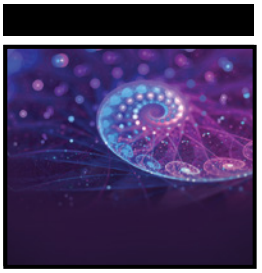
POWERSHELL 7

FOR IT PROFESSIONALS

THOMAS LEE

WILEY

PowerShell 7 for IT Pros



PowerShell 7 for IT Pros

A Guide to Using PowerShell 7 to Manage Windows® Systems

Thomas Lee

WILEY

PowerShell 7 for IT Pros

Published simultaneously in Canada

ISBN: 978-1-119-64472-9
ISBN: 978-1-119-64473-6 (ebk)
ISBN: 978-1-119-64470-5 (ebk)

Manufactured in the United States of America

Library of Congress Control Number: 2020939021

To my wife, Susan, who has been my loving companion and a constant source of strength, affection, and humour, especially during the COVID-19 crisis.

To my daughter, Rebecca. Your smiles keep me going.

To my grandmother, Louise Runk Robinson, who worked in the US Library of Congress and lit a spark in me that would ultimately lead me to write books.

To my friend, Jeffrey Snover, who has been an inspiring role model.

Thank you all for your support. You mean the world to me.

About the Author

Thomas Lee is a technical trainer, consultant, and author who specializes in the Microsoft Windows platform. Thomas began in the IT industry in 1968 as a computer operator operating an IBM 360/67 running IBM's Time-Sharing System. His career spans five decades and has included the mainframe, mini-computers, time sharing (aka very early cloud computing), AI, and other emerging technologies.

Thomas has worked in the consulting field, first with Arthur Andersen (today known as Accenture), where he witnessed the birth of the PC as a tool for clients across the board. He continues to provide consulting services to many long-standing clients.

He has also written extensively, both for the technical trade in the United Kingdom and for the worldwide publishing industry. This book marks his fourth on PowerShell. He has written hundreds of articles about Microsoft and other technologies and maintains a technical blog at `tfl09.blogspot.com`.

Thomas was a Microsoft-certified trainer for more than 25 years and was awarded the Microsoft MVP award 17 times. He was a fellow in the British Computer Society and a chartered engineer. He serves as a global board member for the MVP Reconnect program.

He lives today in an old cottage in the English countryside. He enjoys his garden, a well-stocked wine cellar, an extensive collection of Grateful Dead live recordings, and the love of his family. He loves working with PowerShell too!

About the Technical Editor

Jim Topp is a Windows Server systems administrator with more than 16 years of experience. He has been scripting and automating with PowerShell for more than 12 years. He is a community member of `PowerShell.org` and has edited several books written by Don Jones. Jim lives in Bloomington, Indiana.

Acknowledgments

Every book requires a great team of talented people to ensure a great product. While an author can write text and code, the developmental and technical editors focus the writing and ensure that the scripts are accurate and work as shown. Publishing a book is an honour as well as an incredible learning experience for me. I have always believed that you truly have to know your subject to write a book, and I certainly have learned a lot in the process.

First, my deepest gratitude to Wiley, and the fantastic team at Wiley, for bringing this book to life: Kenyon Brown, Pete Gaughan, and others. Thank you for recognizing that I had a story to tell.

A huge thank-you to my developmental editor, Jim Compton, for your wisdom, kindness, and patience during through my writing "process." You have been a great inspiration to me, and I am indebted for your contributions.

Writing hundreds of lines of PowerShell 7 code and transcribing it accurately into both the book's text and the screenshots is an amazingly difficult job. Jim Topp did an outstanding job of checking every line of code and every screenshot as well as making great suggestions to improve what we have here.

In writing this book, I have been impressed on a daily basis with the entire PowerShell development team. You folks have delivered high-quality code consistently, been able to listen to and accept the views of the community, and deliver a world-class product. Joey Aiello, Steve Lee, and Jim Truher, among others, have been an inspiration.

Another significant inspiration is Jeffrey Snover. From that day in L.A. when he first introduced Monad, he has been an awesome role model. He is smart, focused, and one of the best presenters I have ever had the good fortune to watch. When I noted, on Twitter, that PowerShell Core 6.1 was kind of like Windows PowerShell 2.0 all over again, Jeffrey noted in his understated way that, while it was true, PowerShell had better velocity. An understatement at the very least.

The PowerShell 7 code in this book was developed and tested on a variety of Hyper-V virtual machines running Windows Server 2019. Because I am a member of the MVP Reconnect program, the Visual Studio team at Microsoft generously provided me with a subscription to Visual Studio Online, from which I obtained the necessary ISO images with which you (and I) can build the VMs. I could not have written this book without their generous assistance.

Thanks too to the PowerShell community. The community has created a massive number of PowerShell resources, code, and documentation. Deep dives, new modules, tips, and tricks—so much awesome content. Thanks to Jeff Hick, Don Jones, Richard Siddaway, and many more.

Lastly, a shout-out to the PowerShell community at Spiceworks—a friendly bunch of helpful PowerShell addicts (`community.spiceworks.com/programming/powershell`). If you are keen on PowerShell, consider visiting and joining in.

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

Contents at a Glance

Contents