



Windows® 7

Portable Command Guide

All the MCTS 70-680, and MCITP
70-685 and 70-686 Commands in
One Compact, Portable Resource

PEARSON

DARRIL GIBSON

What Do You Want to Do?

I want to:

Topic	Chapter	Page
Add drivers to a WinPE image	22	247
Add drivers to offline images	21	240
Add features to offline images	21	240
Append an image to a .wim file	20	233
Capture a Windows 7 image	20	231
Capture user state data	24	265
Check and repair core operating system files	12	135
Check and repair disk integrity	14	172
Check connectivity with computers	13	152
Check local and remote system security	10	116
Configure a proxy server with netsh	11	126
Configure a system for remote management	15	184
Configure PowerShell to run scripts	28	309
Convert a disk to NTFS	5	61
Copy permissions when copying files	4	46
Copy text to and from the Command Prompt window	1	9
Create a batch file with Notepad	6	63
Create a bootable VHD file	5	56
Create a hard-link migration store	24	263
Create a list of domain computers	28	315
Create a message box	28	322
Create a system repair disc	14	171
Create and save a PowerShell script in the ISE	27	303
Deploy a Windows 7 image	20	235
Determine whether a file exists	28	318
Disable driver signing checks	14	170
Drag a path from Windows Explorer to the command prompt	3	32
Flush the DNS cache	13	145
Flush the DNS cache with a PowerShell script	28	314
Identify which drivers are signed	12	137
Log an event to an event log	6	73
Manipulate power settings	8	89
Manipulate which processes start on bootup	9	107
Map a drive to a UNC path	4	53

Continues on inside back cover.

Windows 7 Portable Command Guide: MCTS 70-680, and MCITP 70-685 and 70-686

Darril Gibson

Windows 7 Portable Command Guide: MCTS 70-680, and MCITP 70-685 and 70-686

Darril Gibson

Copyright © 2011 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-4735-8

ISBN-10: 0-7897-4735-9

Library of Congress Cataloging-in-Publication data is on file.

Printed in the United States of America

First Printing: March 2011

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Pearson IT Certification cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Bulk Sales

Pearson IT Certification offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact

International Sales
international@pearson.com

Publisher
Paul Boger

Associate Publisher
Dave Dusthimer

Acquisitions Editor
Betsy Brown

Development Editor
Andrew Cupp

Series Editor
Scott Empson

Managing Editor
Sandra Schroeder

Senior Project Editor
Tonya Simpson

Copy Editor
Bill McManus

Proofreader
The Wordsmithery LLC

Technical Editor
Chris Crayton

Publishing Coordinator
Vanessa Evans

Book Designer
Gary Adair

Composition
Studio Galou, LLC

Contents at a Glance

Introduction 1

PART I: Command Prompt Basics

- CHAPTER 1** Launching and Using the Command Prompt 3
- CHAPTER 2** Basic Rules When Using the Command Prompt 15
- CHAPTER 3** Using Basic Commands 27

PART II: Working with Files, Folders, and Disks

- CHAPTER 4** Manipulating Files and Folders 37
- CHAPTER 5** Manipulating Disks 55

PART III: Scripting with Batch Files

- CHAPTER 6** Creating Batch Files 63
- CHAPTER 7** Scheduling Batch Files 81

PART IV: Managing and Maintaining Windows 7

- CHAPTER 8** Using Windows 7 Maintenance Tools 89
- CHAPTER 9** Retrieving Information About Windows 7 103
- CHAPTER 10** Managing Security in Windows 7 111
- CHAPTER 11** Configuring Windows 7 with **netsh** 121

PART V: Troubleshooting Windows 7

- CHAPTER 12** Configuring Windows 7 Using Basic Troubleshooting Tools 135
- CHAPTER 13** Troubleshooting Network Issues 145
- CHAPTER 14** Using Recovery Tools 161

PART VI: Remote Administration

- CHAPTER 15** Managing Remote Windows 7 Systems 177
- CHAPTER 16** Windows Management Instrumentation Command Line 187

PART VII: Creating and Managing Images

- CHAPTER 17** Understanding Images 203
- CHAPTER 18** Preparing a System to Be Imaged with sysprep 215
- CHAPTER 19** Installing the Windows Automated Installation Kit 221
- CHAPTER 20** Using imagex 227
- CHAPTER 21** Using the Deployment Image Service and Management (DISM) Tool 237

PART VIII: Working with WinPE and Setup

- CHAPTER 22** WinPE Commands 247
- CHAPTER 23** Installing Windows 7 with **setup** 251

PART IX: Migrating User Data with USMT

- CHAPTER 24** Capturing User Data with **scanstate** 259
- CHAPTER 25** Restoring User Data with **loadstate** 273

PART X: Using PowerShell

- CHAPTER 26** Understanding PowerShell Commands 279
- CHAPTER 27** Using the Integrated Scripting Environment (ISE) 299
- CHAPTER 28** Creating and Running PowerShell Scripts 309

PART XI: Group Policy and the Command Line

- CHAPTER 29** Group Policy Overview 329
- CHAPTER 30** Group Policy Command-Line Tools 341
- APPENDIX** Create Your Own Journal Here 347

Table of Contents

INTRODUCTION 1

PART I: Command Prompt Basics

CHAPTER 1	Launching and Using the Command Prompt	3
	Launching the Command Prompt	3
	Pinning the Command Prompt to the Start Menu and the Taskbar	4
	Launching the Command Prompt with Elevated Privileges	4
	Responding to the UAC Prompt	5
	Using the Built-in Doskey Program	7
	Creating Mini Macros in Doskey	9
	Copying from and Pasting into the Command Prompt	9
	Copy from the Command Prompt Window	10
	Paste Text into the Command Prompt Window	10
	Changing the Options and Display of the Command Prompt	11
CHAPTER 2	Basic Rules When Using the Command Prompt	15
	Using Uppercase or Lowercase	15
	Using Quotes to Enclose Spaces	16
	Understanding Variables	17
	Understanding Switches	18
	Understanding Wildcards	18
	Getting Help	19
CHAPTER 3	Using Basic Commands	27
	Understanding Folders and Directories	27
	Understanding Paths	28
	Viewing Files and Folders with dir	30
	Changing the Path with cd	31
	Using Drag and Drop to the Command Prompt Window	32
	Deleting Folders with rd	33
	Creating Folders with md	33
	Controlling Output with more	34
	Redirecting Output	35
	Using the Redirect Symbol (>)	35
	Using clip	35
	Using exit	35

PART II: Working with Files, Folders, and Disks

- CHAPTER 4 Manipulating Files and Folders 37
- Associating File Extensions with **assoc** and **ftype** 37
 - assoc** 39
 - ftype** 39
 - Viewing Attributes with **attrib** 41
 - Compressing Files with **compact** 43
 - Encrypting Files with **cipher** 44
 - Copying Files with **copy**, **xcopy**, and **robocopy** 46
 - copy** 46
 - xcopy** 47
 - robocopy** 48
 - Deleting or Removing Files with **del** 50
 - Expanding Cabinet Files with **expand** 51
 - Taking Ownership of Files with **takeown** 51
 - Removing the Windows.old Folder 52
 - Mapping Drives with **net use** 53
- CHAPTER 5 Manipulating Disks 55
- Manipulating and Viewing Disks with **diskpart** 55
 - Creating a VHD File with **diskpart** 56
 - Modifying the BCD Store for the VHD File 57
 - Defragmenting Volumes with **defrag** 59
 - Converting to NTFS with **convert** 61
 - Managing the File System with **fsutil** 62

PART III: Scripting with Batch Files

- CHAPTER 6 Creating Batch Files 63
- Using Notepad 63
 - Giving Feedback with **echo** 64
 - Using Parameters 65
 - Calling Another Batch File with **call** 68
 - Clearing the Screen with **cls** 69
 - Changing the Order of Processing with **goto** 69
 - Checking Conditions with **if** 70
 - Logging Events with **eventcreate** 73
 - Looping Through Tasks with **for** 75

Creating a Menu File	76
Documenting the Script with rem	76
Pausing for the User	77
Giving Users a Choice	77
Creating the Batch File	78

CHAPTER 7 Scheduling Batch Files 81

Scheduling Batch Files with Task Scheduler	81
Scheduling Tasks from the Command Line	82
Creating a Scheduled Task with schtasks	83
Creating a Scheduled Task from an XML File with schtasks	83
Deleting a Scheduled Task with schtasks	85
Viewing Scheduled Tasks with schtasks	85
Changing Scheduled Tasks with schtasks	86
Running and Ending Scheduled Tasks with schtasks	87

PART IV: Managing and Maintaining Windows 7

CHAPTER 8 Using Windows 7 Maintenance Tools 89

Controlling Power Settings with powercfg	89
Query the Power Settings with powercfg	89
Modify the Power Settings with powercfg	91
Analyze the Power Settings with powercfg	93
Checking Memory with mdsched	93
Manipulating Services with sc	95
Collecting Events with wecutil	97
Restarting Systems with shutdown	97
Launching Control Panel Applets from the Command Line	99

CHAPTER 9 Retrieving Information About Windows 7 103

Viewing System Information with systeminfo	103
Viewing User Information with whoami	106
Launching System Configuration with msconfig	107

CHAPTER 10 Managing Security in Windows 7 111

Manipulating Credentials with cmdkey	111
Installing Updates with wusa	112
Installing Applications with msiexec	115
Auditing Systems with mbsacli	116

CHAPTER 11 Configuring Windows 7 with netsh 121

- Understanding **netsh** 121
- Understanding Contexts 122
- Configuring a Proxy Server 126
- Configuring the Network Interface with netsh 126

PART V: Troubleshooting Windows 7

CHAPTER 12 Configuring Windows 7 Using Basic Troubleshooting Tools 135

- Identifying the System Name with **hostname** 135
- Verifying Core System Files with **sfc** 135
- Checking Digital Signatures with **sigverif** 137
- Viewing Active Tasks with **tasklist** 138
- Terminating Processes with **taskkill** 142
- Viewing Installed Drivers with **driverquery** 144

CHAPTER 13 Troubleshooting Network Issues 145

- Viewing and Manipulating TCP/IP Configuration with **ipconfig** 145
- Viewing the Physical Address with **getmac** 151
- Checking Connectivity with **ping** 152
- Viewing the Router Path with **tracert** 155
- Checking for Data Loss with **pathping** 155
- Checking for Records in DNS with **nslookup** 156

CHAPTER 14 Using Recovery Tools 161

- Manipulating the Boot Process with **bcdedit** 161
 - Commands That Operate on a Store 165
 - Commands That Control Output 166
 - Commands That Control the Boot Manager 167
 - Commands That Operate on Entries in the Store 169
 - Commands That Operate on Entry Options 170
 - Disable Driver Signing 170
- Creating a System Repair Disc with **recdisc** 171
- Checking and Repairing Disks with **chkdsk** 172
- Repairing BitLocker Drives with **repair-bde** 175

PART VI: Remote Administration

- CHAPTER 15 Managing Remote Windows 7 Systems 177
- Configuring Windows 7 for Remote Administration 177
 - Configuring System for Remote Desktop Connection (RDC) 177
 - Configuring System for Remote Access Using an MMC 180
 - Connecting to Remote Systems with **mstsc** 181
 - Configuring Windows Remote Management with **wintrm** 184
 - Executing Commands Remotely with **wins** 185
- CHAPTER 16 Windows Management Instrumentation Command Line 187
- Understanding **wmic** 187
 - Configuring the Firewall to Allow **wmic** 188
 - Running **wmic** 189
 - Modifying the Format with the /format Switch 191
 - Retrieving Help from **wmic** 194
 - Understanding Aliases 194
 - Operating System Aliases 196
 - Disk Drive Aliases 197
 - System Hardware Aliases 198
 - User, Group, and Domain Aliases 200
 - Using Verbs 200

PART VII: Creating and Managing Images

- CHAPTER 17 Understanding Images 203
- Understanding WIM Files 203
 - Using Windows Deployment Services and Images 205
 - Comparing Boot and Install Images 207
 - Comparing Thick and Thin Images 213
- CHAPTER 18 Preparing a System to Be Imaged with Sysprep 215
- Understanding Sysprep 215
 - Running the Sysprep GUI 216
 - Running **sysprep** from the Command Line 217

CHAPTER 19	Installing the Windows Automated Installation Kit	221
	Downloading the WAIK	221
	Installing the WAIK	221
	Viewing the WAIK Tools	224
CHAPTER 20	Using imagex	227
	Creating WinPE on a Bootable USB	227
	Capturing Images with imagex	231
	Appending Images with imagex	233
	Deleting Images with imagex	235
	Deploying Images with imagex	235
CHAPTER 21	Using the Deployment Image Service and Management (DISM) Tool	237
	Online Servicing Versus Offline Servicing	237
	Mounting Images	238
	Getting Information About Mounted Images	240
	Modifying Images	240
	Working with Drivers	241
	Working with Packages and Features	242
	Miscellaneous Image Modifications	244
	Saving Images	245
PART VIII: Working with WinPE and Setup		
CHAPTER 22	WinPE Commands	247
	Add Drivers to WinPE with drvload	247
	Run Commands Within WinPE with wpeutil	248
CHAPTER 23	Installing Windows 7 with setup	251
	Installing Windows 7 with setup	251
	Using the Autounattend.xml File	252
	Using slmgr	253
	Basic slmgr Commands	254
	slmgr Activation Commands	255
	slmgr Commands Used for KMS Clients	256
	slmgr Commands Used for Token-Based Activation	257

PART IX: Migrating User Data with USMT

- CHAPTER 24 Capturing User Data with **scanstate** 259
- Understanding Migration Stores 259
 - Reviewing Migration Paths 259
 - Understanding Hard-Link Migration Stores 263
 - Capturing Migration Data with **scanstate** 265
 - Specifying XML Files 266
 - Modifying XML Files 266
 - Identifying Users to Migrate with **scanstate** 268
 - Encrypting the Migration Store with **scanstate** 269
 - Simple Error Handling with **scanstate** 270
 - Handling EFS Files 270
 - Migrating Data from the Windows.old Folder 271
- CHAPTER 25 Restoring User Data with **loadstate** 273
- Running **loadstate** 273
 - Specifying XML Files 274
 - Specifying Hard-Link Options 275
 - Deleting the Hard-Link Store 275
 - Identifying Users to Migrate 276
 - Decrypting an Encrypted Migration Store 277
 - Handling Errors 278

PART X: Using PowerShell

- CHAPTER 26 Understanding PowerShell Commands 279
- Launching PowerShell 279
 - Understanding PowerShell Verbs and Nouns 280
 - Tabbing Through PowerShell Commands 284
 - Understanding the Different Types of PowerShell Commands 285
 - Creating Aliases 286
 - Discovering Windows PowerShell Commands 286
 - Exploring **get-member** 288
 - Redirecting Output with Windows PowerShell 291
 - Understanding PowerShell Errors 291
 - Understanding PowerShell Variables 293
 - Using Comparison Operators 295
 - Understanding Pipelining 296

CHAPTER 27	Using the Integrated Scripting Environment (ISE)	299
	Launching the ISE	299
	Exploring the ISE	300
	Executing Commands in the ISE	302
	Creating and Saving a Script in the ISE	303
CHAPTER 28	Creating and Running PowerShell Scripts	309
	Setting the Security Context	309
	Creating a PowerShell Profile	310
	Creating and Modifying the Global PowerShell Profile	312
	Running PowerShell Scripts	313
	Flushing the DNS Cache	314
	Creating a List of Domain Computers	315
	Logging Processes with a get-process Script	316
	Testing for the Existence of a File	318
	Creating Output as HTML	318
	Running a Script Against Multiple Computers	320
	Creating a PowerShell Message Box	322
	Scheduling PowerShell Scripts	326

PART XI: Group Policy and the Command Line

CHAPTER 29	Group Policy Overview	329
	Understanding Group Policy Settings	329
	Blocking Inheritance	333
	Enforcing GPOs	334
	Using Loopback Processing	335
	Running Scripts with Group Policy	336
	Running PowerShell Scripts via Group Policy	339
CHAPTER 30	Group Policy Command-Line Tools	341
	Viewing Group Policy Settings with gpresult	341
	Refreshing Group Policy Settings with gpupdate	343
APPENDIX A	Create Your Own Journal Here	347

About the Author

Darril Gibson is the CEO of Security Consulting and Training, LLC. He regularly teaches, writes, and consults on a wide variety of security and technical topics. He's been a Microsoft Certified Trainer for more than ten years and holds several certifications, including MCSE (NT 4.0, 2000, 2003), MCDBA (SQL Server), MCITP (Windows 7, Server 2008, SQL Server), ITIL v3, Security+, and CISSP. He has authored, coauthored, or contributed to more than a dozen books. You can view a listing of most of his current books on Amazon: <http://amzn.to/bL0Obo>.

Dedication

To my wife, who continues to provide me with love and encouragement. I'm thankful we are sharing our lives together.

Acknowledgments

A book like this is never done in a vacuum. I'm grateful for all the hard work done behind the scenes by the people at Pearson. I'm thankful to Scott Empson, who had the original vision for these books, and grateful that David Dusthimer had faith in me to head up many of the books in the Microsoft series. I especially appreciated the efforts of two key editors, Andrew Cupp and Chris Crayton. This book is much better due to the efforts of these people.

A book like this is never done in a vacuum. I'm grateful for all the hard work done behind the scenes by the people at Pearson. I'm thankful to Scott Empson, who had the original vision for these books, and grateful that David Dusthimer had faith in me to head up many of the books in the Microsoft series. I especially appreciated the efforts of two key editors, Andrew Cupp and Chris Crayton. This book is much better due to the efforts of these people.

About the Series Editor

Scott Empson is the associate chair of the Bachelor of Applied Information Systems Technology degree program at the Northern Alberta Institute of Technology in Edmonton, Alberta, Canada, where he teaches Cisco routing, switching, and network design courses. Scott is also the program coordinator of the Cisco Networking Academy Program at NAIT, a Regional Academy covering Central and Northern Alberta. He has earned three undergraduate degrees: a Bachelor of Arts, with a major in English; a Bachelor of Education, again with a major in English/Language Arts; and a Bachelor of Applied Information Systems Technology, with a major in Network Management. Scott also has a Masters of Education degree from the University of Portland. He holds several industry certifications, including CCNP, CCAI, Network+, and C|EH.

Scott is the series creator and one of the authors of the Portable Command Guide Series. Portable Command Guides are filled with valuable, easy-to-access information to quickly refresh your memory. Each guide is portable enough for use whether you're in the server room or the equipment closet.

About the Technical Editor

Christopher A. Crayton is an author, technical editor, technical consultant, security consultant, trainer, and SkillsUSA state-level technology competition judge. Formerly, he worked as a computer and networking instructor at Keiser College (2001 Teacher of the Year); as network administrator for Protocol, a global electronic customer relationship management (eCRM) company; and at Eastman Kodak headquarters as a computer and network specialist. Chris has authored several print and online books, including *The A+ Exams Guide*, Second Edition (Cengage Learning, 2008), *Microsoft Windows Vista 70-620 Exam Guide Short Cut* (O'Reilly, 2007), *CompTIA A+ Essentials 220-601 Exam Guide Short Cut* (O'Reilly, 2007), *The A+ Exams Guide*, *The A+ Certification and PC Repair Handbook* (Charles River Media, 2005), *The Security+ Exam Guide* (Charles River Media, 2003), and *A+ Adaptive Exams* (Charles River Media, 2002). He is also co-author of *How to Cheat at Securing Your Network* (Syngress, 2007). As an experienced technical editor, Chris has provided many technical edits/reviews for several major publishing companies, including Pearson Education, McGraw-Hill, Cengage Learning, Wiley, O'Reilly, Syngress, and Apress. He holds MCSE, A+, and Network+ certifications.

We Want to Hear from You!

As the reader of this book, you are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As an associate publisher for Pearson IT Certification, I welcome your comments. You can email or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that I cannot help you with technical problems related to the topic of this book. We do have a User Services group, however, where I will forward specific technical questions related to the book.

When you write, please be sure to include this book's title and author as well as your name, email address, and phone number. I will carefully review your comments and share them with the author and editors who worked on the book.

Email: feedback@pearsonitcertification.com

Mail: David Dusthimer
Associate Publisher
Pearson IT Certification
800 East 96th Street
Indianapolis, IN 46240 USA

Reader Services

Visit our website and register this book at pearsonitcertification.com for convenient access to any updates, downloads, or errata that might be available for this book.

This page intentionally left blank

Introduction

Thanks for buying *Windows 7 Portable Command Guide*. I'd love to say that this book was my idea, but the real credit goes to Scott Empson, who originally developed the vision of this book with Cisco certifications. I've worked with Scott and Pearson Publishing to help bring the same type of books he created for Cisco products to professionals working on Microsoft products. Scott's vision started with the idea that many IT professionals who have already learned the theory still sometimes need help remembering how to implement it.

The book doesn't go into depth teaching these concepts. The idea is that you already understand them. Instead, the goal is to provide enough information to help you remember what you can do and how to do it in a small, portable, and useful journal, not an encyclopedic-sized volume. However, even if a concept is new to you, there's enough information for you to start typing at the command prompt to gain a better understanding.

As an example, you probably know that you can refresh Group Policy from the command prompt, but you might not always remember the exact command is **gpupdate /force**. You might remember that sysprep is used to prepare a computer for imaging, but you might not always remember that the full command is **sysprep /oobe /generalize**. In other words, you know the theory behind why you'd update Group Policy, and why you'd run sysprep, but you might not always remember the syntax. This book is a ready reference of useful commands and procedures with clear-cut examples. It shows the exact syntax of many of the commands needed for administrative tasks performed regularly by Windows 7 administrators.

I started the outline of this book by ensuring that command-prompt commands covered by the Microsoft Certified Information Technology Professional (MCITP) certifications on Windows 7 were included. This includes the 70-680 and 70-685 exams for the MCITP: Enterprise Desktop Support Technician 7 certification, and the 70-680 and 70-686 exams for the MCITP: Enterprise Desktop Administrator 7 certification. I then added the commands I've found valuable in my day-to-day work on networks and from classroom teaching.

Many IT professionals use an engineering journal to help them remember key information needed on the job. It might include specific commands that they sometimes forget, IP addressing schemes used on their networks, steps for important maintenance tasks that are performed infrequently, or anything else they want to easily recall by looking at

the journal. If you already have an engineering journal of your own, you can add this as a Windows 7 addendum. If you don't have one, you can start with this book. It includes the same "Create Your Own Journal Here" appendix that Scott uses in the Cisco series. These are blank pages you can use to add your own notes and make this your journal, not mine.

Command Syntax Conventions

The conventions used to present command syntax in this book are as follows:

- **Boldface** indicates syntax that is entered literally as shown.
- *Italic* indicates syntax for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive choices.
- Square brackets ([]) indicate an optional element.
- Braces ({ }) indicate a required choice.

Windows Management Instrumentation Command Line

This chapter provides information and commands concerning the following topics:

- Understanding **wmic**
- Configuring the firewall to allow **wmic**
- Running **wmic**
- Modifying the format with the **/format** switch
- Retrieving help from **wmic**
- Understanding aliases
- Using verbs

Understanding wmic

wmic is the command-line implementation of Windows Management Instrumentation (WMI). It extends WMI so that you can execute many WMI commands without a full understanding of the underlying details.

WMI is a group of technologies that allows different applications to interact with the Windows operating system. It is based on the Web-Based Enterprise Management (WBEM) standard and it's a full-blown scripting tool. Administrators use WMI scripting to perform a wide variety of administrative tasks, and WMI scripting is included in many third-party vendor tools.

NOTE: Creating scripts with WMI is beyond the scope of this book. However, the scripting pros at Microsoft have an active website with a lot of rich content on WMI. Check it out here: <http://technet.microsoft.com/dd742341.aspx>.

Some of the most valuable commands and switches are summarized in the following table.

Command	Description
<code>wmic alias list brief</code> C:\> <code>wmic alias list brief</code>	Retrieves a list of all aliases.
<code>wmic aliasname list full</code> C:\> <code>wmic computersystem list full</code>	Retrieves a list of all properties and known values for any alias.
<code>wmic aliasname get /?</code> C:\> <code>wmic computersystem get /?</code>	Retrieves a list of properties for an alias, including the data type and available operations.
<code>wmic aliasname set /?</code> C:\> <code>wmic computersystem set /?</code>	Retrieves a list of properties for an alias that can be modified. The list includes the data type and available operations.
<code>/output:target</code> C:\> <code>wmic /output:c:\data\cpu.txt computersystem list full</code>	Redirects the output to a file. You can redirect the output to the Clipboard by using <code>/output:clipboard</code> .
<code>delete (a process)</code> <code>wmic process where (name = process-name) delete</code> C:\> <code>wmic process where (name = 'notepad.exe') delete</code>	Deletes an instance of a running process.

Configuring the Firewall to Allow wmic

If you want to run **wmic** commands on remote computers, you may need to enable the firewall on the remote connections. The primary error you'll see that indicates that wmic commands are prevented by the firewall is "The RPC server is unavailable."

TIP: You'll also see the error "The RPC server is unavailable" if the remote system is unreachable. You can try the **ping** command to determine if the remote system is operational and verify you're using the correct hostname.

You can configure the firewall to allow **wmic** commands by allowing the WMI program through the firewall in the proper profile. Figure 16-1 shows the window for doing so, which you can reach by starting the Control Panel, entering **Firewall** in the Search Control Panel text box, and selecting **Allowing a Program Through Windows Firewall**.

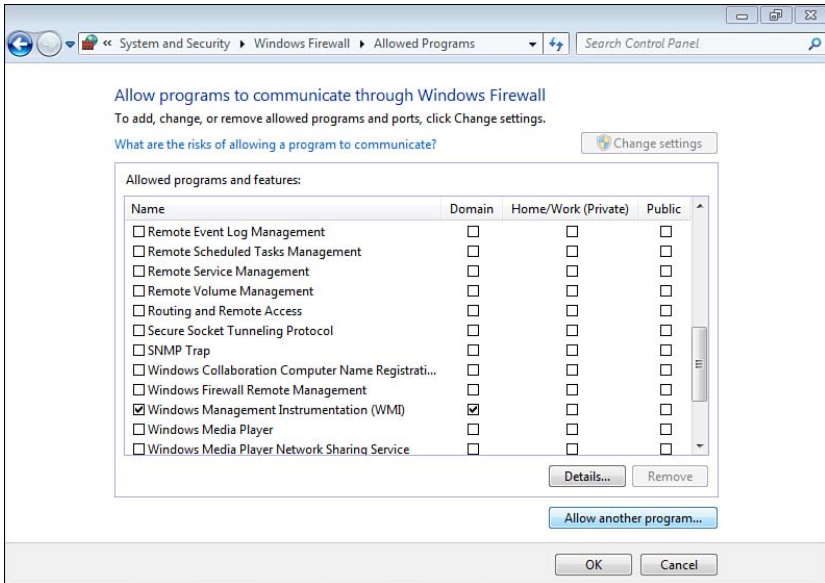


Figure 16-1 Enabling Windows Management Instrumentation in the Firewall

Running wmic

wmic is a shell command similar to **netsh**, covered in Chapter 11, “Configuring Windows 7 with netsh.” You can enter **wmic** from the command prompt to enter the wmic shell. The wmic shell prompt starts in the root/cli name space, from which you can then enter commands. For example, if you want to get detailed information on the computer, you can use the **computersystem list full** command:

```
C:\>wmic
wmic:root\cli>computersystem list full
```

You can also enter the full **wmic** command from the command prompt by preceding it with **wmic**. For example, the following command provides the same output as the previous command:

```
C:\>wmic computersystem list full
```

If you were writing this within WMI (not **wmic**), you would have to understand the query language, and the query would look something like this:

```
Select * from Win32_ComputerSystem
```

However, thanks to the **wmic** built-in aliases, you don't have to learn the query language to use **wmic**.

The **wmic** command includes several switches. Some of the more common switches are listed in the following table.

TIP: Some commands don't recognize the switch unless it is entered before the command (right after **wmic**) rather than after the command (at the end of the **wmic** command string).

Switch	Description
<pre>/? [:brief :full] C:\>wmic /? C:\>wmic /?:full</pre>	<p>Shows the syntax of all global switches and aliases.</p> <p>The default listing is brief, but you can also specify full to get a more verbose listing of help.</p>
<pre>/node:remotecomputer wmic /node:remotecomputer command C:\>wmic /node:win7pcg computersystem C:\>wmic /node:win7pcg printer list brief</pre>	<p>You can use the /node switch to retrieve information from any remote computers.</p> <p>The first example retrieves information with the <code>computersystem</code> alias, and the second example uses the <code>printer</code> alias.</p>
<pre>/user:username wmic /node:remotecomputer /user:username command C:\>wmic /node:win7pcg /user:pearson \administrator computersystem</pre>	<p>Provides the username to be used during the session or for the command. You will be prompted to enter a password.</p> <p>This is useful when connecting to remote systems, but it can't be used to change the credentials on the local system.</p>
<pre>/password:password wmic /node:remotecomputer /user:username /password:password command C:\>wmic /node:win7pcg /user:pearson \administrator /password:P@ssw0rd computersystem</pre>	<p>Provides the password to be used with the specified user. This command must be used with a username.</p>

<pre> /output:target [stdout clipboard file path and name] wmic /node:remotecomputer /output:target command C:\>wmic /node:win7pcg /output:clipboard computersystem C:\>wmic /node:win7pcg /output:filename computersystem C:\>wmic /node:win7pcg /output:c:\ scripts\test.txt computersystem </pre>	<p>Identifies where to redirect the output. The output is normally sent to the screen but can be sent to the Clipboard or to a file. The /output switch needs to go before the alias.</p> <p>When sending it to a file, the path must exist or the command will fail with an “Invalid file name” error.</p> <p>TIP: The normal redirect symbol (>) can also be used, as in the following example:</p> <pre> C:\>wmic /node:win7pcg com- putersystem > c:\scripts\ test.txt </pre>
<pre> /append:target [stdout clipboard file path and name] C:\>wmic /node:remotecomputer /append:filename command C:\>wmic /node:win7pcg /append:c: \scripts\test.txt computersystem </pre>	<p>Identifies where to redirect the output.</p> <p>If the file doesn’t exist, it will be created. If it does exist, the output is appended to the file.</p> <p>When sending it to a file, the path must exist or the command will fail with an “Invalid file name” error.</p> <p>NOTE: The /append switch sends the data to the file and to the screen.</p>

Modifying the Format with the /format Switch

The **/format** switch has a few more options that you might find useful. It can be very useful when you combine it with the **/output** switch to send the data to a file in a specific format.

TIP: Each of these commands uses the **computersystem** alias. However, the format of the command is the same with any alias.

Format Switch	Description
<pre>table /format:table C:\>wmic computersystem list full / format:table</pre>	<p>Formats the output as a table with headers.</p>
<pre>list /format:list C:\>wmic computersystem list full / format:list</pre>	<p>Formats the output as a list of each property followed by the value.</p>
<pre>csv /format:csv C:\>wmic computersystem list full / format:csv C:\>wmic /output:c:\data\test.csv computersystem list full /format:csv</pre>	<p>Formats the output as comma-separated values. The header is displayed first separated by commas. The data is then displayed separated by commas.</p> <p>CSV files are easily read in Microsoft Excel, so it's common to use the output switch to send this data to a file.</p> <p>NOTE: When you use the output switch, it needs go before the alias.</p>
<pre>xml /format:xml C:\>wmic computersystem list full / format:xml C:\>wmic /output:c:\data\test.xml computersystem list full /format:xml</pre>	<p>Formats the output in Extensible Markup Language (XML) format.</p> <p>If you enter the path to the XML file (such as C:\data\test.xml), the file opens in your web browser.</p>
<pre>hform C:\>wmic computersystem list full / format:hform C:\>wmic /output:c:\data\test.html computersystem list full /format:hform</pre>	<p>Formats the output as an HTML document. This is useful to display all the properties of an object on a separate row.</p> <p>If you enter the path to the HTML file (such as C:\data\test.html), the file opens in your web browser.</p> <p>Figure 16-2 shows the output of this command.</p>
<pre>htable C:\>wmic computersystem list brief /format:htable C:\>wmic /output:c:\data\test3.html useraccount list brief /format:htable</pre>	<p>Formats the output as an HTML document. In the table format, each object is a single row.</p> <p>Figure 16-3 shows the output of this command.</p>

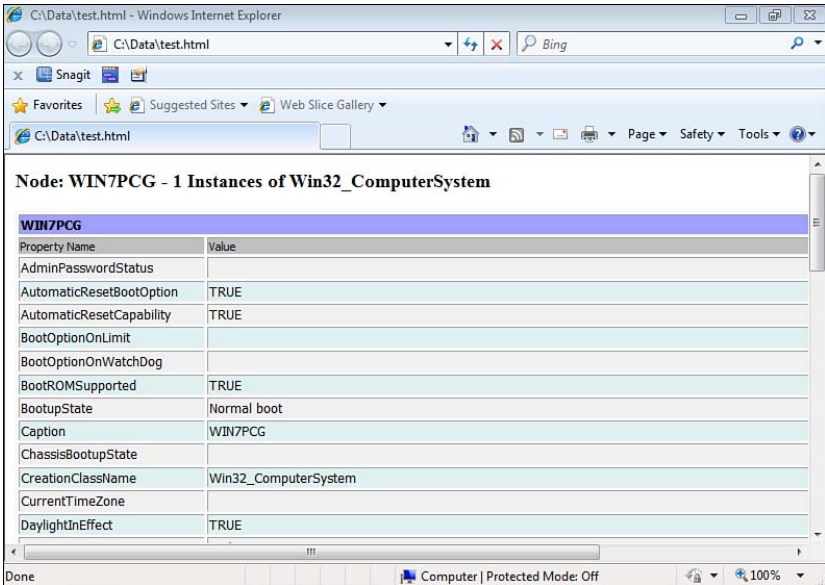


Figure 16-2 Viewing the Output of `wmic` in Internet Explorer in hform Format

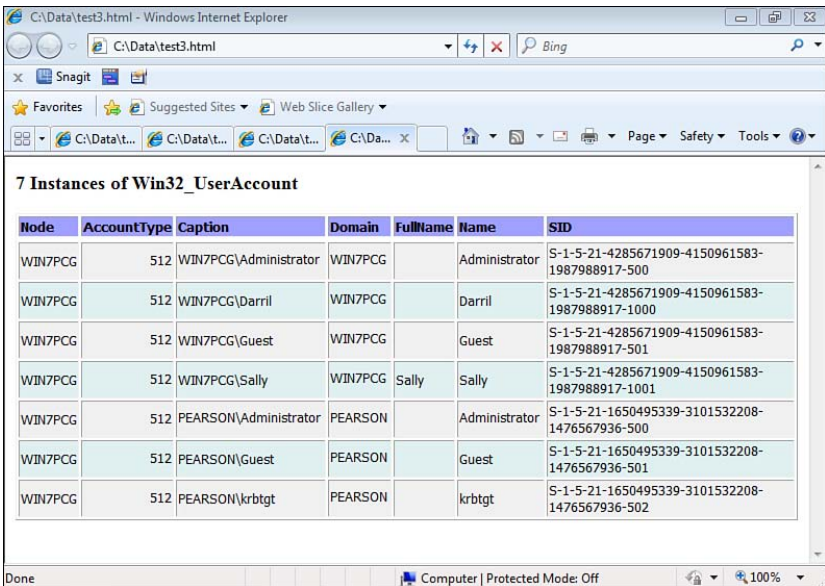


Figure 16-3 Viewing the Output of `wmic` in Internet Explorer in htable Format

Retrieving Help from wmic

You can retrieve help from **wmic** using multiple methods, as shown in the following table.

TIP: Each of these help commands supports the **/?:full** clause. This sometimes provides more verbose output, but other times it doesn't provide any extra information.

Help Command	Description
<pre>/? C:\>wmic /? C:\>wmic /?:full</pre>	Shows the syntax of all global switches and aliases.
<pre>switch /? /switch_name /? C:\>wmic /output /?</pre>	Shows information about any single global switch. The example will show help on the output switch.
<pre>alias /? wmic alias /? C:\>wmic computersystem /?</pre>	Shows information about aliases in general when the word alias is used. If you give the name of an actual alias, it provides information on the alias.
<pre>alias verb /? wmic alias verb /? C:\>wmic computersystem get /? C:\>wmic computersystem get /?:full C:\>wmic computersystem set /? C:\>wmic computersystem set /?:full</pre>	Shows information about one alias and verb combination. This can let you know what properties can be retrieved with the get verb and what properties can be configured with the set verb.

Understanding Aliases

Aliases are simply friendly names for the detailed query. There are dozens of aliases that you can enter instead of a full **wmic** command. You don't have to understand how the underlying WMI language works to use the alias. For example, the **computersystem** alias can be used to retrieve information on a computer:

```
C:\>wmic computersystem list brief
```

```
Domain           Manufacturer      Model              Name
PrimaryOwnerName TotalPhysicalMemory
```

```
Pearson.pub      Microsoft Corporation Virtual Machine    WIN7PCG
Darril           1610145792
```

```
C:\>wmic /node:dc1 computersystem list brief /format:list
```

```

Domain=Pearson.pub
Manufacturer=Microsoft Corporation
Model=Virtual Machine
Name=DC1
PrimaryOwnerName=Windows User
TotalPhysicalMemory=1610063872

```

The **/format:list** switch sends the output as a list instead of a table, which sometimes can be harder to read. The **list brief** clause is used to show some basic details. You can retrieve a much fuller output by using the **list full** clause:

TIP: The **list full** clause sends the output in the list format by default, so this clause is not needed here.

```
C:\>wmic computersystem list full
```

```

AdminPasswordStatus=3
AutomaticResetBootOption=TRUE
. . .
Description=AT/AT COMPATIBLE
Domain=Pearson.pub
DomainRole=1
. . .
EnableDaylightSavingsTime=TRUE
. . .
Manufacturer=Microsoft Corporation
Model=Virtual Machine
Name=WIN7PCG
. . .
ThermalState=1
TotalPhysicalMemory=1610145792
UserName=PEARSON\Administrator
WakeUpType=6
Workgroup=

```

NOTE: The entire output for **computersystem list full** spans multiple pages and thus is not listed in its entirety here.

The following tables show many of the aliases that are available. The first column shows the alias friendly name with a short description and its usage. The second column shows the Pwhere usage. If the alias will list multiple items, such as multiple services, you can retrieve data on a single item. WMI uses the Pwhere clause, but with **wmic** you only need to include the name between two single apostrophes. The third column shows the underlying WMI query that is executed.

NOTE: Some items have only a single instance, so a Pwhere clause is not defined within the alias.

Operating System Aliases

The following table shows some aliases that can retrieve data on the operating system.

Alias Friendly Name and Usage	Pwhere Format	WMI Query
computersystem Details on installed operating system and settings C:\>wmic computersystem list brief	Not defined	Select * from Win32_ComputerSystem
os Operating system details C:\>wmic os list brief	Not defined	Select * from Win32_OperatingSystem
environment Listing of environment variables C:\>wmic environment list brief	Not defined	Select * from Win32_Environment
sysdriver Installed services and drivers and current state C:\>wmic sysdriver list brief	Where Name='#' C:\>wmic sysdriver 'disk' list brief	Select * from Win32_SystemDriver
service System services C:\>wmic service list brief	Where Name='#' C:\>wmic service 'winrm' list full	Select * from Win32_Service
process Running processes C:\>wmic process list brief	Where ProcessId='#' C:\>wmic process '6668' list brief	Select * from Win32_Process
startup Identify startup programs C:\>wmic startup list brief	Where Caption='#' C:\>wmic startup 'sidebar' list brief	Select * from Win32_StartupCommand
registry Information on registry C:\>wmic registry list full	Not defined	Select * from Win32_Registry

qfe Quick fix engineering (hot-fixes) C:\>wmic qfe list brief	Not defined	Select * from Win32_QuickFixEngineering
nventlog Event logs C:\>wmic nventlog list brief	Where LogfileName='#' C:\>wmic nventlog 'application' list brief	Select * from Win32_NTEventlogFile
timezone Time zone data C:\>wmic timezone list full	Not defined	Select * from Win32_TimeZone
bootconfig Boot configuration data C:\>wmic bootconfig list full	Not defined	Select * from Win32_BootConfiguration
recoveros Location of recovery OS C:\>wmic recoveros list brief	Not defined	Select * from Win32_OSRecoveryConfiguration
wmiset WMI settings, including whether it's enabled or not C:\>wmic wmiset list brief	Not defined	Select * from Win32_WMISetting

Disk Drive Aliases

These aliases can be used to retrieve information related to disks.

Alias Friendly Name and Usage	Pwhere Format	WMI Query
diskdrive Details on dis3k drive C:\>wmic diskdrive list full	Where Index='#' C:\>wmic diskdrive '1' list brief	Select * from Win32_DiskDrive
logicaldisk Drive data C:\>wmic logicaldisk list full	Where Name='#' C:\>wmic logicaldisk 'c:' list brief	Select * from Win32_LogicalDisk

partition Information on disk partitions or volumes <code>C:\>wmic partition list full</code> <code>C:\>wmic partition list brief</code>	Where Index='#' <code>C:\>wmic partition '0' list full</code>	<code>Select * from Win32_DiskPartition</code>
diskquota Disk quota settings <code>C:\>wmic diskquota list full</code>	Not defined	<code>Select * from Win32_DiskQuota</code>
quotasetting Disk quota settings <code>C:\>wmic quotasetting list brief</code>	Not defined	<code>Select * from Win32_QuotaSetting</code>
pagefile Details on paging file(s) <code>C:\>wmic pagefile list brief</code>	Not defined	<code>Select * from Win32_PageFileUsage</code>
share Network shares <code>C:\>wmic share list brief</code>	Where Name='#' <code>C:\>wmic share 'c\$' list full</code>	<code>Select * from Win32_Share</code>
idecontroller IDE disk controllers <code>C:\>wmic idecontroller list brief</code>	Not defined	<code>Select * from Win32_IDEController</code>
cdrom CD- and DVD-ROM drives <code>C:\>wmic cdrom list brief</code>	Where Drive='#' <code>C:\>wmic cdrom 'd:' list full</code>	<code>Select * from Win32_CDROMDrive</code>

System Hardware Aliases

These aliases can be used to retrieve information on different hardware within the system.

Alias Friendly Name and Usage	Pwhere Format	WMI Query
csproduct Computer system model <code>C:\>wmic csproduct list full</code>	Not defined	<code>Select * from Win32_ComputerSystemProduct</code>

cpu Processor information C:\>wmic cpu list full	Where DeviceID='#' C:\>wmic cpu 'cpu0' list brief	Select * from WIN32_ PROCESSOR
systemslot Information on expansion slots C:\>wmic systemslot list brief	Not defined	Select * from Win32_ SystemSlot
memorychip Memory sticks C:\>wmic memorychip list full	Where Tag = '#' C:\>wmic memorychip ' physical memory 0 ' list brief	Select * from Win32_ PhysicalMemory
memphysical Memory totals C:\>wmic memphysical list full	Not defined	Select * from Win32_ PhysicalMemoryArray
bios Details on BIOS C:\>wmic bios list full	Not defined	Select * from Win32_BIOS
desktopmonitor Display monitor C:\>wmic desktopmoni- tor list full	Where DeviceID='#' C:\>wmic desktop- monitor 'desktop- monitor1' list full	Select * from WIN32_ DESKTOPMONITOR
nicconfig Configuration of network interface cards (NICs) C:\>wmic nicconfig list brief	Where Index='#' C:\>wmic nicconfig '1' list brief	Select * from Win32_ NetworkAdapter Configuration
nic NICs C:\>wmic nic list brief	Where DeviceID='#' C:\>wmic nic '1' list brief	Select * from Win32_ NetworkAdapter
printer Installed printers C:\>wmic printer list brief	Where Name='#' C:\>wmic printer 'Microsoft xps document writer' list full	Select * from Win32_ Printer

User, Group, and Domain Aliases

You can use these aliases to get information on objects such as users and groups.

Alias Friendly name and Usage	Pwhere Format	WMI Query
useraccount User account details C:\>wmic useraccount list brief	Not defined	Select * from Win32_UserAccount
group User groups C:\>wmic group list brief	Not defined	Select * from Win32_Group
sysaccount Detailed information on all user and groups, including all the built-in accounts C:\>wmic sysaccount list brief	Where Name='#' C:\>wmic sysaccount 'everyone' list brief	Select * from Win32_SystemAccount
ntdomain Information on domain (if joined) C:\>wmic ntdomain list brief	Where DomainName='#' C:\>wmic service 'pearson' list full	Select * from Win32_NTDomain

TIP: These lists of aliases are not complete. If you want to retrieve a full list of all the available aliases, use the command **wmic alias list brief**.

Using Verbs

There are several verbs that can be used with aliases. In simplest terms, the verbs are commands that you can use to work with the aliases.

Verbs (Commands)	Description
where where (<i>property</i> = "value") wmic alias where (<i>property</i> = "value") list full C:\>wmic useraccount where (name = "guest") list full	Use to filter the output. The value must be enclosed in double quotes. Only valid properties of the alias can be used in a where clause. You can view all valid properties of any alias with the following command: wmic alias list full

<pre> get get <i>property</i> wmic /node:remotecomputer <i>alias</i> get <i>property</i> C:\>wmic /node:win7pcg computersystem get username C:\>wmic /node:win7pcg computersystem get username, domain, totalphysicalmemory C:\>wmic useraccount where (name = "sally") get C:\>wmic useraccount where (name = "sally") get disabled </pre>	<p>You can use the get command to retrieve one or more properties of any alias. If you want to retrieve multiple properties, you separate each with a comma.</p> <p>TIP: You can identify all the properties you can retrieve from an alias by using the command wmic alias list full. You can identify all the properties that can be retrieved with the wmic alias get /? command.</p> <p>The first example to the left gets the username of a logged-in user on a remote system of a remote computer. The second example gets the username, the domain, and the amount of physical memory installed on the remote computer.</p> <p>You can also use a where clause to filter the data. In the last two examples, a where clause is used to retrieve properties on a user account named Sally, and then only the value of the disabled property.</p>
<pre> set set <i>property</i> = "value" wmic /node:remotecomputer <i>alias</i> set <i>property</i> = "value" wmic /node:remotecomputer <i>alias</i> set <i>property</i> C:\>wmic /node:win7pcg useraccount where (name = "guest") set disabled = "true" </pre>	<p>The set command allows you to set some alias properties.</p> <p>The example combines the set command with the where clause to disable the guest account on a remote system. The value must be specified in double quotes.</p> <p>TIP: You can't set all properties. For example, the memphysical alias reports what physical memory is installed, but you can't change these properties with the set command. You can identify all the properties that can be configured with the wmic alias set /? command.</p>
<pre> delete wmic <i>alias</i> where (<i>property</i> = value) delete C:\>wmic process where (name = 'notepad.exe') delete </pre>	<p>Deletes an instance.</p> <p>You can use this to terminate processes. The example terminates a running instance of Notepad.</p>

```
assoc
wmic alias assoc
C:\>wmic os assoc
wmic alias where (property =
'value') assoc
C:\>wmic group where (name =
'administrators') assoc
```

assoc shows the associations with an object. In the first example, it displays information about the operating system alias.

The second example shows all WMI objects that are associated with the Administrators group by adding a **where** clause.