

# Pro Windows PowerShell



Hristo Deshev

## **Pro Windows PowerShell**

**Copyright © 2008 by Hristo Deshev**

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

ISBN-13 (pbk): 978-1-59059-940-2

ISBN-10 (pbk): 1-59059-940-3

ISBN-13 (electronic): 978-1-4302-0546-3

ISBN-10 (electronic): 1-4302-0546-6

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

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

Lead Editor: Tony Campbell

Technical Reviewer: Jon Rolfe

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,  
Jonathan Gennick, Kevin Goff, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann,  
Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Sofia Marchant

Copy Editor: Heather Lang

Associate Production Director: Kari Brooks-Copony

Production Editor: Laura Cheu

Compositor: Susan Glinert Stevens

Proofreader: Linda Seifert

Indexer: Julie Grady

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto:info@apress.com), or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

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

The source code for this book is available to readers at <http://www.apress.com>. You will need to answer questions pertaining to this book in order to successfully download the code.

*To my wife, Yana, for all her love and support*

# Contents at a Glance

About the Author .....	xvii
About the Technical Reviewer .....	xix
Acknowledgments .....	xxi
Introduction .....	xxiii

<b>CHAPTER 1</b>	Objects and Object Types .....	1
<b>CHAPTER 2</b>	Controlling Execution Flow .....	33
<b>CHAPTER 3</b>	The Object Pipeline .....	55
<b>CHAPTER 4</b>	Working with Script Blocks .....	69
<b>CHAPTER 5</b>	Working with Functions .....	87
<b>CHAPTER 6</b>	Command Aliases .....	107
<b>CHAPTER 7</b>	Providers .....	127
<b>CHAPTER 8</b>	Script Files .....	139
<b>CHAPTER 9</b>	Error Handling and Debugging .....	155
<b>CHAPTER 10</b>	Signing Scripts .....	197
<b>CHAPTER 11</b>	The Shell Environment and Its Configuration .....	219
<b>CHAPTER 12</b>	Extending the Type System .....	237
<b>CHAPTER 13</b>	Getting Help .....	253
<b>CHAPTER 14</b>	Taming Processes and Services .....	275
<b>CHAPTER 15</b>	Input and Output .....	285
<b>CHAPTER 16</b>	Monitoring Your System .....	297
<b>CHAPTER 17</b>	PowerShell and the World Wide Web .....	315
<b>CHAPTER 18</b>	Sending E-mail .....	349
<b>CHAPTER 19</b>	Talking to COM Objects .....	367
<b>CHAPTER 20</b>	Managing Windows with WMI .....	395
<b>CHAPTER 21</b>	PowerShell Community Extensions .....	417

■ <b>CHAPTER 22</b>	<b>PSEventing: .NET Events in PowerShell</b>	449
■ <b>CHAPTER 23</b>	<b>Enhancing Tab Completion with PowerTab</b>	461
■ <b>INDEX</b>		473



# Contents

About the Author .....	xvii
About the Technical Reviewer .....	xix
Acknowledgments .....	xxi
Introduction .....	xxiii
■ <b>CHAPTER 1    Objects and Object Types</b> .....	<b>1</b>
The Extended Type System .....	1
Accessing Object Properties .....	2
Using Object Methods .....	2
Object Adapters .....	3
Type Extensions .....	5
Built-in Types .....	8
Strings and String Operations .....	8
Numeric Types .....	14
Arrays and Collections .....	16
Dictionaries and Hash Tables .....	21
Type Literals .....	25
Type Conversion .....	26
Accessing Static Members .....	30
Summary .....	31
■ <b>CHAPTER 2    Controlling Execution Flow</b> .....	<b>33</b>
Conditional Expressions .....	33
Comparing Values .....	34
Implicit Type Conversions .....	36
Logic and Bitwise Operations .....	37
Boolean Conversions .....	41
The Power of String Conversions .....	42
Collections and Conditional Expressions .....	43

Branching Instructions .....	44
Simple Branching: if/else .....	44
Multiple Branches and the switch Statement .....	45
Loops: Repeatedly Executing Blocks of Code .....	48
The Simplest Loop: while .....	48
Loops and Counters: for Loop .....	50
Executing Actions on All Objects in a Collection: foreach Loop .....	50
Controlling Loop Execution: break and continue Statements .....	51
Summary .....	54
 <b>CHAPTER 3   The Object Pipeline .....</b>	 <b>55</b>
Text-Based Pipelines .....	55
Object-Based Pipelines .....	56
Processing Collections Using ForEach-Object .....	59
Filtering Collections Using Where-Object .....	60
Adding or Deleting Properties Using Select-Object .....	60
Sorting Collections .....	62
Pipeline Tees .....	63
Grouping Objects .....	64
Gathering Collection Statistics .....	65
Detecting Changes and Differences Among Objects .....	65
The Object Pipeline and Functional Programming .....	66
Summary .....	67
 <b>CHAPTER 4   Working with Script Blocks .....</b>	 <b>69</b>
Defining Script Blocks .....	69
Passing Parameters and Returning Values .....	71
Processing Pipeline Input .....	75
Variable Scope .....	77
Invoking Strings as Expressions .....	83
Script Blocks as Delegates .....	85
Summary .....	86
 <b>CHAPTER 5   Working with Functions .....</b>	 <b>87</b>
Defining Functions .....	87
Function Internals .....	88
Function Parameters .....	89
Passing Parameters by Reference .....	91



Returning Values.....	93
Scoping Rules.....	94
Filters.....	97
Functions and Script Blocks.....	101
Implementing New Control Structures.....	101
Script Blocks as Strategies.....	104
Summary.....	106

## CHAPTER 6 **Command Aliases**..... 107

Working with Aliases.....	107
Creating Aliases.....	108
Modifying Aliases.....	110
Exporting and Importing Aliases.....	111
Aliasing Tips, Techniques, and Pitfalls.....	115
Name Clashes.....	115
Complex Aliases.....	117
Removing Broken Aliases.....	118
Built-in Aliases.....	119
cmd.exe Look-alikes.....	120
UNIX Look-alikes.....	121
Convenience Aliases.....	123
Summary.....	125

## CHAPTER 7 **Providers**..... 127

Enumerating Providers.....	127
Drives.....	129
Drive Scope.....	131
Navigating to Drives.....	132
Provider Capabilities.....	134
Basic Provider Capabilities.....	134
Drive Providers.....	134
Item Providers.....	134
Item Container Providers.....	135
Navigation Providers.....	136
Item Content Providers.....	136
Item Property Providers.....	136
Dynamic Item Property Providers.....	137
Item Security Descriptor Providers.....	137
Summary.....	137

<b>CHAPTER 8</b>	<b>Script Files</b>	139
	Creating Scripts	139
	Invoking Scripts	140
	Passing Parameters	142
	Returning Values	146
	Executing PowerShell Scripts from Other Environments	148
	Developing and Maintaining Script Libraries	149
	Dot-sourcing as a Means to Include Script Libraries	150
	The Library Path Problem	150
	Summary	154
<b>CHAPTER 9</b>	<b>Error Handling and Debugging</b>	155
	Handling Errors	155
	Common Error-Handling Parameters	157
	Trapping Errors	160
	Capturing Nonterminating Errors	170
	Raising Errors	172
	Debugging Your Code	177
	Print Debugging	178
	Stepping Through Scripts and Breaking Execution	188
	Tracing Script Execution Details	190
	Summary	196
<b>CHAPTER 10</b>	<b>Signing Scripts</b>	197
	How Code Signing Works	197
	Guaranteeing Script Integrity	198
	Certifying the Origin of a Script	198
	Managing Certificates	199
	Creating a Self-Signed Certificate	202
	Creating a Certification Authority Certificate	202
	Issuing a Code-Signing Certificate	206
	Signing Scripts	211
	Running Scripts on Other Machines	215
	Summary	217

<b>CHAPTER 11</b>	<b>The Shell Environment and Its Configuration</b>	<b>219</b>
	Shell Hosts	219
	User Profile Scripts	223
	Settings for All Users and All Shells	223
	Settings for All Users Using a Specific Shell	224
	Settings for a Specific User and All Shells	224
	Settings for a Specific User and a Specific Shell	224
	Working with Saved Console Settings	226
	Changing the Prompt Settings	228
	Tab Expansion: How Command Completion Works	231
	Summary	235
<b>CHAPTER 12</b>	<b>Extending the Type System</b>	<b>237</b>
	Modifying Objects and Types	237
	Adding Members to Single Objects	238
	Adding Members to All Instances of a Class	244
	Extending Object Formatting	247
	Creating Our Own Views	249
	Summary	252
<b>CHAPTER 13</b>	<b>Getting Help</b>	<b>253</b>
	The Help System	253
	The Basics	254
	Parameter Details	256
	Advanced Techniques	258
	Getting Command Information	264
	Getting Information About Objects	266
	Using the Internet to Get Help	268
	Summary	273
<b>CHAPTER 14</b>	<b>Taming Processes and Services</b>	<b>275</b>
	Working with Processes	275
	Listing Processes: Getting the Big Picture	276
	Starting and Stopping Processes	277
	Processes and Their Windows	278
	Process Modules: On What Libraries Does This Baby Depend?	279

Getting Information About a Program's Publisher .....	280
Setting Process Priority .....	280
Administering Services .....	281
Starting and Stopping Services .....	282
Configuring Services .....	283
Analyzing Service Dependencies .....	283
Summary .....	284
 <b>■ CHAPTER 15 Input and Output .....</b>	<b>285</b>
Reading Content .....	285
Writing Content .....	288
Content Encoding .....	289
Getting Byte Content .....	290
Different Unicode Encodings .....	291
Extracting Data from Text .....	294
Finding Matches for a Regular Expression .....	294
Finding String Occurrences Inside Files .....	295
Summary .....	296
 <b>■ CHAPTER 16 Monitoring Your System .....</b>	<b>297</b>
Working with the Windows Event Log .....	297
Reading from the System Event Logs .....	298
Writing to the Event Log .....	302
Performance Counters .....	306
Consuming Counter Data .....	307
Monitoring Programs .....	309
Summary .....	313
 <b>■ CHAPTER 17 PowerShell and the World Wide Web .....</b>	<b>315</b>
Laying the Foundation .....	315
Fetching Files from the Web .....	316
Setting Connection Options and Debugging	
Connection Problems .....	318
Testing and Validating Web Sites .....	328
Test-Url: Verifying a Page's Existence .....	328
Checking If a Page Contains Broken Links .....	330

Working with News Feeds .....	332
Getting Feeds .....	333
Calling Web Services .....	338
Calling Web Services Using HTTP GET Requests .....	338
Calling Web Services Using HTTP POST Requests .....	340
Calling Web Services Using the SOAP Protocol .....	342
Handling FTP Transactions .....	345
Downloading Files from an FTP Server .....	345
Uploading Files to an FTP Server .....	346
Summary .....	347
 <b>CHAPTER 18 Sending E-mail .....</b>	<b>349</b>
Using System.Net.Mail .....	349
Building a Reusable Script .....	350
Configuring Recipients .....	352
Working with Message Headers .....	355
Multipart Messages .....	357
Message Views .....	357
Attaching Files to Messages .....	359
Embedding Media in Messages .....	361
Putting It All Together .....	363
Summary .....	365
 <b>CHAPTER 19 Talking to COM Objects .....</b>	<b>367</b>
How COM Works .....	368
How PowerShell Supports COM .....	368
Scripting Programs Through Their COM Interfaces .....	369
Automating Microsoft Word .....	369
Scripting Microsoft Excel .....	378
Driving Internet Explorer .....	384
Windows Script Host Code Interoperability .....	389
Evaluating Code .....	390
Exposing Objects from MSScriptControl .....	392
Summary .....	393

<b>CHAPTER 20</b>	<b>Managing Windows with WMI</b>	<b>395</b>
	A Brief History of WMI	395
	WBEM and WMI Components	396
	The Common Information Model	396
	Network Transport	397
	Object Discovery	397
	PowerShell's WMI Support	398
	Get-WmiObject: The WMI Query Tool	399
	Language Support for WMI Objects	401
	Exploring CIM Classes and Objects	404
	Getting a List of Classes and Namespaces	404
	WMI System Properties	406
	Querying Hardware Devices	407
	Getting Information About Software	408
	Operating System Updates	408
	Installed Programs	409
	The Network Configuration	411
	Generating WMI Code	411
	Scriptomatic	412
	Summary	414
<b>CHAPTER 21</b>	<b>PowerShell Community Extensions</b>	<b>417</b>
	Installation and Configuration	417
	Tab Expansion	418
	Editing Configuration Files	418
	Getting Help	419
	File System Utilities	421
	Navigation Helpers	421
	NTFS Helpers	422
	Compressing and Archiving Files	424
	Network Utilities	425
	Executing Processes and Commands	427
	XML Tools	430
	Working with Image Files	435
	Clipboard Helpers	437

New Providers .....	440
Reading from the .NET Global Assembly Cache .....	440
Exploiting the Internet Explorer Feed Store .....	441
Manipulating Windows Active Directory .....	443
Utility Applications .....	446
Summary .....	448
<b>■ CHAPTER 22 PSEventing: .NET Events in PowerShell .....</b>	<b>449</b>
Introducing PSEventing .....	449
Event-Handling Mechanics .....	450
FileSystemWatcher: A Real-World Example .....	451
Monitoring Entries Written to the System Event Logs .....	452
Handling WMI Events .....	454
Detecting If Our Script Has Been Terminated by the User .....	456
Using Script Blocks As Event Handlers .....	457
Summary .....	459
<b>■ CHAPTER 23 Enhancing Tab Completion with PowerTab .....</b>	<b>461</b>
Installation .....	462
How PowerTab Works .....	463
Data Grid Pop-up Windows .....	466
Configuring PowerTab .....	467
The IntelliSense Completion Handler .....	468
The Tab Expansion Database .....	470
Summary .....	471
<b>■ INDEX .....</b>	<b>473</b>





# About the Author

**HRISTO DESHEV** has been a software developer, a team leader, and a product manager for Telerik, the leading vendor of user interface components (<http://www.telerik.com>). For the past several years, his focus has been on creating rich and, at the same time, robust and reliable web and desktop components running on the Microsoft .NET platform. A major goal for him and his team is to predictably deliver working software—applying agile software engineering practices to create rock-solid solutions targeting all modern web browsers and supporting multiple Visual Studio .NET versions.

Hristo is a reformed UNIX fanatic who now runs Windows and tries to apply his scripting skills and experiences to the Microsoft platform. He is also a tool freak: anything that can help automate a task—be it a scriptable utility or a full-blown scripting language—is a welcome addition to his toolbox. PowerShell has been a passion of his since its early unofficial releases due to its ease of use and almost incredible ability to automate all Windows-related tasks without needing other tools.

You can reach Hristo through his blog at <http://weblogs.asp.net/hristodeshev/>.



# About the Technical Reviewer

**JON ROLFE** has been an IT professional for more than 12 years and is currently a senior solutions architect for one of the world's largest IT services companies. He specializes in designing secure Microsoft-based enterprise architectures for clients in the United Kingdom's public sector and large blue-chip companies. During the course of his career, he has gained extensive experience at all stages of the IT life cycle, including enterprise architecture design, software development, rapid system deployment, and large enterprise systems support. Although he specializes in Microsoft technologies, Jon also has an interest in other technologies including Linux and wrote the book *Using Ubuntu Linux* (Bernard Babani, 2007).

When not working, Jon has an active interest in digital photography, follows motor sports, and enjoys mountain biking and skiing. He can be contacted through his web site at <http://JonRolfe.com>.



# Acknowledgments

I still find it hard to believe that I wrote an entire book! It has been an enjoyable project that was also quite formidable and downright hard. I would never have done it all on my own, without the huge amounts of help I got from many people.

First, I would like to thank the entire Apress team that guided me through the process: Tony Campbell for providing important input and valuable feedback on the content, Sofia Marchant for being so flexible with my sometimes late chapters, Heather Lang for her extraordinary ability to turn my often unreadable writings into well-thought-out text. Jon Rolfe deserves a special mention here; he has been an awesome technical reviewer. He not only spotted a host of problems with the sample scripts but was keen to provide solutions and improvement ideas; thus his efforts became one of the most important factors in getting this book done.

I would also like to thank the Microsoft PowerShell team for creating such a wonderful product. Being a long-time UNIX user, I have been disappointed by the primitive Windows command line shell. With PowerShell, this is no longer the case.

The PowerShell community deserves special recognition too. The wealth of information and thoughtful discussions on blogs and online forums has been an incredible inspiration when writing this book.

Last, but not least, I am personally thankful to the people behind the open source tools covered in the book: Keith Hill and the numerous contributors to the PowerShell Community Extensions project; Marc van Orsouw, the creator of the PowerTab tab completion extension; and Oisín Grehan, the man behind the PowerShell Eventing Library and many other PowerShell-related open source projects. You guys rock!



# Introduction

I remember the first time I got to play with Windows PowerShell—I had some spare time on my hands, and after reading a blog post, I quickly set off to download the unofficial pre-release bits that were available at the time. I originally had very low expectations about the tool—all I wanted was a shell that is up to par with its UNIX cousins. Boy, was I wrong! The moment I learned about object pipelines, I knew this tool was going to change the way I think about automating tasks on Windows. Fast forward a couple of months, and here I am, so much in love with PowerShell that I even wrote a whopping 400-page book about it.

## Why Another Book on Windows PowerShell?

As much as I like PowerShell, its plain-console looks make it very easy for people to dismiss it without really seeing what lies beneath the surface. I feel it is still largely undiscovered—I know many professionals who have yet to try it out. When administrators see PowerShell for the first time, many simply shrug, assuming it's just another version of the primitive command prompt we inherited from DOS.

The most successful way for me to recruit a new person into the ranks of happy PowerShell users is by example. When presented with an automation problem, my first suggestion is, “Let’s try to do it in PowerShell.” A couple of lines later, the usual reaction is, “Wow, can it really do that?” A short and sweet command can win many hearts for PowerShell. Hopefully, the examples in this book will win your heart too—it is my goal to share with you the joy of working with a modern automation environment that will change your professional life forever.

## What’s in This Book?

When working on the book, I separated it into sections, and this distinction may be helpful to you too.

The first section includes Chapters 1 through 14. It introduces the shell’s object-oriented features and its means for providing basic abstraction like script blocks, functions, scripts, aliases, and providers. It is best to read those chapters in sequence to get a solid understanding of the advanced techniques that allow you to work with the shell’s type system, get to know its security infrastructure, learn about the documentation facilities, and start using important debugging techniques.

The second section spans Chapters 15 to 21. The text uses advanced scripting techniques to work with .NET, COM, and WMI objects. This section’s goal is twofold. On one hand, you’ll learn how to work with external code and employ those techniques to automate almost any task on a Windows system. On the other hand, you are solving real-world problems that are very likely to land on your plate some time soon. The chapters describe how to start and stop processes

and services, work with text-based I/O, use performance counters to monitor your system, download and upload files to the web, send e-mail, automate programs via their COM interfaces, and get information and manage operating system objects through WMI. The chapters can be read in random order, and you can easily use them for reference.

The last section of the book is about tools and add-ons that you can use to extend the shell and become even more productive. Many programs can get much easier to use after extending them with a tool that provides an important missing feature. The book shows how you can use freely available open source tools to boost your productivity. An important part of my secret in winning over new people for PowerShell lies in the fact that I always do my demonstrations with PowerShell Community Extensions and PowerTab installed. Those two tools are described in Chapters 21 and 23 respectively.

## **Do I Need Prior Programming Experience?**

The book is written by a programmer—does that mean you need to be a programmer to read it? Not at all! I have done my best to make the text easy to digest for just about anyone with a little prior scripting experience. The PowerShell scripting language bears a lot of similarities to other scripting languages, and users will be able to draw parallels between it and VBScript, JScript, batch scripts, UNIX shell scripts, and maybe even Perl or Python. People new to scripting should be able to just use any of the working code samples as they are, without needing to learn the details or any of the inner workings of the code.