Sample Chapter

PowerShell
IN DEPTH
SECOND EDITION

Don Jones
Jeffery Hicks
Richard Siddaway
brief contents

PART 1 PowerShell Fundamentals .........................................................1

1 ■ Introduction 3  
2 ■ PowerShell hosts 7  
3 ■ Using the PowerShell help system 19  
4 ■ The basics of PowerShell syntax 33  
5 ■ Working with PSSnapins and modules 44  
6 ■ Operators 52  
7 ■ Working with objects 69  
8 ■ The PowerShell pipeline 106  
9 ■ Formatting 125

PART 2 PowerShell Management ............................................................143

10 ■ PowerShell Remoting 145  
11 ■ Background jobs and scheduling 181  
12 ■ Working with credentials 196  
13 ■ Regular expressions 206  
14 ■ Working with HTML and XML data 222
<table>
<thead>
<tr>
<th>Part</th>
<th>Brief Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>PSDrives and PSProviders 244</td>
</tr>
<tr>
<td>16</td>
<td>Variables, arrays, hash tables, and script blocks 260</td>
</tr>
<tr>
<td>17</td>
<td>PowerShell security 282</td>
</tr>
<tr>
<td>18</td>
<td>Advanced PowerShell syntax 296</td>
</tr>
<tr>
<td><strong>PART 3</strong></td>
<td><strong>PowerShell Scripting and Automation</strong> ..........315</td>
</tr>
<tr>
<td>19</td>
<td>PowerShell’s scripting language 317</td>
</tr>
<tr>
<td>20</td>
<td>Basic scripts and functions 332</td>
</tr>
<tr>
<td>21</td>
<td>Creating objects for output 344</td>
</tr>
<tr>
<td>22</td>
<td>Scope 361</td>
</tr>
<tr>
<td>23</td>
<td>PowerShell workflows 376</td>
</tr>
<tr>
<td>24</td>
<td>Advanced syntax for scripts and functions 406</td>
</tr>
<tr>
<td>25</td>
<td>Script modules and manifest modules 427</td>
</tr>
<tr>
<td>26</td>
<td>Custom formatting views 439</td>
</tr>
<tr>
<td>27</td>
<td>Custom type extensions 451</td>
</tr>
<tr>
<td>28</td>
<td>Data language and internationalization 465</td>
</tr>
<tr>
<td>29</td>
<td>Writing help 477</td>
</tr>
<tr>
<td>30</td>
<td>Error handling techniques 484</td>
</tr>
<tr>
<td>31</td>
<td>Debugging tools and techniques 496</td>
</tr>
<tr>
<td>32</td>
<td>Functions that work like cmdlets 520</td>
</tr>
<tr>
<td>33</td>
<td>Tips and tricks for creating reports 539</td>
</tr>
<tr>
<td><strong>PART 4</strong></td>
<td><strong>Advanced PowerShell</strong> ...........................................549</td>
</tr>
<tr>
<td>34</td>
<td>Working with the Component Object Model (COM) 551</td>
</tr>
<tr>
<td>35</td>
<td>Working with .NET Framework objects 560</td>
</tr>
<tr>
<td>36</td>
<td>Accessing databases 573</td>
</tr>
<tr>
<td>37</td>
<td>Proxy functions 581</td>
</tr>
<tr>
<td>38</td>
<td>Building a GUI 593</td>
</tr>
<tr>
<td>39</td>
<td>WMI and CIM 613</td>
</tr>
<tr>
<td>40</td>
<td>Working with the web 640</td>
</tr>
<tr>
<td>41</td>
<td>Desired State Configuration 656</td>
</tr>
</tbody>
</table>
As of this writing, Windows PowerShell is on to its seventh year of existence and in its fourth major release, with a fifth version in preview. In that time, it’s changed the way people look at administering many Microsoft, and even some non-Microsoft, products. Although the graphical user interface (GUI) will always be an important part of administration in many ways, PowerShell has given administrators options: Use an easy, intuitive GUI; manage from a rich, interactive command-line console; or fully automate with a simple scripting language. We’re delighted that so many administrators have started using PowerShell, and we’re honored that you’ve chosen this book to further your own PowerShell education.

1.1 Who this book is for

We wrote this book for system administrators, not developers. In the Microsoft world, administrators go by the catchall title “IT professional” or “IT pro” and that’s who we had in mind. As such, we assume you’re not a full-time programmer, although
if you have some programming or scripting experience it'll make certain parts of Power-
Shell easier to learn.

We assume you’re primarily interested in automating various administrative tasks
and processes, or at least being more efficient, but we don’t make any assumptions
about the products with which you work. You may be an Exchange Server administra-
tor, or maybe SharePoint or SQL Server is your thing. Perhaps you manage Active
Directory, or you’re in charge of file servers. You may even manage a Citrix or VMware
environment (yes, they can be managed by PowerShell). It doesn’t matter, because
what we’ll focus on in this book is the core technologies of PowerShell itself: the tech-
niques and features you’ll need to use no matter what products you’re administering.
We do use Active Directory in a few examples, but every technique, pattern, practice,
and trick we show you will apply equally well, no matter where you’ve chosen to use
PowerShell.

1.2 What this book will teach you

You can certainly read this book cover to cover, and we’ve tried to arrange the chap-
ters in a logical progression. But in the end, we intend for this book to be a reference.
Need to figure out PowerShell Remoting? Skip to that chapter. Confused about how
commands pipe data from one to another? We’ve written a chapter for that. Need to
access a database from within a PowerShell script? There’s a chapter for that.

We’ve no intention of making you a programmer—we don’t claim to be program-
mers—we all have backgrounds as IT pros. Yes, PowerShell can support some robust
scripts, but you can also accomplish a lot by running commands. If you have program-
ming experience, it’ll serve you well, and you may be tempted to approach PowerShell
more as a scripting language, which is fine. If you’ve never scripted or programmed a
single line of code, you’ll probably see PowerShell as a pure command-line interface,
where you run commands to make stuff happen, and that’s fine, too. Either way you
win because you get to automate your tedious, repetitive work. The other winning fea-
ture is that what you learn by using PowerShell at the command line is directly usable
when you start writing scripts—there’s no wasted learning with PowerShell.

1.3 What this book won’t teach you

We assume you’re already an experienced administrator and that you’re familiar with
the inner workings of whatever technology you manage. We aren’t going to teach you
what an Active Directory user account is, or what an Exchange mailbox does, or how
to create a SharePoint site. PowerShell is a tool that lets you accomplish administrative
tasks, but like any tool it assumes you know what you’re doing.

To use a noncomputer analogy, PowerShell is a hammer, and this book will teach
you how to swing that hammer and not smash your thumb. We won’t teach you about
building houses, though—we assume you already know how to do that and that you’re
looking for a more efficient way to do it than pounding nails with a rock.
1.4 Where we drew the line

It’s safe to say that PowerShell can’t do everything for you. You’ll find some things with which it’s completely incapable of helping, as with any technology. But you’ll also find tasks for which PowerShell works well. And you’ll encounter that weird middle ground where you could do something in PowerShell, but to do it you’d have to go beyond the strict boundaries of what PowerShell is. For example, PowerShell doesn’t natively contain a way to map a network printer.

**NOTE** There is a PrintManagement module containing an Add-Printer cmdlet, but it’s part of Windows (specifically Windows 8/2012 and later) rather than PowerShell.

You could instantiate a Component Object Model (COM) object to accomplish the task from within PowerShell, but it has nothing to do with PowerShell. Instead, it’s the shell giving you a way to access completely external technologies. In these cases (which are becoming increasingly rare in the latest version of Windows), we’ll only say, “You can’t do that in PowerShell yet.” We know our statement isn’t 100% true, but we want to keep this book focused on what PowerShell is and what it does natively. If we turn this book into “everything you can do with PowerShell natively, plus all the external stuff like .NET and COM and so on that you can get to from PowerShell,” it’d grow to 7,000 pages in length and we’d never finish.
That said, we’re including material in the book on using some of these external technologies, along with some guidance on where you can find resources to educate yourself on them more completely if you’ve a mind to do so.

1.5 Beyond PowerShell

PowerShell is a lot like the Microsoft Management Console (MMC), with which you’re probably familiar. On its own, it’s useless. Both the MMC and PowerShell only become useful when you add extensions, which in the MMC would be “snap-ins,” and in PowerShell would be either a “snap-in” or a “module.” Those extensions give you access to Exchange, Active Directory, SharePoint, SQL Server, and so on. The later versions of Windows (Windows 8 and later) ship with over 50 additional modules, not counting the Remote Server Administration Tools (RSAT) modules. This additional functionality is blurring the boundaries of PowerShell. The thing to remember is that in this book we’re concentrating on the core of PowerShell so that you understand how it works. Using the other modules will become obvious once you understand PowerShell itself.
Understand that the folks at Microsoft who write PowerShell don’t write the extensions. They provide some tools and rules for the developers who do create extensions, but their job is to create the core PowerShell stuff. Extensions are made by other product teams: The Exchange team makes the Exchange PowerShell extension, the Active Directory team makes its extension, and so on. If you’re looking at a particular extension and don’t like what you see, blame the product team that produced it, not PowerShell. If you’d like to administer something—maybe Windows Internet Name
Service (WINS) Server, for example—and PowerShell has no way to administer it, it’s not the PowerShell team’s fault. Blame the owners of the technology you’re trying to work with, and encourage them to get on board and produce a PowerShell extension for their product.

This division of labor is one reason why we’re keeping this book focused on the core of PowerShell. That core is what you’ll use no matter what extensions you end up deploying to achieve your administrative goals.

1.6 Ready?

Okay, that’s enough of an introduction. If you want to follow along, make sure you have PowerShell v4 installed on a Windows 7 or later client. You’ll also find it useful to have a test server running PowerShell v4, ideally on Windows Server 2012 R2.

**NOTE** The examples and code in this book will work with PowerShell v5 if you have that installed. The major new feature in PowerShell v5 is software management through the OneGet module.

Now, pick a chapter and jump in.
A Windows admin using PowerShell every day may not have the time to search the net every time he or she hits a snag. Wouldn’t it be great to have a team of seasoned PowerShell experts ready to answer even the toughest questions? That’s what you get with this book.

PowerShell in Depth, Second Edition is the go-to reference for administrators working with Windows PowerShell. Every major technique, technology, and tactic is carefully explained and demonstrated, providing a hands-on guide to almost everything an admin would do in the shell. Written by PowerShell MVPs Don Jones, Jeffrey Hicks, and Richard Siddaway, each valuable technique was developed and thoroughly tested, so you’ll be able to consistently write production-quality, maintainable scripts while saving hours of time and effort.

What’s Inside

- Automating tasks
- Packaging and deploying scripts
- Introduction to Desired State Configuration
- PowerShell security
- Covers PowerShell version 3 and later

This book assumes you know the basics of PowerShell.

Don Jones, Jeffery Hicks, and Richard Siddaway are Microsoft MVPs, trainers, and administrators. Collectively, they’ve authored nearly three dozen books on PowerShell and Windows administration.

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit manning.com/PowerShellinDepthSecondEdition

—Lincoln Bovee, Protolabs, Inc.

“Three MVPs thinking deeply about PowerShell. What more can you want?”
—David Moravec
Mainstream Technologies

“Immediately applicable to real-world problems.”
—Jim Gray, Navigant Consulting

“An excellent way to gain more PowerShell wisdom!”
—Arthur Zubarev
Complete Innovations, Inc.