Preface

PowerShell was first introduced to the world at the Professional Developer's conference in Los Angeles in 2003 by Jeffrey Snover. Code-named Monad, it represented a complete revolution in management. A white paper written around that time, The Monad Manifesto (refer to <http://www.jsnover.com/blog/2011/10/01/monad-manifesto/>) remains an amazing analysis of the problem at the time – that of managing large numbers of Windows systems. A key takeaway is that the GUI does not scale, whereas PowerShell does.

PowerShell has transformed the management of complex, network-based Windows infrastructure, and, increasingly, non-Windows infrastructure. Knowledge of PowerShell and how to get the most from PowerShell is now obligatory for any professional IT Pro. The popular adage continues to be true: Learn PowerShell or learn golf.

Windows PowerShell was developed on Windows for Windows administrators. PowerShell 7, the open-source successor, is also available for Mac and most of the more popular Linux distributions. This book, however, concentrates on PowerShell within a Window environment.

This book takes you through the use of PowerShell in a variety of scenarios using many of the rich set of features included in Windows Server 2022. This preface provides you with an introduction to what is in the book, along with some tips on how to get the most out of it

# Who this book is for

This book is aimed at IT professionals, including system administrators, system engineers, architects, and consultants who need to understand PowerShell 7 to simplify and automate their daily tasks. The recipes in this book have been tested on the latest versions of Windows Server.

# What this book covers

Chapter 1, Installing and Configuring PowerShell 7, shows you how you can install and configure both PowerShell 7 and VS Code which replaces the Windows PowerShell Integrated Scripting Environment (ISE) as well as installing a new font, Cascadia Code. This chapter also examines the PowerShell 7 environment including examining the PSReadLine module.

Chapter 2, Introducing PowerShell 7, looks at what’s new in PowerShell 7.This chapter examines the new features you can use with PowerShell 7 including a number of new operators, improvements in parallel processing. The chapter also looks at how PowerShell 7 formats and manages error messages.

Chapter 3, Compatibility with Windows PowerShell, explores PowerShell 7’s compatibility with Windows PowerShell. PowerShell 7 is based on the open-source .NET, which is largely but not fully compatible with the older .NET Framework, which means some features of Windows PowerShell do not work natively withing PowerShell 7. Many of the Windows PowerShell modules which come with Windows Server do not natively work within PowerShell. The various Windows Server feature teams could have updated their modules to work natively with .NET Core and PowerShell 7, and a few did so. The chapter examines the compatibility mechanism adopted by the PowerShell developers to enable older Windows PowerShell modules to function within PowerShell 7 and close the gap between what you can do with Windows PowerShell and what you can do with PowerShell 7.

Chapter 4, Managing PowerShell in the Enterprise, looks at how you can use various PowerShell 7 features that might be more common within larger enterprises. These include the Remote Server Administration Tools (RSAT), package management and the PowerShell gallery and creating a local module repository. The chapter also looks at PowerShell script signing, using short cuts and working with archive (.ZIP) files.

Chapter 5, Exploring .NET, examines .NET, which provides the foundation for PowerShell 7. The chapter looks at ,NET assemblies, classes and methods. The chapter concludes with showing you can create simple C#-based PowerShell extensions and full cmdlets.

Chapter 6, Managing Active Directory, examines how to install, manage, and leverage Active Directory, including installing domains and child domains, managing AD objects and leveraging Group Policy. The chapter also examines how you can use PowerShell to report on your AD environment.

Chapter 7, Managing Networking, shows you how to manage Windows networking with PowerShell. Networks are today central to almost every organization and this chapter looks at a variety of network-related tasks, including looking at new ways (with PowerShell) to do old things, setting up DNS, DHCP, and DHCP failover and load balancing.

Chapter 8, Implementing Enterprise Security, looks at security aspects within the context of an enterprise environment. The chapter looks at Just Enough Administration (JEA – which limits the actions an administrator can perform remotely). The chapter also looks at the event log, and PowerShell 7’s script block logging, setting PowerShell 7 related Group policies and configuring a fine-grained AD password policy. The chapter concludes with looking at the Windows Defender AV product built into Windows Server.

Chapter 9, Managing Storage, looks at managing storage in Windows Server including locally attached devices and Windows Storage Spaces. The chapter also looks at managing storage replica, a feature of Windows Server 2022.

Chapter 10, Managing Shared Data, examines different ways to share data and manage your shared data with Windows Server and PowerShell including managing NTFS permissions, creating and securing SMB shares and setting up and using iSCSI. The chapter concludes with looking at File Server Resource Manager (FSRM), a feature of Windows Server, and managing FSRM quotas, file screening and reporting.

Chapter 11, Managing Printing , shows you how to manage printers, printer queues, and printer drivers as well as how to setup a printer pool. You also examine how to print a test page.

Chapter 12, Managing Hyper-V, demonstrates the use of Hyper-V. This chapter shows you how to build and deploy VMs with Hyper-V. This includes nested Hyper-V running a Hyper-V VM inside another Hyper-V VM which is useful for a number of scenarios

Chapter 13, Managing Azure, looks at managing IAAS and PAAS resources in Azure using PowerShell. To test the recipes in this chapter, you need access to Azure. This chapter describes Azure storage and how to setup a Virtual Machine, an Azure website, and an SMB3 file share.

Chapter 14, Debugging and Troubleshooting Windows Server, looks at a number of aspects of both reactive and proactive troubleshooting. This includes using the PowerShell script debugger, getting events from the event log and using the Best Practice Analyzer contained in Windows Server.

Chapter 15, Managing Window Server with WMI, examines WMI and enables you to investigate WMI namespaces, classes and class occurrences. You retrieve information from WMI classes, update WMI using WMI methods, and manage WMI events including WMI permanent eventing.

# To get the most out of this book

I designed and wrote this book based on some assumptions and with some constraints. Please read this section to understand how I intended the book to be used and what I have assumed about you. This should help you to get the most from this book.

The first assumption I made in writing this book is that you know the very basics of Windows PowerShell. For that reason, this book is not a PowerShell tutorial. The recipes in this book make use of a wide range of PowerShell features, including WMI, Remoting, AD and so on, but you need to know the basics of PowerShell. The book was developed using Windows 10 and both Windows Server 2019 and the emerging Windows Server 2022.

The second, related, assumption is that you have a reasonable background in Windows infrastructure including AD, networking and storage. The recipes in each chapter provide an overview of the various technologies. I’ve tried to provide good links for more information on the topics in this book. The recipes are designed to show you the basics of how to manage aspects of Windows Server and how you might adapt them for your environment.

You start your exploration by installing and configuring PowerShell 7 and VS Code and creating Hyper-V VMs to test out each chapter’s recipes. I built and tested the recipes in this book step-by-step (i.e. not running the entire recipe as a single script file). If you run a recipe as a single step, some of the output may not be what you see here, due to how PowerShell formats objects.

Once you have any recipe working, try to re-factor the recipe's code into your own reusable functions. In some cases, we build simple functions as a guide to richer scripts you could build. Once you have working and useful functions, incorporate them in to organizational or personal modules and reuse the code.

As any author knows, writing PowerShell scripts for publication in a book is a layout and production nightmare. To reduce the issues specifically with line width and line wrapping, I have made extensive use of methods that ensure the command line width fits in the chapters in this book without wrapping. Many recipes use hash tables and property spatting and other devices to ensure that every line of every recipe is both 73 characters or less and that there are no unintended line breaks. I hope there are not too many issues with layout!

Many of the cmdlets, commands, and object methods used in this book produce output that may not be all that helpful or useful, particularly in production. Some cmdlets generate output which would fill many pages of this book but with little added value. For this reason, many recipes pipe cmdlet output to Out-Null. Feel free to remove this where you want to see more details. I have also adjusted the output in many cases to avoid wasted white space. Thus, if you test a recipe, you may see the output that is laid out a bit differently, but it should contain the same information. Finally, remember that the specific output you see may be different based on your environment and the specific values you use in each step.

To write this book, I have used a large VM farm consisting of over 12 Windows Server 2022 hosts and Windows 10 clients. My main development host was a well configured Windows 10 system (96 GB RAM, 2 x 6 core Xeon processors and fast SSDs). All the hosts used in this book are a combination of some physical hardware (running almost entirely on Windows 10 and a large set of VMs) as described in the recipe.

To assist in writing this book, I created a set of scripts that built the Hyper-V VMs which I used  
to develop this book. These scripts are published at: https://github.com/doctordns/  
ReskitBuildScripts. I have also published some details of the network of VMs created by  
using these scripts, complete with host names and IP addresses, is at: https://github.  
com/doctordns/ReskitBuildScripts. The full set of VMs, at the end of this writing, took  
up around 600 GB of storage. Fortunately, storage is cheap! The GitHub repository has more details on the scripts and how to run them. If you have any issues with the scripts, please file an issue on GitHub and I can assist.

PowerShell 7 provides great feature coverage with respect to being able to manage the functions and features of Windows Server 2022 using PowerShell. As with Windows PowerShell you have considerable flexibility what commands you use in your scripts. While PowerShell cmdlets are generally your first choice, in some cases, you need to dip down into .NET, or into WMI to get to objects, properties, and methods that not existing PowerShell command provides. And if that is not enough, you can develop your own .NET classes and full PowerShell 7 cmdlets.

An important aspect of the recipes in this book is the use of third-party modules obtained from the PowerShell gallery. There is a rich and vibrant PowerShell community that has created a substantial amount of functionality for you to use. The PowerShell Gallery, a repository provided by Microsoft, enablers you to download and use these modules. The NTFSSecurity module, for example, makes is simple to manage the Access Control List (ACL) on NTFS files and folders.

All the code provided in this book has been tested. it worked when I tested it, and it did what it says (at least during the writing stage). I have taken some liberties with respect to the layout and formatting to cater for the book’s production and printing process, but you should get the same results. That said, the book production process is very complex and it is possible that errors can creep in during the production stages. So if you find step in any recipe that fails for you, file an issue my GitHub repository for this book (see below). And for generic issues, please post issues to the Spiceworks PowerShell forum.

In writing the recipes, I have used full cmdlet names with all parameter names spelled out in full. This  
makes the text a bit longer, but hopefully easier to read and understand.

In writing this book, I set out to create content around a number of features of Windows Server 2022. In order to publish the book, it was necessary to avoid going too deep into every Windows Feature. I have had to decide which features (and commands) to show and which to not cover, since every chapter could easily have become a small book. To paraphrase Jeffrey Snover, *To ship is to choose*. I hope I chose well.

Some recipes in this book rely on you having run other recipes in prior chapters. These related recipes worked well when we wrote and tested them and hopefully work for you as well. If you have problems with any of the recipes, then raise issues on my GitHub repository.

Finally, there is a fine line between PowerShell and a Windows feature. To use PowerShell to manage a Windows feature, you need to understand the Windows feature itself. The chapters provide short overviews of the Windows Server features and I have provided links to help you get more information. And as ever, Bing and Google are your best friends.

## Download the example code files and VM build scripts

I have published every recipe (and a bit more) to a public GitHub repository <https://github.com/doctordns/PACKT-PS7>. There is a README.md file at the top of the repository introducing what is in the repo. Within the scripts folder, you can find all the recipes within this book.

Should you find any issues with the recipes in this repository, please file an issue at: <https://github.com/doctordns/PACKT-PS7/issues> and I can assist.

This book makes use of a farm of Hyper-V VMs which you can use to replicate the recipes. I have created a set of VM build scripts you can download from my GitHub repository at <https://github.com/doctordns/ReskitBuildScripts>. To use these scripts, you need to obtain an ISO image of Windows Server 2022 to serve as a base image. You can get that from https://www.microsoft.com/ evalcenter/evaluate-windows-server-2022-preview/ or via your Visual Studio or other subscription. You use the ISO image to create a ‘reference disk’ – then for each VM, the scripts create a unique VM based on the reference disk. This VM is a brand new install based on unattended XML used to pre-configure each VM. In theory, you could create these VMs in Azure, but I have not tested that. You can read more about how to use the build scripts from the README.md document in the repository.

## Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: <http://www.packtpub.com/sites/default/files/downloads/Bookname_ColorImages.pdf>.

## Conventions used

There are a number of text conventions used throughout this book.

CodeInText: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. For example; "Mount the downloaded WebStorm-10\*.dmg disk image file as another disk in your system."

A block of code is set as follows:

[default]

exten => s,1,Dial(Zap/1|30)

exten => s,2,Voicemail(u100)

exten => s,102,Voicemail(b100)

exten => i,1,Voicemail(s0)

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

[default]

exten => s,1,Dial(Zap/1|30)

exten => s,2,Voicemail(u100)

exten => s,102,Voicemail(b100)

exten => i,1,Voicemail(s0)

Any command-line input or output is written as follows:

# cp /usr/src/asterisk-addons/configs/cdr\_mysql.conf.sample

/etc/asterisk/cdr\_mysql.conf

Bold: Indicates a new term, an important word, or words that you see on the screen, for example, in menus or dialog boxes, also appear in the text like this. For example: "Select System info from the Administration panel."

Warnings or important notes appear like this.

Tips and tricks appear like this.

# Get in touch

Feedback from our readers is always welcome.

General feedback: Email feedback@packtpub.com, and mention the book's title in the subject of your message. If you have questions about any aspect of this book, please email us at questions@packtpub.com.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book we would be grateful if you would report this to us. Please visit, <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packtpub.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit [http://authors.packtpub.com](http://authors.packtpub.com/).

## Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit [packtpub.com](https://www.packtpub.com/).