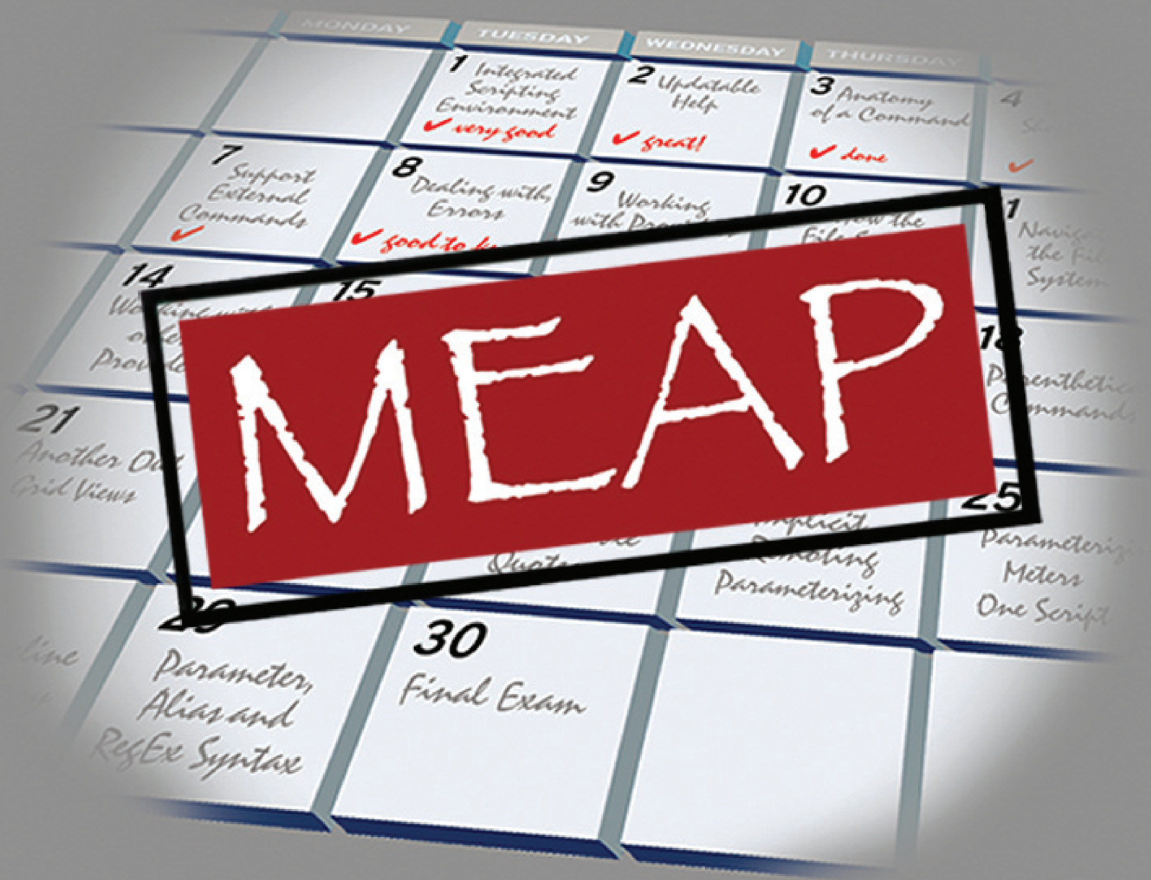
Learn WINDOWS POWERSHELL IN A MONTH OF LUNCHES

THIRD EDITION



DON JONES
JEFFERY D. HICKS

MANNING

MEAP Edition
Manning Early Access Program

Learn Windows PowerShell in a Month of Lunches

Third Edition

Version 2

welcome

Thank you for purchasing this MEAP early access copy for this third edition of *Learn Windows PowerShell in a Month of Lunches, Third Edition*. It's been revised to cover versions 2 through 5, including the new package management features and updated hands-on exercises. You'll be glad you chose to spend your next month of lunches with us.

To get the most out of this book, you'll find experience with Windows administration helpful, but we don't assume that you have any programming or scripting experience.

Using the Month of Lunches philosophy, with its frequent labs and exercises and practical, immediately applicable instruction, you'll learn PowerShell from the beginning. Each lesson will take you an hour or less.

This updated book covers PowerShell features that run on Windows 7, Windows Server 2008 R2, and later. This edition is appropriate for PowerShell version 3 and later. There is coverage for new PowerShell version 5 features like PowerShellGet; however, PowerShell fundamentals are unchanged. So if you are just getting started with PowerShell it doesn't matter what version you are using, this book is for you.

Thanks!

—Don Jones and Jeffrey Hicks

brief contents

preface

We've been teaching and writing about Windows PowerShell for a long time. When Don began contemplating the first edition of this book, he realized that most PowerShell writers and teachers—including himself—were forcing our students to approach the shell as a kind of programming language. Most PowerShell books are into "scripting" by the third or fourth chapter, yet more and more PowerShell students were backing away from that programming-oriented approach. Those students wanted to use the shell as a shell, at least at first, and we simply weren't delivering a learning experience that matched that desire.

So he decided to take a swing at it. A blog post on WindowsITPro.com proposed a table of contents for this book, and ample feedback from the blog's readers fine-tuned it into the book you're about to read. He wanted to keep each chapter short, focused, and easy to cover in a short period of time—because we know administrators don't have a lot of free time, and often have to learn on the fly. When PowerShell v3 came out, it was obviously a good time to update the book, and Don turned to long-time collaborator and fellow MVP Jeffery Hicks to help out.

We both wanted a book that would focus on PowerShell itself, and not on the myriad technologies that PowerShell touches, like Exchange Server, SQL Server, System Center, and so on. We truly feel that by learning to use the shell properly, you can teach yourself to administer all of those "PowerShell-ed" server products. So this book tries to focus on the core of using PowerShell. Even if you're also using a "cookbook" style of book, which provides ready-to-use answers for specific administrative tasks, this book will help you understand what those examples are doing. That understanding will make it easier to modify those examples for other purposes, and eventually to construct your own commands and scripts from scratch.

We hope this book won't be the only PowerShell education that you pursue. We've also co-authored *Learn PowerShell Toolmaking in a Month of Lunches*, which offers the same day-at-a-time approach to learning PowerShell's scripting and tool-creation capabilities. You can also find videos we've produced on YouTube, read articles we've authored for sites like Petri.com and Windows IT Pro, not to mention courses from Pluralsight.

If you need any further help, we encourage you to log on to www.PowerShell.org. We both answer questions in several of the discussion forums there, and we'd be happy to try and get you out of whatever you're stuck on. The site is also a great portal into the robust and active PowerShell community—you can learn about free e-books the in-person PowerShell and

DevOps Summit, and about all of the regional and local user groups and PowerShell-related events that happen throughout the year. Get involved—it's a great way to make PowerShell a more powerful part of your career.

Enjoy—and good luck with the shell.

about this book

Most of what you'll need to know about this book is covered in chapter 1, but there are a few things that we should mention up front.

First of all, if you plan to follow along with our examples and complete the hands-on exercises, you'll need a virtual machine or computer running Windows 8.1 or Windows Server 2012, or later. We cover that in more detail in chapter 1. You can get by with Windows 7, but you'll miss out on a few of the hands-on labs.

Second, be prepared to read this book from start to finish, covering each chapter in order. Again, this is something we'll explain in more detail in chapter 1, but the idea is that each chapter introduces a few new things that you will need in subsequent chapters. You really shouldn't try to push through the whole book – stick with the one chapter per day approach. The human brain can only absorb so much information at once, and by taking on PowerShell in small chunks, you'll actually learn it a lot faster and more thoroughly.

Third, this book contains a lot of code snippets. Most of them are quite short, so you should be able to type them quite easily. In fact, we recommend that you do type them, since doing so will help reinforce an essential PowerShell skill: accurate typing! Longer code snippets are given in listings and are available for download from the book's page on the publisher's website at https://www.manning.com/books/learn-windows-powershell-in-a-month-of-lunches-third-edition.

That said, there are a few conventions that you should be aware of. Code will always appear in a special font, just like this example:

```
Get-WmiObject –class Win32_OperatingSystem
[CA]–computerName SERVER-R2
```

That example also illustrates the line-continuation character used in this book. It indicates that those two lines should actually be typed as a single line in PowerShell. In other words, don't hit Enter or Return after Win32_OperatingSystem—keep right on typing. PowerShell allows for very long lines, but the pages of this book can only hold so much.

Sometimes, you'll also see that code font within the text itself, such as when we write `Get-Command`. That just lets you know that you're looking at a command, parameter, or other element that you would actually type within the shell.

Fourth is a tricky topic that we'll bring up again in several chapters: the backtick character (`). Here's an example:

```
Invoke-Command –scriptblock { Dir } `
-computerName SERVER-R2,localhost
```

The character at the end of the first line isn't a stray bit of ink—it's a real character that you would type. On a U.S. keyboard, the backtick (or grave accent) is usually near the upper left, under the Escape key, on the same key as the tilde character (~). When you see the backtick in a code listing, type it exactly as is. Furthermore, when it appears at the end of a line—as in the preceding example—make sure that it's the very last character on that line. If you allow any spaces or tabs to appear after it, the backtick won't work correctly, and neither will the code example.

Finally, we'll occasionally direct you to Internet resources. Where those URLs are particularly long and difficult to type, we've replaced them with Manning-based shortened URLs that look like http://mng.bz/S085 (you'll see that one in chapter 1).

Author Online

The purchase of *Learn Windows PowerShell in a Month of Lunches, Third Edition* includes access to a private forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the authors and other users. To access and subscribe to the forum, point your browser to https://www.manning.com/books/learn-windows-powershell-in-a-month-of-lunches-third-edition and click the Author Online link. This page provides information on how to get on the forum once you are registered, what kind of help is available, and the rules of conduct in the forum.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the authors can take place. It's not a commitment to any specific amount of participation on the part of the authors, whose contribution to the book's forum remains voluntary (and unpaid). We suggest you try asking the authors some challenging questions, lest their interest stray!

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

about the authors

Don Jones is a multiple-year recipient of Microsoft's prestigious Most Valuable Professional (MVP) Award for his work with Windows PowerShell. For five years, he wrote the Windows PowerShell column for *Microsoft TechNet Magazine,* currently blogs at PowerShell.org, and authored the "Decision Maker" column and blog for *Redmond Magazine.* Don is a prolific technology author and has published more than a dozen print books since 2001. He's now a Curriculum Director for IT Ops content at Pluralsight, an online video training platform. Don's first Windows scripting language was KiXtart, going back all the way to the mid-1990s. He quickly graduated to VBScript in 1995 and was one of the first IT pros to start using early releases of a new Microsoft product code-named "Monad"—which later became Windows PowerShell. Don lives in Las Vegas and, when it gets too hot there, near Duck Creek Village in Utah.

Jeffery Hicks is an IT veteran with over 25 years of experience, much of it spent as an IT infrastructure consultant specializing in Microsoft server technologies with an emphasis in automation and efficiency. He is a multi-year recipient of the Microsoft MVP Award in Windows PowerShell. He works today as an independent author, trainer and consultant. He has taught and presented on PowerShell and the benefits of automation to IT Pros all over the world. Jeff has written for numerous online sites and print publications, is a contributing editor at Petri.com, a Pluralsight author, and a frequent speaker at technology conferences and user groups. You can keep up with Jeff at his blog, http://jdhitsolutions.com/blog and on Twitter (@JeffHicks).

acknowledgments

Books simply don’t write, edit, and publish themselves. Don would like to thank everyone at Manning Publications who decided to take a chance on a very different kind of book for Windows PowerShell, and who worked so hard to make the first edition of this book happen. Jeff would like to thank Don for inviting him along for the ride, and all the PowerShell community for their enthusiasm and support. Don and Jeff are both grateful to Manning for allowing them to continue the “Month of Lunches” series with this third edition.

Thanks also to the following peer reviewers who read the manuscript during its development and provided feedback: Bennett Scharf, Dave Pawson, David Moravec, Keith Hill, and Rajesh Attaluri; also to James Berkenbile and Trent Whiteley for their technical review of the manuscript and code during production.

1 Before you begin

We've been teaching Windows PowerShell since version 1 was released in 2006. Back then, most of the folks using the shell were experienced VBScript users, and they were eager to apply their VBScript skills to learning PowerShell. As a result, we and the other folks who taught the shell, wrote books and articles, and so forth, all adopted a teaching style that takes advantage of prior programming or scripting skills.

But since late 2009, a shift has occurred. More and more administrators who *don't* have prior VBScript experience have started trying to learn the shell. All of a sudden, our old teaching patterns didn't work as well, because we had focused on scripting and programming. That's when we realized that PowerShell isn't a scripting language. It's a command-line shell where you run command-line utilities. Like all good shells, it has scripting capabilities, but you don't have to use them, and you certainly don't have to *start* with them. We started changing our teaching patterns, beginning with the many conferences we speak at each year. Don also implemented these changes into his instructor-led training courseware.

This book is the result of that process, and it's the best that we've yet devised to teach PowerShell to someone who might not have a scripting background (although it certainly doesn't hurt if you do). But before we jump into the instruction, let's set the stage for you.

1.1 Why you can't afford to ignore PowerShell

Batch. KiXtart. VBScript. Let's face it, Windows PowerShell isn't exactly Microsoft's (or anyone else's) first effort at providing automation capabilities to Windows administrators. We think it's valuable to understand why you should care about PowerShell, because when you do, you'll feel comfortable that the time you commit to learning PowerShell will pay off. Let's start by considering what life was like before PowerShell came along, and look at some of the advantages of using this shell.

1.1.1 Life without PowerShell

Windows administrators have always been happy to click around in the graphical user interface (GUI) to accomplish their chores. After all, the GUI is largely the whole point of Windows—the operating system isn't called "Text," after all. GUIs are great because they enable you to discover what you can do. Don remembers the first time he opened Active Directory Users and Computers. He hovered over icons and read tooltips, pulled down menus, and right-clicked on things, all to see what was available. GUIs make learning a tool easier. Unfortunately, GUIs have zero return on that investment. If it takes you five minutes to create a new user in Active Directory (and assuming you're filling in a lot of the fields, that's a reasonable estimate), you'll never get any faster than that. One hundred users will take five hundred minutes—there's no way, short of learning to type and click faster, to make the process go any quicker.

Microsoft has tried to deal with that problem a bit haphazardly, and VBScript was probably its most successful attempt. It might have taken you an hour to write a VBScript that could import new users from a CSV file, but once you'd invested that hour, creating users in the future would take only a few seconds. The problem with VBScript is that Microsoft didn't make a wholehearted effort in supporting it. Microsoft had to remember to make things VBScript-accessible, and when developers forgot (or didn't have time), you were stuck. Want to change the IP address of a network adapter using VBScript? OK, you can. Want to check its link speed? You can't, because nobody remembered to hook that up in a way that VBScript could get to. Sorry. Jeffrey Snover, the architect of Windows PowerShell, calls this "the last mile." You can do a lot with VBScript (and other, similar technologies), but it tends to let you down at some point, never getting you through that "last mile" to the finish line.

Windows PowerShell is an express attempt on Microsoft's part to do a better job, and to get you through the last mile. And it's been a successful attempt so far – there are dozens of product groups within Microsoft who've adopted PowerShell, an extensive ecosystem of third parties who are depending on it, and a global community of experts and enthusiasts who are pushing the PowerShell envelope every day.

1.1.2 Life with PowerShell

Microsoft's goal for Windows PowerShell is to build 100% of a product's administrative functionality in the shell. Microsoft continues to build GUI consoles, but those consoles are executing PowerShell commands behind the scenes. That approach forces the company to make sure that every possible thing you can do with the product is accessible through the shell. If you need to automate a repetitive task or create a process that the GUI doesn't enable well, you can drop into the shell and take full control for yourself.

A number of Microsoft products have already adopted this approach, including Exchange Server 2007 and beyond, SharePoint Server 2010 and later, many of the System Center products, Office 365, and many components of Windows itself. Going forward, more and more products and Windows components will follow this pattern. Windows Server 2012, which was where PowerShell v3 was introduced, is almost completely managed from PowerShell—or by a

GUI sitting atop PowerShell. That's why you can't afford to ignore PowerShell—over the next few years, it'll become the basis for more and more administration. In fact, it's already become the foundation for numerous higher-level technologies, including Desired State Configuration (DSC), PowerShell Workflow, and much more. PowerShell is everywhere!

Ask yourself this question: If you were in charge of a team of IT administrators (and perhaps you are), who would you want in your senior, higher-paying positions? Administrators who need several minutes to click their way through a GUI each time they need to perform a task, or ones who can perform tasks in a few seconds after automating them? We already know the answer from almost every other part of the IT world. Ask a Cisco administrator, or an AS/400 operator, or a Unix administrator. The answer is, "I'd rather have the person who can run things more efficiently from the command line." Going forward, the Windows world will start to split into two groups: administrators who can use PowerShell, and those who can't. As Don famously said at Microsoft's TechEd 2010 conference, "your choice is 'learn PowerShell,' or 'would you like fries with that?'"

We're glad *you've* decided to learn PowerShell.

1.2 Is this book for you?

This book doesn't try to be all things to all people. In fact, Microsoft's PowerShell team loosely defines three audiences who use PowerShell:

- Administrators who primarily run commands and consume tools written by others
- Administrators who combine commands and tools into more complex processes, and perhaps package those as tools that less-experienced administrators can use
- Administrators and developers who create reusable tools and applications

This book is designed primarily for the first audience. We think it's valuable for anyone, even a developer, to understand how the shell is used to run commands. After all, if you're going to create your own tools and commands, you should know the patterns that the shell uses, as they allow you to make tools and commands that work as well as they can within the shell.

If you're interested in creating scripts to automate complex processes, such as new user provisioning, then you'll absolutely see how to do that by the end of this book. You'll even see how to get started on creating your own commands that other administrators can use. But this book won't plumb the depths of everything that PowerShell can possibly do. Our goal is to get you using the shell and being effective with it in a production environment.

We'll also show you a couple of ways to use PowerShell to connect to external management technologies—Windows Management Instrumentation (WMI) and regular expressions are the two examples that come quickly to mind. For the most part, we're going to introduce only those technologies and focus on how PowerShell connects to them. Those topics deserve their own books (and have them; we'll provide recommendations when we get there); we'll concentrate solely on the PowerShell side of things. We'll provide suggestions for further exploration if you'd like to pursue those technologies further on your own. In short, this book isn't meant to be the last thing you use to learn about PowerShell, but it's definitely designed to be a great first step.

1.3 How to use this book

The idea behind this book is that you'll read one chapter each day. You don't have to read it during lunch, but each chapter should take you only about 40 minutes to read, giving you an extra 20 minutes to gobble down the rest of your sandwich and practice what the chapter showed you.

THE MAIN CHAPTERS

Of the chapters in this book, chapters 2 through 25 contain the main content, giving you 24 days' worth of lunches to look forward to. This means you can expect to complete the main content of the book in about a month. Try to stick with that schedule as much as possible, and don't feel the need to read extra chapters in a given day. It's more important that you spend some time practicing what each chapter shows you, because using the shell will help cement what you've learned. Not every chapter will require a full hour, so sometimes you'll be able to spend some additional time practicing (and eating lunch) before you have to get back to work. We find that a lot of people actually learn more quickly when they stick with just one chapter a day, because it gives your brain time to mull over the new ideas, and gives you time to practice them on your own. Don't rush it, and you may find yourself moving more quickly than you thought possible.

HANDS-ON LABS

Most of the main content chapters include a short lab for you to complete. You'll be given instructions, and perhaps a hint or two. The answers for all chapter exercises can be found in Appendix B. But try your best to complete each lab without looking at the answers.

CODE SAMPLES

Throughout the book you will encounter a number of code listings. These are longer PowerShell examples. But don't feel you need to copy them. If you head to Manning.com and find the page for this book, you should see a link to download all of the code listings.

SUPPLEMENTARY MATERIALS

Don's YouTube channel, YouTube.com/PowerShellDon, contains a bunch of free videos that he made for the original edition of this book – and they're all still 100% applicable. They're a great way to get some short, quick demos. He also hosts videos from recorded conference workshops and more, and they're all worth a look. We'll also suggest the PowerShell.org channel, YouTube.com/powershellorg, which contains a ton of video content. You'll find recorded sessions from the PowerShell + DevOps Global Summit events, online community webinars, and a lot more. All free!

Jeff does a lot of writing for Petri.com and you'll find a huge collection of content covering all sorts of PowerShell content. You might also see if Jeff has anything new on his YouTube channel, YouTube.com/jdhitsolutions.

FURTHER EXPLORATION

A few chapters in this book only skim the surface of some cool technologies, and we'll end those chapters with suggestions for how you might explore those technologies on your own. We'll point out additional resources, including free stuff that you can use to expand your skill set as the need arises.

ABOVE AND BEYOND

As we learned PowerShell, there were often times when we wanted to go off on a tangent and explore why something worked the way it did. We didn't learn a lot of extra practical skills that way, but we did gain a deeper understanding of what the shell is and how it works. We've included some of that tangential information throughout the book in sections labeled "Above and beyond." None of those will take you more than a couple of minutes or so to read, but if you're the type of person who likes to know why something works the way it does, they can provide some fun additional facts. If you feel those sections might distract you from the practical stuff, ignore them on your first read-through. You can always come back and explore them later when you've mastered the chapter's main material.

1.4 Setting up your lab environment

You're going to be doing a lot of practicing in Windows PowerShell throughout this book, and you'll want to have a lab environment to work in—please don't practice in your company's production environment.

All you'll need to run most of the examples in this book—and to complete all of the labs—is a copy of Windows that has PowerShell v3 or later installed. We suggest Windows 8 or later, or Windows Server 2012 or later, which both come with PowerShell v4. Note that PowerShell might not exist on certain editions of Windows, such as "Starter" editions. If you're going to play with PowerShell, you'll have to invest in a version of Windows that has it. Also note that some of the labs rely on functionality that was new in Windows 8 and Windows Server 2012, so if you're using something older, things might work differently. At the start of each lab, we'll tell you what operating system you need in order to complete the lab. We do recommend having at least a Windows 8 or Windows Server 2012 computer to play with—even if it's in a virtual machine. And if you have a newer version, that's perfect – everything in here should work the same.

Keep in mind that, throughout this book, we're assuming you'll be working on a 64-bit operating system, also referred to as an "x64" operating system. As such, it comes with two copies of Windows PowerShell and the graphically oriented Windows PowerShell ISE. In the Start menu (or, in Windows 8, the Start screen), the 64-bit versions of these are listed as "Windows PowerShell" and "Windows PowerShell ISE." The 32-bit versions are identified by an "(x86)" in the shortcut name, and you'll also see "(x86)" in the window's title bar when running those versions. If you're on a 32-bit operating system, you'll have only the 32-bit version of PowerShell, and it won't specifically say "(x86)."

The examples in this book are based on the 64-bit versions of PowerShell and the ISE. If you're not using those, you may sometimes get slightly different results than ours when running examples, and a few of the labs might not work properly. The 32-bit versions are primarily provided for backward compatibility. For example, some shell extensions are available only in 32-bit flavors and can be loaded into only the 32-bit (or "x86") shell. Unless you need to use such an extension, we recommend using the 64-bit shell when you're on a 64-bit operating system. Microsoft's investments going forward are primarily in 64-bit; if you're stuck with a 32-bit operating system, unfortunately that's going to hold you back.

TIP You should be able to accomplish everything in this book with a single computer running PowerShell, although some stuff gets more interesting if you have two or three computers, all in the same domain, to play with. We've used CloudShare.com as an inexpensive way to spin up several virtual machines in the cloud—if such a scenario interests you, look into that service, or something like it. Note that CloudShare.com isn't currently available in all countries. Another possibility if you are running Windows 8 or later is to use the Hyper-V feature and run a few virtual machines there.

1.5 Installing Windows PowerShell

Windows PowerShell v3 has been available for most versions of Windows since the release of Windows Server 2008, Windows Server 2008 R2, Windows 7, and later versions. Windows Vista is not supported, but it can still run v2. The shell is preinstalled only on the most recent versions of Windows; it must be manually installed on older versions. PowerShell v4 is available for Windows 7 and later and Windows Server 2008R2 or later, although those versions of Windows don't have as many components that are "hooked up" to PowerShell, which is why we recommend Windows 8 or Windows Server 2012 as minimum versions. And while PowerShell v4 isn't the latest version of the shell, that or anything later will suffice for what we're going to show you in this book.

TIP You should check on your version of PowerShell: Open the PowerShell console, type `$PSVersionTable`, and hit Enter. If you get an error, or if the output doesn't say "PSVersion 4.0," then you don't have PowerShell v4.

If you want to check the latest available version of PowerShell, or download it, go to http://msdn.microsoft.com/powershell. This is the official PowerShell home page, and will have links to the latest Windows Management Framework (WMF) installer, which is what installs PowerShell and its related technologies. Again, because this book is covering entry-level stuff, you'll find that not much has changed from v3, but it's always fun to have the latest version to play with.

PowerShell has two application components: the standard, text-based console host (PowerShell.exe) and the more visual Integrated Scripting Environment (ISE; PowerShell _ISE.exe). We use the text-based console most of the time, but you're welcome to use the ISE if you prefer.

NOTE The PowerShell ISE isn't preinstalled on server operating systems. If you want to use it, you'll need to go in to Windows Features (using Server Manager) and manually add the ISE feature (you can also open the PowerShell console and run `Add-WindowsFeature powershell-ise`). The ISE isn't available at all on server installations that don't have the full GUI (for example, Server Core or Nano Server).

Before you go any further, take a few minutes to customize the shell. If you're using the text-based console host, we strongly recommend that you change the font it uses to the Lucida fixed-width font instead of the default console font. The default font makes it difficult to distinguish some of the special punctuation characters that PowerShell uses. Follow these steps to customize the font:

1. Click the control box (that's the PowerShell icon in the upper left of the console window) and select Properties from the menu.
2. In the dialog box that appears, browse through the various tabs to change the font, window colors, window size and position, and so forth.

TIP We strongly recommend you make sure that both the Window Size and Screen Buffer have the same Width values.

Your changes will apply to the default console, meaning they'll stick around when you open new windows.

1.6 Contacting us

We're passionate about helping folks like you learn Windows PowerShell, and we try to provide as many different resources as we can. We also appreciate your feedback because that helps us come up with ideas for new resources that we can add to the site, and ways to improve future editions of this book. You can reach Don on Twitter @concentratedDon, or Jeff @JeffHicks. We also both hang out in the forums of PowerShell.org if you have PowerShell questions. PowerShell.org is also a wonderful place for more resources, including free eBooks, an in-person annual conference, free webinars, and tons more. We help run the organization, and we can't recommend it highly enough as a place to continue your PowerShell education once you've finished this book.

1.7 Being immediately effective with PowerShell

"Immediately effective" is a phrase we've made our primary goal for this entire book. As much as possible, we'll try to have each chapter focus on something that you could use in a real production environment, right away. That means we'll sometimes gloss over some details in the beginning, but when necessary we promise to circle back and cover those details at the right time. In many cases, we had to choose between hitting you with 20 pages of theory first, or diving right in and accomplishing something without explaining all the nuances, caveats, and details. When those choices came along, we almost always chose to dive right in, with the goal of making you *immediately effective*. But all of those important details and nuances will still be explained at a different time in the book.

OK, that’s enough background. It’s time to start being immediately effective. Your first lunch lesson awaits.