POWERSHELL 7

FOR IT PROFESSIONALS

THOMAS LEE

WILEY

Table of Contents

- Cover
- Title Page
- Foreword
- Introduction
- Chapter 1: Setting Up a PowerShell 7 Environment
  - What Is New in PowerShell 7
  - Systems Used in This Book and Chapter
  - Installing PowerShell 7
  - Installing and Configuring VS Code
  - Using the PowerShell Gallery
  - Creating a Local PowerShellGet Repository
  - Creating a Code-Signing Environment
  - Summary
- Chapter 2: PowerShell 7 Compatibility with Windows PowerShell
  - Examining PowerShell Modules
  - Introducing the Compatibility Solution
  - Things That Do Not Work with PowerShell 7
  - Summary
- Chapter 3: Managing Active Directory
  - Systems Used in This Chapter
  - Establishing a Forest Root Domain
  - Installing a Replica DC
  - Installing a Child Domain
  - Configuring a Cross-Forest Trust
  - Managing AD Users, Computers, and OUs

Adding Users to AD via a CSV
Configuring Just Enough Administration (JEA)
Summary
Chapter 4: Managing Networking
Systems Used in This Chapter
Configuring IP Addressing
Testing Network Connectivity
Installing the DHCP Service
Configuring DHCP Scopes
Configuring DHCP Failover
Configuring the DNS Service
Configuring DNS Zones and Resource Records
Summary
Chapter 5: Managing Storage
Systems Used in This Chapter
Managing Disks and Volumes
Managing NTFS Permissions
Managing Storage Replica
Managing Filestore Quotas
Managing File Screening
Summary
Chapter 6: Managing Shared Data
Systems Used in This Chapter
Setting Up and Securing an SMB File Server
Creating and Securing SMB Shares
Creating and Using an iSCSI Target
Setting Up a Clustered Scale-Out File Server
Summary

Chapter 7: Managing Printing
- Systems Used in This Chapter
- Installing and Sharing Printers
- Publishing a Printer in AD
- Changing the Spool Folder
- Printing a Test Page
- Creating a Printer Pool
- Summary

Chapter 8: Managing Hyper-V
- Systems Used in This Chapter
- Installing and Configuring Hyper-V
- Creating a Hyper-V VM
- Using PowerShell Direct
- Configuring VM Networking
- Configuring VM Hardware
- Implementing Nested Virtualization
- Using VM Checkpoints
- Using VM Replication
- Managing VM Movement
- Measuring VM Resource Usage
- Summary

Chapter 9: Using WMI with CIM Cmdlets
- Exploring WMI Namespaces
- Exploring WMI Classes
- Getting Local and Remote Objects
- Invoking WMI Methods
- Managing WMI Events
- Implementing Permanent WMI Event Handling

Summary
Chapter 10: Reporting
Systems Used in This Chapter
Reporting on AD Users and Computers
Managing Filesystem Reporting
Collecting Performance Information Using PLA
Reporting on PLA Performance Data
Creating a Performance Monitoring Graph
Creating a System Diagnostics Report
Reporting on Printer Usage
Creating a Hyper-V Status Report
Reviewing Event Logs
Summary
Index
Copyright
Dedication
About the Author
About the Technical Editor
Acknowledgments
End User License Agreement

List of Illustrations

Chapter 1
Figure 1.1: Systems used in this chapter
Figure 1.2: Viewing help information
Figure 1.3: Examining the installation folder
Figure 1.4: Viewing module paths

Figure 1.5: Viewing profile file locations

Figure 1.6: Viewing help information

Figure 1.7: Viewing PowerShell 7 module paths

Figure 1.8: Viewing PowerShell profile file locations

Figure 1.9: Installing VS Code on *DC1*

Figure 1.10: Updated taskbar

Figure 1.11: Count of modules available in the PowerShell Gallery

Figure 1.12: Count of available modules that support .NET core

Figure 1.13: NTFS-related modules

Figure 1.14: Commands in the NTFSSecurity module

Figure 1.15: Testing the Hello World module

Figure 1.16: Viewing configured repositories

Figure 1.17: Viewing the repository folder

Figure 1.18: Viewing the RKRepo modules

Figure 1.19: Viewing the certificate

Figure 1.20: Attempting to run the script

Figure 1.21: Signing a PowerShell script

Figure 1.22: Signing a PowerShell script with a trusted certificate

Figure 1.23: Running a signed PowerShell script

Figure 1.24: Testing a script's digital signature

Chapter 2

Figure 2.1: System used in this chapter

Figure 2.2: Creating a new module
Figure 2.3: Importing the *MyModule1* module
Figure 2.4: Creating and using a manifest
Figure 2.5: Viewing files in *MyModule1*
Figure 2.6: Using *MyModule2*
Figure 2.7: Viewing *MyModule2*
Figure 2.8: Using module autoload
Figure 2.9: Viewing the module analysis cache
Figure 2.10: Loading the ServerManager module
Figure 2.11: Viewing the module deny list

Chapter 3

Figure 3.1: Forests, domains, and hosts used in this chapter
Figure 3.2: Installing AD DS Domain Services
Figure 3.3: Promoting `DC1`
Figure 3.4: Viewing the DSE
Figure 3.5: Viewing the Reskit.Org forest
Figure 3.6: Viewing the Reskit.Org domain
Figure 3.7: Viewing DNS settings and configuration
Figure 3.8: Testing connectivity with `DC1`
Figure 3.9: Installing Windows features
Figure 3.10: Viewing DCs in the `Reskit.Org` domain
Figure 3.11: Viewing DNS settings and configuration
Figure 3.12: Verifying connectivity to `DC1`
Figure 3.13: Adding features to `UKDC1`

Figure 3.14: Viewing the `Reskit` forest

Figure 3.15: Viewing the child domain information

Figure 3.16: Installing the AD DS feature

Figure 3.17: Testing network connectivity

Figure 3.18: Viewing details of the Kapoho.Com forest

Figure 3.19: Adding a DNS conditional forwarder

Figure 3.20: Testing conditional DNS forwarding

Figure 3.21: Setting up a conditional forwarder on DC1

Figure 3.22: Viewing the Reskit.Org forest

Figure 3.23: Viewing the Kahopo.Com forest

Figure 3.24: Updating the ACL and viewing its details

Figure 3.25: Viewing users in Reskit.Org

Figure 3.26: Viewing updated user details

Figure 3.27: Viewing updated user details

Figure 3.28: Viewing group membership

Figure 3.29: Viewing computers in the AD

Figure 3.30: Viewing users to be added to AD

Figure 3.31: Creating new users

Figure 3.32: Viewing all users

Figure 3.33: Testing the session configuration file

Figure 3.34: Registering the session configuration file

Figure 3.35: Determining session capabilities

Figure 3.36: Checking on commands in the JEA session

Figure 3.37: Invoking a JEA-defined function

Figure 3.38: Counting DNS server commands available

Figure 3.39: Viewing the JEA `Transcripts` folder

Figure 3.40: Viewing a JEA transcript

Chapter 4

Figure 4.1: Systems used in this chapter

Figure 4.2: IP address details returned after checking the current state

Figure 4.3: Verifying the NIC IP address

Figure 4.4: Verifying the new IP configuration

Figure 4.5: Testing the TCP/IP stack

Figure 4.6: Verifying connectivity to `DC1`

Figure 4.7: Verifying SMB/LDAP connectivity to DC1

Figure 4.8: Verifying access to an Internet server

Figure 4.9: Installing the DHCP service

Figure 4.10: Checking the DHCP service

Figure 4.11: Reviewing IPv4 scopes

Figure 4.12: Checking DHCP operation

Figure 4.13: Installing the DHCP feature

Figure 4.14: Viewing authorized DHCP servers

Figure 4.15: Verbose output when creating DHCP failover

Figure 4.16: Viewing active DHCP leases

Figure 4.17: Viewing DHCP server statistics

Figure 4.18: Installing the DNS feature

Figure 4.19: Examining key DNS configuration settings

Figure 4.20: Starting NSLookup

Figure 4.21: Checking on recently created DNS zones

Figure 4.22: Checking on DNS RRs

Figure 4.23: Checking DNS name resolution on DC1

Chapter 5

Figure 5.1: Systems used in this chapter

Figure 5.2: Examining physical disks in SRV1

Figure 5.3: Examining the initialized disks in SRV1

Figure 5.4: Creating an F: volume

Figure 5.5: Creating a G: volume

Figure 5.6: Creating an H: volume

Figure 5.7: Viewing created volumes

Figure 5.8: Formatting volumes G: and H:

Figure 5.9: Viewing partitions on SRV1

Figure 5.10: Viewing partitions on SRV1

Figure 5.11: Viewing cmdlets in the NTFSSecurity module

Figure 5.12: Creating a folder and file

Figure 5.13: Viewing the ACL of the folder

Figure 5.14: Viewing the ACL of the file

Figure 5.15: Checking the sales group

Figure 5.16: Viewing the ACL of the folder
Figure 5.17: Viewing the ACL of the file
Figure 5.18: Viewing the F: drive
Figure 5.19: Viewing the content of SRV2
Figure 5.20: The result of adding SR to SRV1
Figure 5.21: Testing the SR configuration
Figure 5.22: Viewing topology test results
Figure 5.23: Creating an SR partnership
Figure 5.24: Viewing the SR partnership
Figure 5.25: Examining volumes on SRV2
Figure 5.26: Examining updated RG status
Figure 5.27: Examining volumes on SRV2
Figure 5.28: Installing FSRM
Figure 5.29: Viewing the test email
Figure 5.30: Viewing quota templates
Figure 5.31: Building a quota
Figure 5.32: Examining the FSRM email
Figure 5.33: Testing the quota limit
Figure 5.34: Viewing the total size of the folder contents
Figure 5.35: Examining the existing file groups
Figure 5.36: Examining existing templates
Figure 5.37: Creating a file screen
Figure 5.38: Testing a file screen
Figure 5.39: Viewing notification limits
Figure 5.40: Testing the active file screen

Figure 5.41: Viewing the screening email

Chapter 6

Figure 6.1: Systems used in this chapter

Figure 6.2: Installing file server features

Figure 6.3: Viewing SMB server settings

Figure 6.4: Viewing the reconfigured SMB server settings

Figure 6.5: Viewing shares on FS1

Figure 6.6: Creating an SMB share

Figure 6.7: Viewing share access

Figure 6.8: Reviewing NTFS permissions

Figure 6.9: Viewing folder ACL

Figure 6.10: Installing the iSCSI target feature

Figure 6.11: Viewing iSCSI target server settings

Figure 6.12: Creating an iSCSI virtual disk

Figure 6.13: Creating an iSCSI target

Figure 6.14: Creating the iSCSI target portal

Figure 6.15: Viewing the SalesTarget

Figure 6.16: Connecting to the `SalesTarget`

Figure 6.17: Viewing the disk

Figure 6.18: Creating an `S:` drive

Figure 6.19: Using the iSCSI `S:` drive

Figure 6.20: Setting up the iSCSI portal

Figure 6.21: Installing clustering on `FS1` and `FS2`

Figure 6.22: Viewing cluster test results

Figure 6.23: Adding an iSCSI disk to the cluster

Figure 6.24: Adding the new disk to the CSV
Figure 6.25: Adding a continuously available share
Figure 6.26: Viewing shares
Chapter 7
Figure 7.1: Systems used in this chapter
Figure 7.2: Installing the Print Server feature
Figure 7.3: Reviewing the setup of the new printer
Figure 7.4: Checking on the `SalesPrinter1` printer
Figure 7.5: Reviewing the setup of the new printer
Figure 7.6: Reviewing existing spool folder
Figure 7.7: Reviewing the spool folder
Figure 7.8: Reviewing the spool folder after the change
Figure 7.9: Displaying the number of printers
Figure 7.10: Displaying the printer's details
Figure 7.11: Printing a test page
Figure 7.12: Viewing printer pool details
Chapter 8
Figure 8.1: Systems used in this chapter
Figure 8.2: Installing Hyper-V on `HV1`
Figure 8.3: Reviewing VM host settings
Figure 8.4: Mounting the ISO disk image
Figure 8.5: Viewing the contents of `install.wim`
Figure 8.6: Creating a new VM
Figure 8.7: Viewing newly created VM details
Figure 8.8: Displaying VM details

Figure 8.9: Getting the VM host name
Figure 8.10: Displaying VM details using VM ID
Figure 8.11: Viewing NIC details
Figure 8.12: Viewing the NIC configuration
Figure 8.13: Creating a virtual switch
Figure 8.14: Viewing the virtual NIC configuration
Figure 8.15: Viewing the NIC IP address
Figure 8.16: Joining the domain
Figure 8.17: Obtaining the host name
Figure 8.18: Shutting down a VM
Figure 8.19: Updating VM BIOS
Figure 8.20: Changing the VM processor count
Figure 8.21: Changing the VM memory allocation
Figure 8.22: Adding an SCSI controller
Figure 8.23: Creating a new virtual disk
Figure 8.24: Viewing SCSI disks
Figure 8.25: Configuring the VM processor
Figure 8.26: Restarting HVDirect
Figure 8.27: Installing Hyper-V in HVDirect
Figure 8.28: Restarting HVDirect
Figure 8.29: Examining Hyper-V within HVDirect
Figure 8.30: Examining the `HVDirect C:` drive
Figure 8.31: Examining the checkpoint files
Figure 8.32: Creating content on `HVDirect`
Figure 8.33: Viewing Hyper-V checkpoints

Figure 8.34: Viewing checkpoint files

Figure 8.35: Viewing checkpoint files after the second checkpoint

Figure 8.36: Viewing files in the VM

Figure 8.37: Viewing files in the VM after rolling forward

Figure 8.38: Viewing checkpoints for HVDirect

Figure 8.39: Viewing virtual hard disks

Figure 8.40: Viewing replication status

Figure 8.41: Viewing VM status

Figure 8.42: Examining the replication status

Figure 8.43: Viewing the progress of replication

Figure 8.44: Viewing the VMs running on HV2

Figure 8.45: Getting the VM details on the replica source

Figure 8.46: Testing the replica VM

Figure 8.47: Viewing VM status

Figure 8.48: Checking VM status after failover

Figure 8.49: Verifying the VM networking configuration

Figure 8.50: Viewing `HVDirect` VM

Figure 8.51: Viewing the VM configuration path

Figure 8.52: Getting the VM hard drive locations for `HVDirect`

Figure 8.53: Viewing the VM configuration path

Figure 8.54: Displaying migration time

Figure 8.55: Viewing the VMs on HV2

Figure 8.56: Examining VM details

Figure 8.57: Examining VM details

Figure 8.58: Measuring VM resource usage

Figure 8.59: Remeasuring VM resource usage after computation

Chapter 9

Figure 9.1: WMI architecture

Figure 9.2: Viewing classes in root namespace

Figure 9.3: Viewing namespaces in root

Figure 9.4: Getting classes in the `root\CIMV2` namespace

Figure 9.5: Getting all namespaces

Figure 9.6: Viewing some namespaces

Figure 9.7: Counting classes

Figure 9.8: Viewing namespaces on SRV2

Figure 9.9: Counting namespaces and classes on SRV2

Figure 9.10: Counting namespaces and classes on DC2

Figure 9.11: Viewing the `Win32_Share` WMI class

Figure 9.12: Counting namespaces and classes on `DC1`

Figure 9.13: Viewing the methods supported by the `Win32_Share` class

Figure 9.14: Viewing class details with a specified namespace

Figure 9.15: Viewing details of the `Win32_Share` class

Figure 9.16: Viewing details of a class in a non-default namespace

Figure 9.17: Using the `-Filter` parameter

Figure 9.18: Using a WQL query

Figure 9.19: Getting remote WMI objects

Figure 9.20: Viewing methods of the `Win32_Share` class

Figure 9.21: Viewing methods of the `Win32_Share` class

Figure 9.22: Invoking a WMI method

Figure 9.23: Viewing the SMB share

Figure 9.24: Viewing an SMB share

Figure 9.25: Removing an SMB share

Figure 9.26: Viewing WMI event details

Figure 9.27: Creating an extrinsic event subscription

Figure 9.28: Updating a registry value

Figure 9.29: Viewing event details

Figure 9.30: Creating a temporary WMI event subscription

Figure 9.31: Adding a user to an AD group

Figure 9.32: Viewing event details

Figure 9.33: Viewing event details

Figure 9.34: Viewing event details

Chapter 10

Figure 10.1: Systems used in this chapter

Figure 10.2: AD user and computer report

Figure 10.3: Creating a new FSRM report

Figure 10.4: Viewing the FSRM report

Figure 10.5: Viewing the FSRM report output files

Figure 10.6: Viewing the FSRM report output files

Figure 10.7: Using FSRM XML output

Figure 10.8: Viewing a scheduled task

Figure 10.9: Running the report interactively

Figure 10.10: Viewing the report

Figure 10.11: Viewing data collector sets with `perfmon.exe`

Figure 10.12: Counting available performance counters

Figure 10.13: Viewing CPU Information

Figure 10.14: Viewing the CPU usage chart

Figure 10.15: Viewing the System Diagnostics Report

Figure 10.16: Viewing PDF output files

Figure 10.17: Viewing printer usage

Figure 10.18: Viewing the VM report

Figure 10.19: Viewing the VM report on another Hyper-V host

Figure 10.20: Viewing a count of event logs

Figure 10.21: Viewing the total number of event logs

Figure 10.22: Viewing the total numbers of System, Application, and Security ...

Figure 10.23: Viewing the Security log events

Figure 10.24: Getting logon events
Figure 10.25: Getting logon event types
Figure 10.26: Getting RDP logons

PowerShell 7 for IT Pros

A Guide to Using PowerShell 7 to Manage Windows® Systems

Thomas Lee

WILEY

Foreword

Ten years after the release of Windows PowerShell 1.0, the PowerShell team announced PowerShell Core 6. The work toward PowerShell Core 6 started a few years earlier, and that was when I became the engineering manager for PowerShell. It was not easy early on, particularly in terms of compatibility with Windows PowerShell, but with the PowerShell 7 release we are officially starting a new chapter for PowerShell in which PowerShell 7 can be used as a replacement for (or side by side with) Windows PowerShell 5.1.

PowerShell 7 represents the future of PowerShell based on three big changes:

- **A single automation language for Windows, Linux, and macOS:** Windows PowerShell has been heavily adopted by the Windows community. With IT moving toward the cloud, this presented an opportunity for PowerShell to be the glue language for the cloud. Improvements to the web cmdlets make it simple to call REST APIs. Early partnerships with Azure, Amazon Web Services, Google Compute Cloud, and VMware ensured PowerShell cmdlet coverage for any cloud you would want to use on any platform. Along the way, we also made PowerShell a great shell to use whether you are using Windows, Linux, or macOS.
- **Moving to open source:** This was a huge change in how we write software and also how we engage with the community. We could now be much more transparent with our plans and also accept contributions from the community to address issues or add features that would not have necessarily been a

priority for the team. With about 50 percent of the pull requests coming from the community, PowerShell's future really is a community-driven project!

- **Early adopter of .NET Core:** It was not without challenges that we moved from .NET Framework to .NET Core (and now just .NET). Compatibility with Windows PowerShell modules was the biggest issue initially. .NET has come a long way in addressing the compatibility gap, adding back many APIs to enable PowerShell to be compatible with existing modules. In addition, .NET Core has substantial performance improvements that make existing PowerShell scripts and modules simply work faster without any modifications!

The mission statement of PowerShell is to make it easier for users to use compute resources. With PowerShell 7, this includes different platforms such as Windows, Linux, and macOS, but also new architectures such as ARM32 and ARM64. With PowerShell modules available for the major public and private clouds, you can leverage PowerShell to be more productive in cross-cloud or hybrid scenarios. We still maintain PowerShell's “sacred vow”— that learning a new language is hard, but with time invested learning PowerShell, we will continue to enable PowerShell users to expand their impact and productivity, such as serverless functions as a service and Jupyter Notebooks. I'm excited about the next phase of PowerShell that we started with PowerShell 7, but for me, this is really just a beginning, with many more great things to come! This book from Thomas is a great way to get started on PowerShell 7, leveraging existing experience from Windows PowerShell.

Thomas Lee has been part of the PowerShell community far longer than I have been the engineering manager for PowerShell. Some of the things I've learned about

PowerShell have come from reading his articles and blog posts. As the PowerShell team was making progress toward our substantial PowerShell 7 release, Thomas was there the whole way, promoting, teaching, and informing the community of all the great things to come with PowerShell 7. The most important aspect of what makes PowerShell successful has been the community, and Thomas has been a significant part of that.

Steve Lee

Principal Software Engineering Manager

PowerShell Team

Introduction

Hello, and thank you for buying this book. I sat in the audience at the Professional Developers Conference in Los Angeles in 2003, where Jeffrey Snover introduced Monad, which was later to become Windows PowerShell. I was excited about what I saw and heard; it was a seminal moment in my career.

Today we have a new version of PowerShell, PowerShell 7, to get excited about all over again. The PowerShell development team, combined with a fantastic community, has taken PowerShell to a new level. I continue to be excited, and I hope you are.

Before you dive into the body of this book, I hope you might take a few moments to read this short introduction where I explain my motivation for writing the book, its structure, and how you can use the PowerShell scripts in this book using Hyper-V VMs.

This book contains 10 chapters. The first chapter looks at setting up PowerShell 7 in your environment. Chapter 2 examines the issue of Windows PowerShell compatibility and shows how PowerShell 7 addresses this challenge. The remaining eight chapters cover various Windows Server features and how you manage them with PowerShell 7. Here's a short overview of what is in this book:

> **Chapter 1: Setting Up a PowerShell 7 Environment:** In this chapter, you look at how to install PowerShell 7 and VS Code. VS Code is your replacement for the older Windows PowerShell ISE. The screenshots in this book show PowerShell code running in VS Code. In production, you could consider not using VS Code, or any GUI tool for that matter, on

your server and instead rely on the PowerShell 7 console and remote text editing.

Chapter 2: PowerShell 7 Compatibility with Windows PowerShell: Compatibility with Windows PowerShell is both an important objective and a significant engineering task. This chapter describes the compatibility issue as well as providing some additional background on modules. The chapter then looks at how backward compatibility works and discusses the small number of Windows PowerShell that you cannot use in PowerShell 7.

Chapter 3: Managing Active Directory: AD is at the heart of almost every organization's network. This chapter shows how you can deploy and manage AD, including creating forests and domains as well as linking forests with cross-forest trusts. The chapter also looks at how you manage AD users, computers, groups, and more.

Chapter 4: Managing Networking: In this chapter, you look at managing your network with PowerShell 7. You examine NIC configuration, as well as installing and managing both DNS and DHCP.

Chapter 5: Managing Storage: Storage is a crucial aspect of any computer system. You need somewhere to store your files and other data. This chapter looks at managing disks and volumes/partitions as well as using a third-party module to manage NTFS permissions. The chapter also examines Storage Replica to replicate storage, possibly for disaster recovery. Finally, the chapter looks at using File Server Resource Manager to manage file quotas and file screening.

Chapter 6: Managing Shared Data: Once you have disks configured as volumes and partitions and you

have set up permissions appropriately, you need to share that data across the network. This chapter looks at how you set up and configure an SMB file server and how to create and secure SMB file shares. The chapter also looks at setting up an iSCSI target and then using that target to deploy a highly resilient clustered scale-out file server.

Chapter 7: Managing Printing: Printing has been a core feature of Windows since the beginning of Windows itself. This chapter shows how to set up and manage a print server. The chapter shows how to add a printer, how to add print drivers, how to print a test page, and how to set up a printer pool.

Chapter 8: Managing Hyper-V: Hyper-V is Microsoft's core virtualization product. This chapter shows you how to set up and manage Hyper V and how to create and manage Hyper-V VMs. The chapter also looks at VM and VM storage movement and replication, vital topics for today's VM-focused world.

Chapter 9: Using WMI with CIM Cmdlets: Windows Management Instrumentation has been a feature within Windows since NT 4. WMI provides you with access to information about your system and allows you to manage aspects of the system. WMI is useful to provide you with access to Windows functionality you cannot get via PowerShell cmdlets. This chapter explores the WMI components and shows you how to discover more. The chapter also looks at managing WMI events and shows how you can set up a permanent event handler to manage critical security events.

Chapter 10: Reporting: Knowing the status of your IT infrastructure is vital to being able to manage your computing estate. This chapter demonstrates how you can use PowerShell 7 to learn more about your

infrastructure. The chapter looks at reporting on AD users and computers, the filesystem via FSRM, printer usage, and Hyper-V host and VM usage. This chapter also looks at using performance logging and alerting to capture detailed performance information and create rich performance reports and graphs that show the performance of your infrastructure.

I wrote this book to show you, the IT pro, that moving to PowerShell 7 is easy and worth your while. Just like when moving your home, things are a bit different in PowerShell 7. But once you get settled in, you are unlikely to look back. Along with VS Code, PowerShell 7 is just better. And I hope that each chapter of this book demonstrates that.

This book assumes you are an IT professional wanting to learn how to make the most of PowerShell 7. You might be an active administrator, a consultant, or a manager. You should have a background in both Windows Server features and broadly what they do, along with an understanding of Windows PowerShell itself.

The book looks at a variety of core Windows features including Active Directory, File Services Resource Manager, WMI, printing, and more. Each chapter describes a feature area and the components with which you interact. Then the chapter shows you how you can use PowerShell 7 to deploy, manage, and leverage that feature.

In this book (and indeed any book on PowerShell), it's not possible to cover every aspect of every feature set of Windows. As Jeffrey Snover says, “To ship is to choose,” and I hope I have chosen wisely. I have also provided pointers to where you can find more information. You are welcome to email me and give me feedback (`DoctorDNS@Gmail.Com`).

This book contains a variety of scripts that you can use to manage some aspects of Windows using PowerShell 7. You can download these scripts either from the Wiley site or from my GitHub repository at github.com/doctordns/Wiley20. In the unlikely event you discover an issue with any of the scripts or find issues with the documentation, please file an issue report on the GitHub repository (github.com/doctordns/Wiley20/issues).

A key goal in developing this book is to demonstrate how easily you can use PowerShell 7 to manage a Windows Server infrastructure. There is a difference in how you install it, and you have to get used to VS Code as a replacement to the ISE. Along the way, I discovered a few issues around compatibility with Windows PowerShell, and I discuss these in Chapter 2. It is time to move forward to PowerShell 7.

I built the scripts and the book content based on a set of Windows Server 2019 Datacenter edition Hyper-V VMs. To get the most value from this book and the scripts it contains, you should build the VMs yourself and use them to test the scripts. Of course, you can use physical hosts as an alternative to virtual machines, but VMs are simpler to use. For readers who may not have the necessary hardware at hand, I include screenshots showing the output of each step of each script. To assist in creating the VMs, I have created a set of scripts. You can find these on GitHub; see Chapter 1 for more information on these scripts and how to obtain them.

One impressive aspect of PowerShell, from the beginning, is the rich and vibrant PowerShell community. There are hundreds of people around the world who love PowerShell and have delivered all kinds of goodness: tweets, forum posts, blog articles, scripts, modules, web sites, and more.

A fair number of features in PowerShell 7 come from the community.

Should you have any problem with any aspect of any component of this book—or any aspect of Windows—there is no shortage of help and guidance you can find on the Internet.

Pretty much any social media site where techies can congregate is going to have PowerShell content, help, and assistance. Feel free to visit the PowerShell forum on Spiceworks where I am a moderator (community.spiceworks.com/programming/powershell).

With that said, enjoy the book and enjoy PowerShell 7.

Fare thee well now,
Let your life proceed by its own design.
Nothing to tell now,
Let the words be yours; I'm done with mine.
"Cassidy," John Barlow/Robert Weir

Thomas Lee
June 2020
Cookham, England

Chapter 1
Setting Up a PowerShell 7 Environment

The first versions of Windows PowerShell were provided via a user-installed download, initially for Windows XP and Windows Server 2008. Today, both Windows Server and Windows 10 come with Windows PowerShell version 5.1—which in this book I'll call simply Windows PowerShell to distinguish it from PowerShell 7 (and the Windows PowerShell Integrated Scripting Environment) installed and available by default. Windows PowerShell comes with a range of commands available for basic administration of Windows.

PowerShell 7 itself does not ship as part of Windows at the time of writing. At some point, the PowerShell team may ship PowerShell 7 as a Windows component, but until that time, you need to download and install it yourself.

The Windows PowerShell Integrated Scripting Environment (ISE) does not support PowerShell 7. IT pros who want a good interactive development environment for PowerShell can use Visual Studio Code (VS Code), a free tool you can also easily download and install. VS Code comes with an array of extensions that provide a much-improved development experience for IT pros (and others).

With earlier versions of PowerShell, the vast majority of commands came bundled into Windows or were added as part of installing an application (such as Exchange Server) or adding a Windows feature to your system. With PowerShell 7, the PowerShell Gallery has become a core source of modules/commands that you can use to perform various administrative tasks. To ensure that you can take