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

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

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

800 East 96th Street
Indianapolis, Indiana 46240 USA

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

Second Printing: March 2014

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 **bcdedi**t 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 **winrm** 184

Executing Commands Remotely with **winrs** 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.

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.

This page intentionally left blank

CHAPTER 12

Configuring Windows 7 Using Basic Troubleshooting Tools

This chapter provides information and commands concerning the following topics:

- Identifying the system name with **hostname**
- Verifying core system files with **sfc**
- Checking digital signatures with **sigverif**
- Viewing active tasks with **tasklist**
- Terminating processes with **taskkill**
- Viewing installed drivers with **driverquery**

Identifying the System Name with hostname

There are times when you simply want to know the name of the system you're working on. One of the easiest ways to determine this from the command prompt is with the **hostname** command as follows:

```
c:\>hostname
```

The **hostname** command doesn't have any switches.

Verifying Core System Files with sfc

The sfc tool can be used to check the integrity of all protected system files and registry keys. If any of the files have been modified or corrupted, **it** can replace them with the correct version of the file. The basic syntax of the command is

```
sfc /scannow
```

TIP: Knowledge base article 929833 provides some additional information on the System File Checker, including some other uses. You can check it out here: http://support.microsoft.com/kb/929833.

Malicious software often attempts to modify system files. If it succeeds, the malware may have extensive control over the system. The **sfc** utility can be very useful in repairing system files that have been corrupted by malware. While some files can be repaired using the internal protected source files, you may be prompted to provide the installation DVD to restore system files.

NOTE: You can view the results of a scan in the cbs.log file, which is located in the %windir%\Logs\CBS folder.

The **sfc** command must be run from an elevated command prompt.

There are several switches that can be used with the **sfc** command, as shown in the following table.

Command	Description

`/scannow`
`C:\>`**`sfc /scannow`**
	The **/scannow** switch initiates a full check of all the system resources. It checks the integrity of the resources and repairs problems when possible. This check does take some time.

`/verifyonly`
`C:\>`**`sfc /verifyonly`**
	This is similar to the **/scannow** switch but doesn't attempt repairs. It only scans the files and reports any issues. If it finds problems, it reports the following:
	Windows Resource Protection found integrity violations. Details are included in the CBS.Log windir\Logs\CBS\CBS.log. For example C:\Windows\Logs\CBS\CBS.log.

`/scanfile`
`sfc /scanfile=`*`path and filename`*
`C:\>`**`sfc /scanfile=c:\windows\system32 \kernel32.dll`**
	You can scan specific files and check for integrity problems on these specific files. Just as the **/scannow** switch repairs any detected problems, the **/scanfile** switch repairs the file if problems are identified. You must specify the full path to the file.
	NOTE: You cannot include spaces before or after the **=** sign in this command. In other words, this command will not work: **sfc /scanfile =** *path*.

`/verifyfile`
`sfc /verifyfile=`*`path and filename`*
`C:\>`**`sfc /verifyfile=c:\windows\system32 \kernel32.dll`**
	The **/verifyfile** switch checks specific files for integrity problems and reports the results. It does not attempt a repair. You must specify the full path to the file.
	NOTE: You cannot include spaces before or after the **=** sign in this command. In other words, this command will not work: **sfc /verifyfile =** *path*.

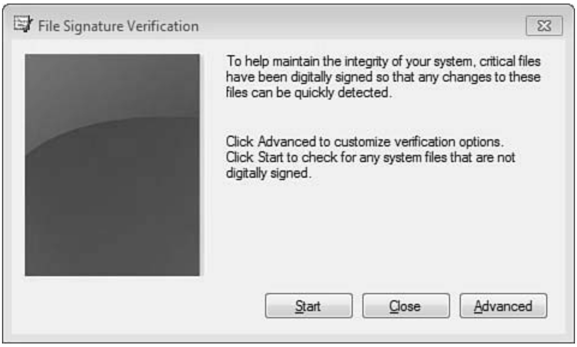
/offbootdir **/offwindir** **sfc /scannow /offbootdir=***driveletter* :\ **/offwindir=***drive letter and folder* C:\>**sfc /scannow /offbootdir=d:\** **/offwindir=d:\windows**	You can also scan an offline instance of Windows. In other words, if you have a dual-boot system, you can boot into one system and scan the other system. The **/offbootdir** switch specifies the offline drive and the **/offwindir** switch specifies the offline Windows folder. These switches are used in conjunction with other switches, such as the **/scannow** switch shown in the example. **NOTE:** You cannot include spaces before or after the = sign in this command.

Checking Digital Signatures with sigverif

The **sigverif** command launches the File Signature Verification tool, which can detect which files have been digitally signed and which files are not digitally signed. The syntax is

```
sigverif
```

There aren't any switches used with **sigverif**. Instead, it launches the graphical user interface shown in Figure 12-1. You can then run the tool by clicking **Start**.



File Signature Verification

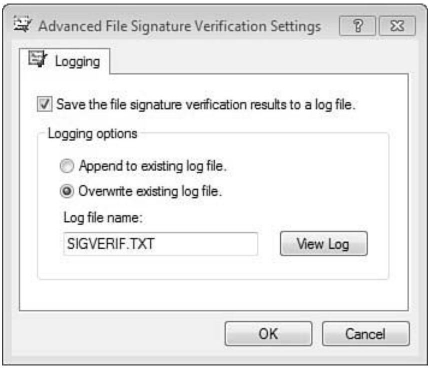
To help maintain the integrity of your system, critical files have been digitally signed so that any changes to these files can be quickly detected.

Click Advanced to customize verification options.
Click Start to check for any system files that are not digitally signed.

Figure 12-1 The File Signature Verification Tool

If all the files are digitally signed, a dialog box appears indicating "Your files have been scanned and verified as digitally signed." If any of the files are not digitally signed, you'll see a dialog box indicating which files are not digitally signed. The most common files you're likely to see that are not digitally signed are those related to display drivers because they are often newer. The display shows the name of the files, the path, the file type, and the version.

Figure 12-2 shows the logging options for **sigverif**. You can view this screen by clicking **Advanced**. It enables you to view the results of **sigverif** scans by clicking the View Log button. The sigverif.txt log file is located in the %systemdrive%\Users\Public\Documents folder.



Advanced File Signature Verification Settings

☑ Save the file signature verification results to a log file.

Logging options

○ Append to existing log file.

● Overwrite existing log file.

Log file name: SIGVERIF.TXT

Figure 12-2 File Signature Verification Logging Options

> **TIP: sigverif** can't be run exclusively from the command line. However, Microsoft has a free download available called **sigcheck** that was written by Mark Russinovich at Sysinternals. You can get it from http://technet.microsoft.com/bb897441.aspx. After you download and extract it, you can execute it from the command line. For example, you could use the following command to check for unsigned files in the system32 folder and log the results into a file named sigcheck.txt: **sigcheck -u c:\windows\system32 > sigcheck.txt**.

You can also identify signed and unsigned drivers using the **driverquery /si** command, as shown later in this chapter. This command lists all drivers with an IsSigned column indicating True or False.

Viewing Active Tasks with tasklist

You've probably used Task Manager to view a list of active tasks running on a system. You can also use the **tasklist** command to view a list of running tasks. The basic syntax is

```
tasklist
```

The output includes the tasks identified with their image name, process ID (PID), session name, session number, and memory usage. The following text shows partial output. Note that the system refers to the running tasks as images (under the column Image Name).

```
Image Name         PID   Session Name       Session#    Mem Usage
==============     ====  ============       =========   ============
System Idle Proc   0     Services           0           24 K
System             4     Services           0           3,592 K
smss.exe           288   Services           0           724 K
csrss.exe          440   Services           0           3,104 K
. . .
wininit.exe        500   Services           0           1,760 K
csrss.exe          536   RDP-Tcp#0          1           11,720 K
services.exe       560   Services           0           6,552 K
lsass.exe          576   Services           0           10,160 K
```

You have several different options that you can use with the **tasklist** command, as shown in the following table.

Command	Description
/m [*module name*] C:/>**tasklist /m** C:/>**tasklist /m ntdll.dll**	You can use the **/m** switch to determine what DLL modules are being used by the different tasks. When you omit the module name, all of the tasks are listed, and if any of the tasks are using modules, the modules are also included. You can also specify the module to determine which tasks are using the specific DLL module.
/svc C:/>**tasklist /svc**	You can list all of the associated services (if any) that are running to support each task in the task list.
/fi **tasklist /svc /fi "imagename eq** *taskname***"** C:\\>**tasklist /svc /fi "imagename eq svchost.exe"**	You can use the **/fi** switch to filter the output. The filter options are explored in greater depth later in this section, but this example might be useful. It's common to have multiple instances of the **svchost** process running, and you might want to know what services are related to the process. You can combine the **/svc** switch with **/fi** switch to show specifically what services are running for each instance of the **svchost** task.
/v C:\\>**tasklist /v**	The **/v** switch (verbose) will give you more detailed task information. It cannot be combined with the **/m** or **/svc** options.

/fo
tasklist /fo table | list | csv
C:\>**tasklist /fo list**
C:\>**tasklist /fo csv > c:\data\tasks.csv**
	You can use the **/fo** switch to change the format of the output. The default format is **table**. The comma-separated value (**csv**) format can be combined with the redirect symbol (**>**) to create a file that can easily be viewed using Microsoft Excel.

/nh C:\>**tasklist /fo csv /nh**	The **/nh** switch suppresses headers in the output. It is only valid when used with the **table** or **csv** outputs because the **list** output does not use headers.
/s **/u** **/p** **tasklist /s** *system* **/u** *user* **/p** *password* C:/>**tasklist /s dc1 /u pearson\administrator /p P@ssw0rd**	You can use the **/s** switch to run the command on a different computer. You'll also need to specify the username and password of an account that has permissions on the remote system. In the example, the command is run on the computer named dc1 in the pearson domain with the administrator account.

The **tasklist /svc /fi "imagename eq svchost.exe"** command shown in the preceding table can be used to only show services with an image name of svchost.exe. However, you have several additional options you can use to filter the output. These filters use the following operators:

- **eq**: Equal
- **ne**: Not equal
- **gt**: Greater than
- **lt**: Less than
- **ge**: Greater than or equal to
- **le**: Less than or equal to

TIP: When using filters, you need to enclose the entire filter comparison in quotes. In other words, this will work:

```
c:\>tasklist /fi "status eq Running"
```

However, this command will result in an error:

```
c:\>tasklist /fi status eq Running
```

The following table shows some examples using different operators.

Filter Name	Valid Operators	Valid Values
status **tasklist /fi "status eq running | not responding | unknown "** C:\>**tasklist /fi "status eq not responding"**	**eq**, **ne**	You can use a status of running, not responding, or unknown.
imagename **tasklist /svc /fi "imagename eq** *task name***"** C:\>**tasklist /svc /fi "imagename eq svchost.exe"**	**eq**, **ne**	The **imagename** is the valid name of a task or image in the list.
pid **tasklist /fi "pid gt** *###***"** C:\>**tasklist /fi "pid gt 100"**	**eq**, **ne**, **gt**, **lt**, **ge**, **le**	The process ID (PID) can be any number between 0 and 999,999,999. The actual PID can be identified from the **tasklist** output.

session
tasklist /fi "session ge *#***"**
C:\>**tasklist /fi "session ge 1"**
C:\>**tasklist /fi "session eq 0"**
C:\>**tasklist /fi "session eq 1"**
	eq, **ne**, **gt**, **lt**, **ge**, **le**	The number for the session can be used.
		Session 0 is the services session.
		Session 1 is the console session.
		Other sessions are from remote desktop sessions.
		The example will show all non-services sessions.

sessionname
tasklist /fi "sessionname ne *session name***"**
C:\>**tasklist /fi "sessionname ne services"**
C:\>**tasklist /fi "sessionname ne console"**
	eq, **ne**	You can enter the session name. Two common session names are services and console. Remote desktop sessions are numbered as RDP-Tcp#x, with the first remote session identified as RDP-Tcp#0. The example will show all non-services sessions.

cputime **tasklist /fi "cputime gt** *HH:MM:SS***"** C:\>**tasklist /fi "cputime gt 01:00:00"**	**eq**, **ne**, **gt**, **lt**, **ge**, **le**	The CPU time is in the format of HH:MM:SS. Hours (HH) can be any positive value. Minutes (MM) and seconds (SS) can be any value between 0 and 59.

memusage **tasklist /fi "memusage gt** *xxxx***"** C:\>**tasklist /fi "memusage gt 10240"**	**eq**, **ne**, **gt**, **lt**, **ge**, **le**	The memory usage enables you to view processes using specific amounts of memory expressed in KB. The example will show any processes using more than 10 MB of memory.

username
tasklist /fi "username eq *username***"**
C:\>**tasklist /fi "username eq darril"**
C:\>**tasklist /fi "username eq n/a"**
	eq, **ne**	You can view processes being run by specific users with the **username** filter. The **username** value doesn't normally show up unless you use the **/v** switch. In the first example, it will show all the services run by the user darril, and the second example shows the services running without a username.

services **tasklist /fi "services eq** *servicename***"** C:\>**tasklist /fi "services eq wsearch"**	**eq**, **ne**	You can search for tasks that are running specific services with the **services** filter. The example will show any services that are running the wsearch service (the searchindexer.exe task).
modules **tasklist /fi "modules eq** *module name***"** C:\>**tasklist /fi "modules eq ntdll.dll"**	**eq**, **ne**	The **modules** filter enables you to search for any tasks using specific modules. The example will list all the services using the ntdll.dll module.

Terminating Processes with taskkill

The **taskkill** command can be used to terminate a running process using its process ID (PID) or image name. You can use **taskkill** to terminate processes on the local system or remote system. The basic syntax is

```
taskkill /pid processID
```

or

```
taskkill /im imagename
```

NOTE: If there are multiple instances of the process, the command terminates all instances matching the **imagename** parameter in the **/im** switch. However, if the /**pid** switch is used it terminates only the process matching the **processID**. For example, if you have two instances of Notepad running, both will be terminated with **tasklist /im notepad.exe**.

The following table shows how to use some of these switches. An easy process to test these commands is Notepad. You can launch Notepad by entering **notepad** at the command prompt. You can then use the following command to get the details about it:

```
tasklist /fi "imagename eq notepad.exe"
```

You'll see a result similar to the following output. The PID is 1084 in the listing, but you will probably have a different PID.

```
Image Name          PID   Session Name       Session#    Mem Usage
================= ===== =============      =========== ============
notepad.exe         1084  Console            1           4,756 K
```

Command	Description
/im **taskkill /im** *task name* C:\>**taskkill /im notepad.exe**	The **/im** switch terminates the process identified by the image name.
/pid **taskkill /pid** *xxxx* C:\>**taskkill / pid 1084**	The **/pid** switch terminates the process identified by the PID number.
/s **/u** **/p** **taskkill /s** *system* **/u** *user* **/p** *password* **/im** image name C:\>**taskkill /s dc1 /u pearson\administrator /p P@ssw0rd /im notepad.exe**	You can use the **/s** switch to run the command on a different computer. You also need to specify the username and password of an account that has permissions on the remote system. In the example, the command is run on the computer named dc1 in the pearson domain with the administrator account.
/f C:\>**taskkill /im notepad.exe /f**	If an application has unsaved data, it will often prompt the user to save the data. You can use the **/f** switch to forcefully terminate the process and override the prompt. This parameter is ignored for remote processes; all remote processes are forcefully terminated.
/t C:\>**taskkill /im notepad.exe /t**	If an application has spawned child processes, you can use the **/t** switch to terminate the process and any child processes that it started.

TIP: You can use the same filters (with the **/fi** switch) in the **taskkill** command as you can use in the **tasklist** command.

Viewing Installed Drivers with driverquery

The **driverquery** command can be used to view a list of installed device drivers and properties about the drivers. The basic syntax is

```
driverquery
```

You can use some additional switches to modify the output, as shown in the following table.

Command	Description
/v C:\>**driverquery /v > c:\data\drivers.txt**	The **/v** switch is used to provide verbose output. This can be very extensive and isn’t very easy to read from the command prompt. If you redirect it to a text file, you can read it using Notepad.
/si C:\>**driverquery /si**	You can view information on both signed and unsigned drivers with the **/si** switch. You can’t combine the **/v** and **/si** switches.

/fo
driverquery /fo table | list | csv
C:\>**driverquery /fo list**
C:\>**driverquery /fo csv > c:\data\drivers.csv**
	You can use the **/fo** switch to change the format of the output. The default format is **table**. The comma-separated value (**csv**) format can be combined with the redirect symbol (**>**) to create a file that can easily be viewed using Microsoft Excel.

/nh C:\>**driverquery /fo csv /nh**	The **/nh** switch suppresses headers in the output. It is only valid when used with the **table** or **csv** outputs because the **list** output does not use headers.
/s **/u** **/p** **driverquery /s** *system* **/u** *user* **/p** *password* C:\>**driverquery /s dc1 /u pearson\administrator /p P@ssw0rd**	You can use the **/s** switch to run the command on a different computer. You also need to specify the username and password of an account that has permissions on the remote system. In the example, the command is run on the computer named dc1 in the pearson domain with the administrator account.