results you desire. Further information is provided in Chapter 4.

Development servers are a perfect candidate for virtualization. Typically your development servers are not production servers, and most companies don't have the budget or extra hardware lying around to house a server just for testing code. There's almost no better use for a virtual machine.

The final common requirement is the test bed. There were two ways to handle multiplatform testing in the past. Either you had a room full of physical computers with various operating systems installed on them or you had one machine that you just spent hours in front of. More often than not in the latter situation testing would take a back seat due to the huge time requirement of testing on one machine. This is prime time stuff for VMware Server. Not only can you accomplish multiplatform testing for your code with multiple virtual machines, but I show you later in this book how you can automate the process so you can push builds out, automatically start virtual machines, and even retrieve status back from your installers and applications while testing. Imagine your nightly build starting up the necessary virtual machines, copying over the bits, and e-mailing your QA department that it's ready to be tested. That's some pretty cool stuff.

Summary

Over the past few sections you've learned about the following:

- ❑ The basics of virtualization, the types of architecture utilized, and terminology
- ❑ What VMware Server is and the advantages of using virtual machines
- ❑ The difficulties of life without VMware Server, or virtualization in general

Chapter 2 goes over the installation steps required to get VMware server up and running on your Windows or Linux-based server. In the following chapters you will take that installation and go through some ways in which you can integrate virtual machines into your existing development processes and hopefully increase your productivity.