

# Vorwort zur zweiten Auflage

»Windows PowerShell« wurde 2006 das erste Mal ausgeliefert und 2016 als Bestandteil von Windows 10 und Server 2016 fertiggestellt.

Nach der Fertigstellung der in Windows integrierten »Windows PowerShell« begann Microsoft ab 2016 mit der Entwicklung einer »neuen« PowerShell, die zuerst PowerShell Core, dann PowerShell 6 und schließlich PowerShell 7 oder nur noch »PowerShell« hieß. Hauptsächlicher Vorzug der »neuen« PowerShell: Sie läuft nicht nur unter dem Betriebssystem Windows, sondern plattformübergreifend auch unter Linux und macOS.

Warum also die Neuauflage eines Buchs nur zur »Windows PowerShell«? Weil weniger manchmal genau richtig ist.

Fast alle Unternehmen, die Windows-Betriebssysteme und Microsoft-Services einsetzen, nutzen in ihrer Unternehmens-IT weiterhin die in Windows integrierte »Windows PowerShell«. Nur sie garantiert derzeit 100% Skriptkompatibilität und wird vollständig von Microsoft »supportet«.

Und anders als die »modernere« PowerShell 7 ändert sich die in Windows integrierte »Windows PowerShell« auch nicht mehr in unvorhersehbarer Weise mit dem Einspielen von Windows-Updates. Bei der »neuen« PowerShell 7 kann es nämlich durchaus passieren, dass nach dem neuesten Update ein Befehl (»Cmdlet«) anders oder gar nicht mehr funktioniert, auch wenn das eher selten vorkommt.

Wenn Sie im Windows-Umfeld arbeiten und ein solides PowerShell-Wissen für den produktiven Alltag aufbauen möchten, wird Ihnen dieses Buch sehr helfen. Sie werden in diesem Praxisbuch nur das finden, was im Windows-Umfeld tatsächlich funktioniert. Die neue umfassende Funktionalität von PowerShell 7 wird nicht behandelt. So laufen Sie nicht Gefahr, Skripte zu entwickeln, die anschließend im Unternehmensumfeld auf »Windows PowerShell« nicht oder nur fehlerhaft laufen.

Und wenn Sie später (oder parallel zu Ihrem beruflichen Einsatz) doch die neue PowerShell 7 testen, werden Sie feststellen: Alles, was Sie in diesem Buch lernen, findet sich auch in der »neuen« PowerShell wieder, die schließlich die logische Weiterentwicklung der »Windows PowerShell« ist.

## **Vorwort zur zweiten Auflage**

Da der Funktionsumfang der »Windows PowerShell« unverändert geblieben ist, enthält diese Neuauflage keine neuen Themen oder Kapitel. Der Text wurde komplett durchgesehen, kleinere Fehler wurden berichtigt und einige didaktisch motivierte Ergänzungen vorgenommen.

Herzlichst Ihr

Dr. Tobias Weltner

# Einleitung

PowerShell ist als Automations-sprache entwickelt worden: IT-Administratoren und ambitionierte Computeranwender können damit Routineaufgaben automatisch erledigen. Das spart Zeit und lästige Handgriffe und ist aus der modernen IT nicht mehr wegzudenken.

Darüber hinaus ist PowerShell inzwischen aber auch eine sehr effektive Programmiersprache geworden, mit der man ohne komplexes Vorwissen und in nur wenigen Zeilen Code kleine Programme und Tools schreiben kann. Das ist für nahezu jeden ambitionierten Computeranwender interessant, nicht mehr lediglich für IT-Profis.

Und spätestens mit PowerShell 6 laufen PowerShell-Skripte nun sogar nicht mehr nur auf Windows, sondern auch auf Linux und macOS. Das macht PowerShell-Fachwissen noch erheblich vielseitiger.

Dieses Buch beginnt bei null und führt Sie Schritt für Schritt in alle wichtigen Grundlagen der PowerShell ein. Ob Sie PowerShell beruflich nutzen oder privat, quer über das Netzwerk auf Hunderten von Computern Programme konfigurieren oder bloß auf dem eigenen Notebook Urlaubsbilder umbenennen wollen: Dieses Buch liefert zahlreiche Praxisbeispiele und erklärt die eingesetzten Techniken so, dass Sie die Beispiele leicht an eigene Bedürfnisse anpassen können.

Sieben didaktisch aufeinander aufbauende Teile führen Sie ohne notwendiges Vorwissen in die Möglichkeiten der PowerShell ein.

Im ersten Kapitel erfahren Sie, wie PowerShell auf dem Computer eingerichtet wird und welche essenziellen Einstellungen und ersten Schritte wichtig sind, um PowerShell-Befehle ausführen zu können.

In **Teil A** wird die PowerShell als praktische interaktive Konsole betrachtet – schnelle Ergebnisse mit ganz wenigen Eingaben. Sie entdecken, welche Befehle die PowerShell versteht, wie man schnell die benötigten Befehle findet und neue Befehle nachrüsten kann. In diesem Teil wird auch das universelle Laufwerk-konzept der PowerShell vorgestellt, mit dem beinahe alles – vom Dateisystem über Zertifikate und Registrierung bis hin zu Datenbanken und Benutzerverwaltung – als Laufwerk gesehen werden kann, auf dem man Informationen findet.

**Teil B** zeigt dann, wie aus mehreren Einzelbefehlen und der PowerShell-Pipeline komplexere und mächtigere Automations-skripte entstehen, mit denen sich vielfältige Aufgaben vollständig automatisieren lassen.

## Einleitung

**Teil C** beschäftigt sich mit Operatoren, die oft unterschätzt werden. Hier erfahren Sie, wie man intelligente Entscheidungen (Bedingungen) formuliert. Einen besonderen Schwerpunkt bilden die Textoperatoren und sogenannten »regulären Ausdrücke«, mit denen PowerShell die für Sie wichtigen Informationen aus Texten, Logdateien und von Webseiten fischen kann.

PowerShell kommt bereits mit Hunderten eigener Befehle, doch bietet es vor allem eine Schnittstelle zu den unzähligen Betriebssystemfunktionalitäten, die im .NET Framework bereitgestellt werden. In **Teil D** erfahren Sie, wie PowerShell auf all diese Funktionen zugreifen kann. So erhalten Ihre PowerShell-Skripte Zugriff auf genau die gleichen Möglichkeiten, die auch Anwendungsentwicklern anderer .NET-Sprachen zur Verfügung stehen, und können von DNS-Auflösung über Sprachausgabe bis hin zu eigenen Fenstern und Oberflächen alles nachrüsten, was noch nicht über vorgefertigte PowerShell-Cmdlets erreichbar war.

Weil PowerShell eine dynamische Sprache ist, kann man den Befehlssatz sehr leicht erweitern. **Teil E** erklärt zuerst, wie Ihr PowerShell-Code zu einem PowerShell-Befehl wird, der dann in allen PowerShells zur Verfügung steht. Danach erfahren Sie, wie PowerShell-Module erzeugt werden: Mit ihnen lassen sich Befehlssammlungen an Kollegen und Kunden weitergeben. Und schließlich wird das brandneue PowerShellGet vorgestellt, über das Module automatisiert verteilt werden können.

Wenn PowerShell dabei einmal nicht genau das tut, was Sie sich vorstellen, hilft **Teil F** und zeigt, wie Fehler im Code erkannt und »behandelt« werden können. Dazu zählt auch das Mitprotokollieren von Fehlern und ungewöhnlichen Zuständen.

In **Teil G** wird zum Abschluss das »PowerShell Remoting« vorgestellt. Mit diesem extrem mächtigen Feature kann PowerShell-Code remote auf einem oder vielen anderen Computern parallel ausgeführt werden. Weil PowerShell inzwischen auch auf Linux und macOS ausführbar ist, kann man darüber unter anderem auch Informationen in heterogenen Welten austauschen.

Dieses Buch ist eine überarbeitete und gekürzte Fassung des über 1.100 Seiten starken »Windows PowerShell 5: Windows Automation für Einsteiger & Profis«, das sich auf die für Einsteiger wichtigen Inhalte fokussiert.

## Wie Sie dieses Buch nutzen

Dieses Buch setzt keinerlei Grundkenntnisse voraus, wenn Sie von vorn zu lesen beginnen – und das ist auch empfehlenswert. Die Kapitel bauen aufeinander auf. Am Anfang jedes Kapitels finden Sie eine kurze Zusammenfassung, falls es einmal eilig ist.

Die PowerShell-Beispiele im Buch sind jeweils in einer anderen Schriftart formatiert. Damit Sie leichter erkennen, welche Eingaben von Ihnen erwartet werden, wird bei allen Eingaben die PowerShell-Eingabeaufforderung `PS>` (einschließlich der Leerstelle hinter dem `>`) vorangestellt. Diese Eingabeaufforderung kann bei Ihnen auch anders aussehen und sollte in den Beispielen natürlich nicht mit eingegeben werden.

## Achtung


Bitte verwenden Sie die Begleitmaterialien immer im Kontext des entsprechenden Buchkapitels. Viele der Beispiele funktionieren nur, wenn Sie die entsprechenden Vorarbeiten im Kapitel beachtet haben, oder können auch unerwartete Resultate liefern, wenn man die Beispiele aus dem Zusammenhang des Kapitels reißt.

## Noch mehr Unterstützung

Falls bei der Arbeit mit diesem Buch Fragen auftauchen oder Sie Anregungen haben, senden Sie mir eine Nachricht an meine Mailadresse *tobias@powershell.one*.

Bevor ich Ihnen viel Spaß und Erfolg mit PowerShell wünsche, geht noch ein großes Dankeschön an meine Lektorin Ariane Hesse und die Korrektorin Sibylle Feldmann, die dieses Buch mit allergrößtem Sachverstand und mit Sorgfalt begleitet haben.

Herzlichst Ihr  
Dr. Tobias Weltner

Diese Leseprobe haben Sie beim  
 **edv-buchversand.de** heruntergeladen.  
Das Buch können Sie online in unserem  
Shop bestellen.

[Hier zum Shop](#)