

Für  
PowerShell  
2 bis 6

holger SCHWICHTENBERG

WINDOWS

# PowerShell 5.1 und PowerShell Core 6.1



3. Auflage

## DAS PRAXISBUCH



»Sehr gut« dotnetpro zur 2. Auflage



Im Internet: Codebeispiele, Forum,  
PowerShell-Kurzreferenz

HANSER

[www.IT-Visions.de](http://www.IT-Visions.de)  
Dr. Holger Schwichtenberg

## Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



[www.hanser-fachbuch.de/newsletter](http://www.hanser-fachbuch.de/newsletter)



**Hanser Update** ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



[www.hanser-fachbuch.de/update](http://www.hanser-fachbuch.de/update)





Holger Schwichtenberg

# **Windows PowerShell 5.1 und PowerShell Core 6.1**

Das Praxisbuch

3., aktualisierte Auflage

**HANSER**

Der Autor:  
*Dr. Holger Schwichtenberg*, Essen  
www.IT-Visions.de

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.



MIX  
Papier aus verantwortungsvollen Quellen  
FSC® C013736

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2019 Carl Hanser Verlag München, [www.hanser-fachbuch.de](http://www.hanser-fachbuch.de)

Lektorat: Sylvia Hasselbach

Copy editing: Sandra Gottmann, Nienberge

Umschlagdesign: Marc Müller-Bremer, [www.rebranding.de](http://www.rebranding.de), München

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

E-Book-ISBN: 978-3-446-45923-6

# Inhalt

<b>Vorwort</b> .....	<b>XXIII</b>
<b>Über den Autor Dr. Holger Schwichtenberg</b> .....	<b>XXIX</b>
<b>Teil A: PowerShell-Basiswissen</b> .....	<b>1</b>
<b>1 Erste Schritte mit der PowerShell</b> .....	<b>3</b>
1.1 Was ist die PowerShell? .....	3
1.2 Windows PowerShell versus PowerShell Core .....	4
1.3 Windows PowerShell herunterladen und auf anderen Windows- Betriebssystemen installieren .....	4
1.4 Die Windows PowerShell testen .....	8
1.5 PowerShell Core installieren und testen .....	18
1.6 Woher kommen die Commandlets? .....	24
1.7 PowerShell Community Extensions (PSCX) herunterladen und installieren ..	25
1.8 Den Windows PowerShell-Editor „ISE“ verwenden .....	26
<b>2 Fakten zur PowerShell</b> .....	<b>31</b>
2.1 Geschichte der PowerShell .....	31
2.2 Motivation zur PowerShell .....	33
2.3 Betriebssysteme mit vorinstallierter PowerShell .....	36
2.4 Einflussfaktoren auf die Entwicklung der PowerShell .....	37
2.5 Anbindung an Klassenbibliotheken .....	39
2.6 PowerShell versus WSH .....	39
<b>3 Einzelbefehle der PowerShell</b> .....	<b>43</b>
3.1 Commandlets .....	43
3.2 Aliase .....	56
3.3 Ausdrücke .....	64
3.4 Externe Befehle .....	65
3.5 Dateinamen .....	66

<b>4</b>	<b>Hilfefunktionen</b>	<b>69</b>
4.1	Auflisten der verfügbaren Befehle	69
4.2	Volltextsuche	71
4.3	Erläuterungen zu den Befehlen	72
4.4	Hilfe zu Parametern	73
4.5	Hilfe mit Show-Command	75
4.6	Hilfefenster	76
4.7	Allgemeine Hilfetexte	78
4.8	Aktualisieren der Hilfsdateien	79
4.9	Online-Hilfe	81
4.10	Fehlende Hilfetexte	82
4.11	Dokumentation der .NET-Klassen	83
<b>5</b>	<b>Objektorientiertes Pipelining</b>	<b>87</b>
5.1	Pipeline-Operator	87
5.2	.NET-Objekte in der Pipeline	88
5.3	Pipeline Processor	90
5.4	Pipelining von Parametern	91
5.5	Pipelining von klassischen Befehlen	94
5.6	Anzahl der Objekte in der Pipeline	95
5.7	Zeilenumbrüche in Pipelines	96
5.8	Zugriff auf einzelne Objekte aus einer Menge	96
5.9	Zugriff auf einzelne Werte in einem Objekt	98
5.10	Methoden ausführen	99
5.11	Analyse des Pipeline-Inhalts	101
5.12	Filtern	113
5.13	Zusammenfassung von Pipeline-Inhalten	116
5.14	„Kastrierung“ von Objekten in der Pipeline	117
5.15	Sortieren	118
5.16	Duplikate entfernen	119
5.17	Gruppierung	120
5.18	Berechnungen	122
5.19	Zwischenschritte in der Pipeline mit Variablen	122
5.20	Verzweigungen in der Pipeline	123
5.21	Vergleiche zwischen Objekten	125
5.22	Zusammenfassung	126
5.23	Praxisbeispiele	127
<b>6</b>	<b>PowerShell-Skripte</b>	<b>129</b>
6.1	Skriptdateien	129

6.2	Start eines Skripts .....	131
6.3	Aliase für Skripte verwenden .....	132
6.4	Parameter für Skripte .....	133
6.5	Skripte dauerhaft einbinden (Dot Sourcing) .....	134
6.6	Das aktuelle Skriptverzeichnis .....	135
6.7	Sicherheitsfunktionen für PowerShell-Skripte .....	135
6.8	Anforderungsdefinitionen von Skripten .....	138
6.9	Skripte anhalten .....	138
6.10	Versionierung und Versionsverwaltung von Skripten .....	139
<b>7</b>	<b>PowerShell-Skriptsprache .....</b>	<b>141</b>
7.1	Hilfe zur PowerShell-Skriptsprache .....	141
7.2	Befehlstrennung .....	142
7.3	Kommentare .....	142
7.4	Variablen .....	143
7.5	Variablenbedingungen .....	153
7.6	Zahlen .....	154
7.7	Zeichenketten (Strings) .....	156
7.8	Reguläre Ausdrücke .....	165
7.9	Datum und Uhrzeit .....	171
7.10	Arrays .....	173
7.11	ArrayList .....	176
7.12	Assoziative Arrays (Hash-Tabellen) .....	177
7.13	Operatoren .....	178
7.14	Überblick über die Kontrollkonstrukte .....	182
7.15	Schleifen .....	183
7.16	Bedingungen .....	188
7.17	Unterroutinen (Prozedur/Funktionen) .....	190
7.18	Eingebaute Funktionen .....	196
7.19	Fehlerbehandlung .....	197
7.20	Objektorientiertes Programmieren mit Klassen .....	204
<b>8</b>	<b>Ausgaben .....</b>	<b>209</b>
8.1	Ausgabe-Commandlets .....	209
8.2	Benutzerdefinierte Tabellenformatierung .....	212
8.3	Benutzerdefinierte Listenausgabe .....	214
8.4	Mehrspaltige Ausgabe .....	214
8.5	Out-GridView .....	215
8.6	Standardausgabe .....	217
8.7	Einschränkung der Ausgabe .....	219

8.8	Seitenweise Ausgabe .....	219
8.9	Ausgabe einzelner Werte .....	220
8.10	Details zum Ausgabeoperator .....	222
8.11	Ausgabe von Methodenergebnissen und Unterobjekten in Pipelines .....	226
8.12	Ausgabe von Methodenergebnissen und Unterobjekten in Zeichenketten .....	226
8.13	Unterdrückung der Ausgabe .....	227
8.14	Ausgaben an Drucker .....	228
8.15	Ausgaben in Dateien .....	228
8.16	Umleitungen (Redirection) .....	229
8.17	Fortschrittsanzeige .....	229
8.18	Sprachausgabe .....	230
<b>9</b>	<b>Das PowerShell-Navigationsmodell (PowerShell Provider) .....</b>	<b>233</b>
9.1	Einführungsbeispiel: Navigation in der Registrierungsdatenbank .....	233
9.2	Provider und Laufwerke .....	234
9.3	Navigationsbefehle .....	237
9.4	Pfadangaben .....	237
9.5	Beispiel .....	239
9.6	Eigene Laufwerke definieren .....	240
<b>10</b>	<b>Fernauführung (Remoting) .....</b>	<b>241</b>
10.1	RPC-Fernabfrage ohne WS-Management .....	242
10.2	Anforderungen an PowerShell Remoting .....	243
10.3	Rechte für PowerShell-Remoting .....	244
10.4	Einrichten von PowerShell Remoting .....	245
10.5	Überblick über die Fernauführungs-Commandlets .....	247
10.6	Interaktive Fernverbindungen im Telnet-Stil .....	248
10.7	Fernauführung von Befehlen .....	249
10.8	Parameterübergabe an die Fernauführung .....	253
10.9	Fernauführung von Skripten .....	254
10.10	Ausführung auf mehreren Computern .....	255
10.11	Sitzungen .....	256
10.12	Implizites Remoting .....	261
10.13	Zugriff auf entfernte Computer außerhalb der eigenen Domäne .....	262
10.14	Verwaltung des WS-Management-Dienstes .....	265
10.15	PowerShell Direct für Hyper-V .....	267
10.16	Praxisbeispiel zu PowerShell Direct .....	269
<b>11</b>	<b>PowerShell-Werkzeuge .....</b>	<b>273</b>
11.1	PowerShell-Standardkonsole .....	273

11.2	PowerShell Integrated Scripting Environment (ISE)	282
11.3	PowerShell Script Analyzer	292
11.4	PowerShell Analyzer	298
11.5	PowerShell Tools for Visual Studio	299
11.6	PowerShell Pro Tools for Visual Studio	300
11.7	NuGet Package Manager	301
11.8	PowerShell-Erweiterung für Visual Studio Code	301
11.9	PowerShell Web Access (PSWA)	304
11.10	Azure Cloud Shell	310
11.11	ISE Steroids	310
11.12	PowerShellPlus	311
11.13	PoshConsole	314
11.14	PowerGUI	315
11.15	PrimalScript	316
11.16	PowerShell Help	318
11.17	CIM Explorer for PowerShell ISE	318
11.18	PowerShell Help Reader	319
11.19	PowerShell Remoting	320
<b>12</b>	<b>Windows PowerShell Core 5.1 in Windows Nano Server</b>	<b>321</b>
<b>13</b>	<b>PowerShell Core 6.x für Windows, Linux und macOS</b>	<b>323</b>
13.1	Geschichte der PowerShell Core	323
13.2	Motivation für den Einsatz der PowerShell Core auf Linux und macOS	324
13.3	Funktionsumfang der PowerShell Core	325
13.4	Entfallene Commandlets in PowerShell Core	327
13.5	Erweiterungsmodule nutzen in PowerShell Core	332
13.6	Geänderte Funktionen in PowerShell Core	337
13.7	Neue Funktionen der PowerShell Core	340
13.8	PowerShell Core-Konsole	342
13.9	VSCoDe-PowerShell als Editor für PowerShell Core	342
13.10	Verwendung auf Linux und macOS	347
13.11	PowerShell-Remoting via SSH	353
13.12	Dokumentation zur PowerShell Core	357
13.13	Quellcode zur PowerShell Core	359
<b>Teil B:</b>	<b>PowerShell-Aufbauwissen</b>	<b>361</b>
<b>14</b>	<b>Verwendung von .NET-Klassen</b>	<b>363</b>
14.1	.NET versus .NET Core	363
14.2	.NET-Bibliotheken	364

14.3	Microsoft Developer Network (MSDN) .....	365
14.4	Überblick über die Verwendung von .NET-Klassen .....	366
14.5	Erzeugen von Instanzen .....	367
14.6	Parameterbehaftete Konstruktoren .....	369
14.7	Initialisierung von Objekten .....	370
14.8	Nutzung von Attributen und Methoden .....	371
14.9	Statische Mitglieder in .NET-Klassen und statische .NET-Klassen .....	373
14.10	Generische Klassen nutzen .....	376
14.11	Zugriff auf bestehende Objekte .....	378
14.12	Laden von Assemblies .....	378
14.13	Verwenden von Nuget-Assemblies .....	381
14.14	Objektanalyse .....	383
14.15	Auflistungen (Enumerationen) .....	384
14.16	Verknüpfen von Aufzählungswerten .....	385
<b>15</b>	<b>Verwendung von COM-Klassen .....</b>	<b>387</b>
15.1	Erzeugen von COM-Instanzen .....	387
15.2	Nutzung von Attributen und Methoden .....	388
15.3	Liste aller COM-Klassen .....	389
15.4	Holen bestehender COM-Instanzen .....	390
15.5	Distributed COM (DCOM) .....	390
<b>16</b>	<b>Zugriff auf die Windows Management Instrumentation (WMI) ..</b>	<b>391</b>
16.1	Einführung in WMI .....	391
16.2	WMI in der PowerShell .....	418
16.3	Open Management Infrastructure (OMI) .....	420
16.4	Abruf von WMI-Objektmenen .....	420
16.5	Fernzugriffe .....	421
16.6	Filtern und Abfragen .....	421
16.7	Liste aller WMI-Klassen .....	425
16.8	Hintergrundwissen: WMI-Klassenprojektion mit dem PowerShell-WMI-Objektadapter .....	426
16.9	Beschränkung der Ausgabeliste bei WMI-Objekten .....	430
16.10	Zugriff auf einzelne Mitglieder von WMI-Klassen .....	432
16.11	Werte setzen in WMI-Objekten .....	432
16.12	Umgang mit WMI-Datumsangaben .....	434
16.13	Methodenaufrufe .....	435
16.14	Neue WMI-Instanzen erzeugen .....	436
16.15	Instanzen entfernen .....	437
16.16	Commandlet Definition XML-Datei (CDXML) .....	438

<b>17</b>	<b>Dynamische Objekte</b>	<b>441</b>
17.1	Erweitern bestehender Objekte	441
17.2	Komplett dynamische Objekte	443
<b>18</b>	<b>Einbinden von C# und Visual Basic .NET</b>	<b>445</b>
<b>19</b>	<b>Win32-API-Aufrufe</b>	<b>447</b>
<b>20</b>	<b>Benutzereingaben</b>	<b>451</b>
20.1	Read-Host	451
20.2	Benutzerauswahl	452
20.3	Grafischer Eingabedialog	453
20.4	Dialogfenster	454
20.5	Authentifizierungsdialg	454
20.6	Zwischenablage (Clipboard)	456
<b>21</b>	<b>Fehlersuche</b>	<b>459</b>
21.1	Detailinformationen	459
21.2	Einzelschrittmodus	460
21.3	Zeitmessung	461
21.4	Ablaufverfolgung (Tracing)	462
21.5	Erweiterte Protokollierung aktivieren	463
21.6	Script-Debugging in der ISE	465
21.7	Kommandozeilenbasiertes Script-Debugging	465
<b>22</b>	<b>Transaktionen</b>	<b>467</b>
22.1	Commandlets für Transaktionen	467
22.2	Start und Ende einer Transaktion	468
22.3	Zurücksetzen der Transaktion	469
22.4	Mehrere Transaktionen	470
<b>23</b>	<b>Standardeinstellungen ändern mit Profilskripten</b>	<b>471</b>
23.1	Profilpfade	471
23.2	Ausführungsreihenfolge	473
23.3	Beispiel für eine Profildatei	473
23.4	Starten der PowerShell ohne Profilskripte	474
<b>24</b>	<b>Digitale Signaturen für PowerShell-Skripte</b>	<b>475</b>
24.1	Zertifikat erstellen	475
24.2	Skripte signieren	477
24.3	Verwenden signierter Skripte	478
24.4	Mögliche Fehlerquellen	479

<b>25</b>	<b>Hintergrundaufträge („Jobs“)</b> .....	<b>481</b>
25.1	Voraussetzungen .....	481
25.2	Architektur .....	482
25.3	Starten eines Hintergrundauftrags .....	482
25.4	Hintergrundaufträge abfragen .....	483
25.5	Warten auf einen Hintergrundauftrag .....	484
25.6	Abbrechen und Löschen von Aufträgen .....	484
25.7	Analyse von Fehlermeldungen .....	485
25.8	Fernausführung von Hintergrundaufträgen .....	485
25.9	Praxisbeispiel .....	486
<b>26</b>	<b>Geplante Aufgaben und zeitgesteuerte Jobs</b> .....	<b>489</b>
26.1	Geplante Aufgaben (Scheduled Tasks) .....	489
26.2	Zeitgesteuerte Jobs .....	493
<b>27</b>	<b>PowerShell-Workflows</b> .....	<b>499</b>
27.1	Ein erstes Beispiel .....	499
27.2	Unterschiede zu einer Function bzw. einem Skript .....	504
27.3	Einschränkungen bei Workflows .....	504
27.4	Workflows in der Praxis .....	506
27.5	Workflows in Visual Studio erstellen .....	513
<b>28</b>	<b>Ereignissystem</b> .....	<b>531</b>
28.1	WMI-Ereignisse .....	531
28.2	WMI-Ereignisabfragen .....	531
28.3	WMI-Ereignisse seit PowerShell 1.0 .....	533
28.4	Registrieren von WMI-Ereignisquellen seit PowerShell 2.0 .....	534
28.5	Auslesen der Ereignisliste .....	535
28.6	Reagieren auf Ereignisse .....	537
28.7	WMI-Ereignisse ab PowerShell-Version 3.0 .....	539
28.8	Registrieren von .NET-Ereignissen .....	539
28.9	Erzeugen von Ereignissen .....	540
<b>29</b>	<b>Datenbereiche und Datendateien</b> .....	<b>543</b>
29.1	Datenbereiche .....	543
29.2	Datendateien .....	545
29.3	Mehrsprachigkeit/Lokalisierung .....	546
<b>30</b>	<b>Desired State Configuration (DSC)</b> .....	<b>549</b>
30.1	Grundprinzipien .....	550
30.2	DSC für Linux .....	550

30.3	Ressourcen	551
30.4	Verfügbare DSC-Ressourcen	551
30.5	Eigenschaften einer Ressource	554
30.6	Aufbau eines DSC-Dokuments	554
30.7	Commandlets für die Arbeit mit DSC	555
30.8	Ein erstes DSC-Beispiel	555
30.9	Kompilieren und Anwendung eines DSC-Dokuments	556
30.10	Variablen in DSC-Dateien	558
30.11	Parameter für DSC-Dateien	559
30.12	Konfigurationsdaten	560
30.13	Entfernen einer DSC-Konfiguration	563
30.14	DSC Pull Server	566
30.15	DSC-Praxisbeispiel 1: IIS installieren	574
30.16	DSC-Praxisbeispiel 2: Software installieren	575
30.17	DSC-Praxisbeispiel 3: Software deinstallieren	577
30.18	Realisierung einer DSC-Ressource	578
30.19	Weitere Möglichkeiten	579
<b>31</b>	<b>PowerShell-Snap-Ins</b>	<b>581</b>
31.1	Einbinden von Snap-Ins	581
31.2	Liste der Commandlets	585
<b>32</b>	<b>PowerShell-Module</b>	<b>587</b>
32.1	Überblick über die Commandlets	587
32.2	Modulararchitektur	588
32.3	Aufbau eines Moduls	589
32.4	Module aus dem Netz herunterladen und installieren mit PowerShellGet	589
32.5	Module manuell installieren	596
32.6	Doppeldeutige Namen	596
32.7	Auflisten der verfügbaren Module	598
32.8	Importieren von Modulen	599
32.9	Entfernen von Modulen	602
<b>33</b>	<b>Ausgewählte PowerShell-Erweiterungen</b>	<b>603</b>
33.1	PowerShell-Module in Windows 7 und Windows Server 2008 R2	604
33.2	PowerShell-Module in Windows 8.0 und Windows Server 2012	605
33.3	PowerShell-Module in Windows 8.1 und Windows Server 2012 R2	607
33.4	PowerShell-Module in Windows 10 und Windows Server 2016	610
33.5	PowerShell Community Extensions (PSCX)	614
33.6	PowerShellPack	618
33.7	www.IT-Visions.de: PowerShell Extensions	619

33.8	Quest Management Shell for Active Directory .....	620
33.9	Microsoft Exchange Server .....	621
33.10	System Center Virtual Machine Manager .....	622
33.11	PowerShell Management Library for Hyper-V (pshyperv) .....	623
33.12	Powershell Outlook Account Manager .....	624
33.13	PowerShell Configurator (PSConfig) .....	625
33.14	Weitere Erweiterungen .....	626
<b>34</b>	<b>Delegierte Administration/Just Enough Administration (JEA) ..</b>	<b>627</b>
34.1	JEA-Konzept .....	627
34.2	PowerShell-Sitzungskonfiguration erstellen .....	627
34.3	Sitzungskonfiguration nutzen .....	631
34.4	Delegierte Administration per Webseite .....	632
<b>35</b>	<b>Tipps und Tricks zur PowerShell .....</b>	<b>633</b>
35.1	Alle Anzeigen löschen .....	633
35.2	Befehlsgeschichte .....	633
35.3	System- und Hostinformationen .....	634
35.4	Anpassen der Eingabeaufforderung (Prompt) .....	635
35.5	PowerShell-Befehle aus anderen Anwendungen heraus starten .....	636
35.6	ISE erweitern .....	637
35.7	PowerShell für Gruppenrichtlinienskripte .....	638
35.8	Einblicke in die Interna der Pipeline-Verarbeitung .....	640
	<b>Teil C: PowerShell im Praxiseinsatz .....</b>	<b>643</b>
<b>36</b>	<b>Dateisystem .....</b>	<b>645</b>
36.1	Laufwerke .....	646
36.2	Ordnerinhalte .....	651
36.3	Dateieigenschaften verändern .....	653
36.4	Eigenschaften ausführbarer Dateien .....	654
36.5	Kurznamen .....	656
36.6	Lange Pfade .....	656
36.7	Dateisystemoperationen .....	657
36.8	Praxisbeispiel: Zufällige Dateisystemstruktur erzeugen .....	658
36.9	Praxisbeispiel: Leere Ordner löschen .....	659
36.10	Einsatz von Robocopy .....	660
36.11	Dateisystemkataloge .....	663
36.12	Papierkorb leeren .....	664
36.13	Dateieigenschaften lesen .....	664
36.14	Praxisbeispiel: Fotos nach Aufnahmedatum sortieren .....	665

36.15	Datei-Hash .....	666
36.16	Finden von Duplikaten .....	667
36.17	Verknüpfungen im Dateisystem .....	669
36.18	Komprimierung .....	674
36.19	Dateisystemfreigaben .....	676
36.20	Überwachung des Dateisystems .....	687
36.21	Dateiversionsverlauf .....	688
36.22	Windows Explorer öffnen .....	689
36.23	Windows Server Backup .....	689
<b>37</b>	<b>Festplattenverschlüsselung mit BitLocker .....</b>	<b>693</b>
37.1	Übersicht über das BitLocker-Modul .....	694
37.2	Verschlüsseln eines Laufwerks .....	695
<b>38</b>	<b>Dokumente .....</b>	<b>697</b>
38.1	Textdateien .....	697
38.2	CSV-Dateien .....	698
38.3	Analysieren von Textdateien .....	701
38.4	INI-Dateien .....	704
38.5	XML-Dateien .....	705
38.6	HTML-Dateien .....	713
38.7	Binärdateien .....	713
<b>39</b>	<b>Datenbanken .....</b>	<b>715</b>
39.1	ADO.NET-Grundlagen .....	715
39.2	Beispieldatenbank .....	721
39.3	Datenzugriff mit den Bordmitteln der PowerShell .....	722
39.4	Hilfsroutinen für den Datenbankzugriff (DBUtil.ps1) .....	733
39.5	Datenzugriff mit den PowerShell-Erweiterungen .....	736
39.6	Datenbankzugriff mit SQLPS .....	740
39.7	Datenbankzugriff mit SQLPSX .....	740
<b>40</b>	<b>Microsoft-SQL-Server-Administration .....</b>	<b>741</b>
40.1	PowerShell-Integration im SQL Server Management Studio .....	742
40.2	SQL-Server-Laufwerk „SQLSERVER:“ .....	743
40.3	Die SQLPS-Commandlets .....	746
40.4	Die SQL Server Management Objects (SMO) .....	748
40.5	SQLPSX .....	751
40.6	Microsoft-SQL-Server-Administration mit der PowerShell in der Praxis .....	759

<b>41</b>	<b>ODBC-Datenquellen</b>	<b>765</b>
41.1	ODBC-Treiber und -Datenquellen auflisten	766
41.2	Anlegen einer ODBC-Datenquelle	767
41.3	Zugriff auf eine ODBC-Datenquelle	768
<b>42</b>	<b>Registrierungsdatenbank (Registry)</b>	<b>771</b>
42.1	Schlüssel auslesen	771
42.2	Schlüssel anlegen und löschen	772
42.3	Laufwerke definieren	772
42.4	Werte anlegen und löschen	773
42.5	Werte auslesen	774
42.6	Praxisbeispiel: Windows-Explorer-Einstellungen	774
42.7	Praxisbeispiel: Massenanlegen von Registry-Schlüsseln	775
<b>43</b>	<b>Computer- und Betriebssystemverwaltung</b>	<b>777</b>
43.1	Computerinformationen	777
43.2	Versionsnummer des Betriebssystems	779
43.3	Zeitdauer seit dem letzten Start des Betriebssystems	779
43.4	BIOS- und Startinformationen	780
43.5	Windows-Produktaktivierung	781
43.6	Umgebungsvariablen	781
43.7	Schriftarten	784
43.8	Computernamen und Domäne	784
43.9	Herunterfahren und Neustarten	785
43.10	Windows Updates installieren	786
43.11	Wiederherstellungspunkte verwalten	789
<b>44</b>	<b>Windows Defender</b>	<b>791</b>
<b>45</b>	<b>Hardwareverwaltung</b>	<b>793</b>
45.1	Hardwarebausteine	793
45.2	Plug-and-Play-Geräte	795
45.3	Druckerverwaltung (ältere Betriebssysteme)	795
45.4	Druckerverwaltung (seit Windows 8 und Windows Server 2012)	796
<b>46</b>	<b>Softwareverwaltung</b>	<b>799</b>
46.1	Softwareinventarisierung	799
46.2	Installation von Anwendungen	802
46.3	Deinstallation von Anwendungen	803
46.4	Praxisbeispiel: Installationstest	803
46.5	Installationen mit PowerShell Package Management („OneGet“)	804

46.6	Versionsnummer ermitteln .....	807
46.7	Servermanager .....	808
46.8	Windows-Features installieren auf Windows-Clientbetriebssystemen .....	819
46.9	Praxisbeispiel: IIS-Installation .....	822
46.10	Softwareeinschränkungen mit dem PowerShell-Modul „AppLocker“ .....	824
<b>47</b>	<b>Prozessverwaltung .....</b>	<b>831</b>
47.1	Prozesse auflisten .....	831
47.2	Prozesse starten .....	832
47.3	Prozesse mit vollen Administratorrechten starten .....	833
47.4	Prozesse unter einem anderen Benutzerkonto starten .....	834
47.5	Prozesse beenden .....	835
47.6	Warten auf das Beenden einer Anwendung .....	836
<b>48</b>	<b>Windows- Systemdienste .....</b>	<b>837</b>
48.1	Dienste auflisten .....	837
48.2	Dienstzustand ändern .....	840
48.3	Diensteigenschaften ändern .....	840
48.4	Dienste hinzufügen .....	841
48.5	Dienste entfernen .....	842
<b>49</b>	<b>Netzwerk .....</b>	<b>843</b>
49.1	Netzwerkkonfiguration (ältere Betriebssysteme) .....	843
49.2	Netzwerkkonfiguration (ab Windows 8 und Windows Server 2012) .....	845
49.3	DNS-Client-Konfiguration .....	848
49.4	DNS-Namensauflösung .....	851
49.5	Erreichbarkeit prüfen (Ping) .....	853
49.6	Windows Firewall .....	854
49.7	Remote Desktop (RDP) einrichten .....	860
49.8	E-Mails senden (SMTP) .....	861
49.9	Auseinandernehmen von E-Mail-Adressen .....	863
49.10	Abruf von Daten von einem HTTP-Server .....	863
49.11	Praxisbeispiel: Linkprüfer für eine Website .....	865
49.12	Aufrufe von SOAP-Webdiensten .....	868
49.13	Aufruf von REST-Diensten .....	870
49.14	Aufrufe von OData-Diensten .....	872
49.15	Hintergrunddatentransfer mit BITS .....	873
<b>50</b>	<b>Ereignisprotokolle (Event Log) .....</b>	<b>877</b>

<b>51</b>	<b>Leistungsdaten (Performance Counter)</b>	<b>881</b>
51.1	Zugriff auf Leistungsindikatoren über WMI	881
51.2	Get-Counter	882
<b>52</b>	<b>Sicherheitseinstellungen</b>	<b>885</b>
52.1	Aktueller Benutzer	885
52.2	Grundlagen	886
52.3	Zugriffsrechtelisten auslesen	891
52.4	Einzelne Rechteinträge auslesen	892
52.5	Besitzer auslesen	894
52.6	Benutzer und SID	894
52.7	Hinzufügen eines Rechteintrags zu einer Zugriffsrechteliste	897
52.8	Entfernen eines Rechteintrags aus einer Zugriffsrechteliste	900
52.9	Zugriffsrechteliste übertragen	901
52.10	Zugriffsrechteliste über SDDL setzen	902
52.11	Zertifikate verwalten	903
<b>53</b>	<b>Optimierungen und Problemlösungen</b>	<b>907</b>
53.1	PowerShell-Modul „TroubleshootingPack“	907
53.2	PowerShell-Modul „Best Practices“	911
<b>54</b>	<b>Active Directory</b>	<b>913</b>
54.1	Benutzer- und Gruppenverwaltung mit WMI	914
54.2	Einführung in System.DirectoryServices	915
54.3	Basiseigenschaften	926
54.4	Benutzer- und Gruppenverwaltung im Active Directory	928
54.5	Verwaltung der Organisationseinheiten	936
54.6	Suche im Active Directory	937
54.7	Navigation im Active Directory mit den PowerShell Extensions	944
54.8	Verwendung der Active-Directory-Erweiterungen von <a href="http://www.IT-Visions.de">www.IT-Visions.de</a>	945
54.9	PowerShell-Modul „Active Directory“ (ADPowerShell)	947
54.10	PowerShell-Modul „ADDSDeployment“	975
54.11	Informationen über die Active-Directory-Struktur	978
<b>55</b>	<b>Gruppenrichtlinien</b>	<b>981</b>
55.1	Verwaltung der Gruppenrichtlinien	982
55.2	Verknüpfung der Gruppenrichtlinien	983
55.3	Gruppenrichtlinienberichte	985
55.4	Gruppenrichtlinienvererbung	986
55.5	Weitere Möglichkeiten	987

<b>56</b>	<b>Lokale Benutzer und Gruppen</b>	<b>989</b>
56.1	Modul „Microsoft.PowerShell.LocalAccounts“	989
56.2	Lokale Benutzerverwaltung in älteren PowerShell-Versionen	991
<b>57</b>	<b>Microsoft Exchange Server</b>	<b>993</b>
57.1	Daten abrufen	993
57.2	Postfächer verwalten	994
57.3	Öffentliche Ordner verwalten	995
<b>58</b>	<b>Internet Information Services (IIS)</b>	<b>997</b>
58.1	Überblick	997
58.2	Navigationsprovider	999
58.3	Anlegen von Websites	1001
58.4	Praxisbeispiel: Massenanlegen von Websites	1002
58.5	Ändern von Website-Eigenschaften	1005
58.6	Anwendungspool anlegen	1005
58.7	Virtuelle Verzeichnisse und IIS-Anwendungen	1006
58.8	Website-Zustand ändern	1007
58.9	Anwendungspools starten und stoppen	1007
58.10	Löschen von Websites	1008
<b>59</b>	<b>Virtuelle Systeme mit Hyper-V</b>	<b>1009</b>
59.1	Das Hyper-V-Modul von Microsoft	1010
59.2	Die ersten Schritte mit dem Hyper-V-Modul	1012
59.3	Virtuelle Maschinen anlegen	1016
59.4	Umgang mit virtuellen Festplatten	1022
59.5	Konfiguration virtueller Maschinen	1025
59.6	Dateien kopieren in virtuelle Systeme	1029
59.7	PowerShell Management Library for Hyper-V (für ältere Betriebssysteme)	1031
<b>60</b>	<b>Windows Nano Server</b>	<b>1035</b>
60.1	Das Konzept von Nano Server	1035
60.2	Einschränkungen von Nano Server	1037
60.3	Varianten des Nano Servers	1039
60.4	Installation eines Nano Servers	1039
60.5	Docker-Image	1041
60.6	Fernverwaltung mit PowerShell	1041
60.7	Windows Update auf einem Nano Server	1044
60.8	Nachträgliche Paketinstallation	1044
60.9	Abgespeckter IIS unter Nano Server	1046
60.10	Nano-Serververwaltung aus der Cloud heraus	1047

<b>61</b>	<b>Docker-Container</b>	<b>1049</b>
61.1	Docker-Varianten für Windows	1050
61.2	Docker-Installation auf Windows 10	1051
61.3	Docker-Installation auf Windows Server 2016	1053
61.4	Installation von „Docker for Windows“	1054
61.5	Docker-Registries	1056
61.6	Docker-Images laden	1056
61.7	Container starten	1057
61.8	Container-Identifikation	1058
61.9	Container mit Visual Studio	1059
61.10	Befehle in einem Container ausführen	1061
61.11	Ressourcenbeschränkungen für Container	1063
61.12	Dateien zwischen Container und Host kopieren	1063
61.13	Dockerfile	1063
61.14	Docker-Netzwerke	1064
61.15	Container anlegen, ohne sie zu starten	1065
61.16	Container starten und stoppen	1065
61.17	Container beenden und löschen	1065
61.18	Images löschen	1066
61.19	Images aus Containern erstellen	1066
61.20	.NET Core-Container	1066
61.21	Images verbreiten	1069
61.22	Azure Container Service (ACS)	1071
<b>62</b>	<b>Grafische Benutzeroberflächen (GUI)</b>	<b>1073</b>
62.1	Einfache Nachfragedialoge	1073
62.2	Einfache Eingabe mit Inputbox	1075
62.3	Komplexere Eingabemasken	1076
62.4	Universelle Objektdarstellung	1078
62.5	WPF PowerShell Kit (WPK)	1079
62.6	Direkte Verwendung von WPF	1087
<b>Teil D: Profiwissen – Erweitern der PowerShell</b>		<b>1089</b>
<b>63</b>	<b>Entwicklung von Commandlets in der PowerShell-Skriptsprache</b>	<b>1091</b>
63.1	Aufbau eines skriptbasierten Commandlets	1091
63.2	Verwendung per Dot Sourcing	1093
63.3	Parameterfestlegung	1094
63.4	Fortgeschrittene Funktion (Advanced Function)	1100

63.5	Mehrere Parameter und Parametersätze .....	1102
63.6	Unterstützung für Sicherheitsabfragen (-Whatif und -Confirm) .....	1104
63.7	Kaufmännisches Beispiel: Test-CustomerID .....	1106
63.8	Erweitern bestehender Commandlets durch Proxy-Commandlets .....	1109
63.9	Dokumentation .....	1115
<b>64</b>	<b>Entwicklung eigener Commandlets mit C# .....</b>	<b>1119</b>
64.1	Technische Voraussetzungen .....	1120
64.2	Grundkonzept der .NET-basierten Commandlets .....	1121
64.3	Schrittweise Erstellung eines minimalen Commandlets .....	1123
64.4	Erstellung eines Commandlets mit einem Rückgabeobjekt .....	1131
64.5	Erstellung eines Commandlets mit mehreren Rückgabeobjekten .....	1133
64.6	Erstellen eines Commandlets mit Parametern .....	1137
64.7	Verarbeiten von Pipeline-Eingaben .....	1139
64.8	Verkettung von Commandlets .....	1142
64.9	Fehlersuche in Commandlets .....	1146
64.10	Statusinformationen .....	1149
64.11	Unterstützung für Sicherheitsabfragen (-whatif und -confirm) .....	1154
64.12	Festlegung der Hilfeinformationen .....	1156
64.13	Erstellung von Commandlets für den Zugriff auf eine Geschäftsanwendung ..	1160
64.14	Konventionen für Commandlets .....	1161
64.15	Weitere Möglichkeiten .....	1163
<b>65</b>	<b>PowerShell-Module erstellen .....</b>	<b>1165</b>
65.1	Erstellen eines Skriptmoduls .....	1165
65.2	Praxisbeispiel: Umwandlung einer Skriptdatei in ein Modul .....	1167
65.3	Erstellen eines Moduls mit Binärdateien .....	1167
65.4	Erstellen eines Moduls mit Manifest .....	1168
65.5	Erstellung eines Manifest-Moduls mit Visual Studio .....	1175
<b>66</b>	<b>Hosting der PowerShell .....</b>	<b>1177</b>
66.1	Voraussetzungen für das Hosting .....	1178
66.2	Hosting mit PSHost .....	1179
66.3	Vereinfachtes Hosting seit PowerShell 2.0 .....	1182
<b>Anhang A: Crashkurs „Objektorientierung“ .....</b>		<b>1185</b>
<b>Anhang B: Crashkurs .NET .....</b>		<b>1193</b>
B.1	Was ist das .NET Framework? .....	1195
B.2	Was ist .NET Core? .....	1196
B.3	Eigenschaften von .NET .....	1197

B.4	.NET-Klassen .....	1198
B.5	Namensgebung von .NET-Klassen (Namensräume) .....	1198
B.6	Namensräume und Softwarekomponenten .....	1200
B.7	Bestandteile einer .NET-Klasse .....	1201
B.8	Vererbung .....	1202
B.9	Schnittstellen .....	1203
<b>Anhang C: Literatur .....</b>		<b>1204</b>
<b>Anhang D: Weitere Informationen im Internet .....</b>		<b>1207</b>
<b>Anhang E: Abkürzungsverzeichnis .....</b>		<b>1209</b>
<b>Stichwortverzeichnis .....</b>		<b>1235</b>

# Vorwort

Liebe Leserin, lieber Leser,

willkommen zur aktuellen Auflage des PowerShell-Buchs! Es handelt sich hierbei um die dritte Auflage des Windows PowerShell 5-Buches und die siebte Auflage des PowerShell-Buches insgesamt, das erstmalig 2007 bei Addison-Wesley erschienen ist. Das vor Ihnen liegende Buch behandelt die Windows PowerShell in der Version 5.1 sowie die PowerShell Core in der Version 6.0/6.1 von Microsoft sowie ergänzende Werkzeuge von Microsoft und Drittanbietern (z. B. PowerShell Community Extensions). Das Buch ist aber auch geeignet, wenn Sie noch PowerShell 2.0, 3.0, 4.0 oder 5.0 einsetzen. Welche Funktionen neu hinzugekommen sind, wird jeweils erwähnt.

## ■ Wer bin ich?

Mein Name ist Holger Schwichtenberg, ich bin derzeit 45 Jahre alt und habe im Fachgebiet Wirtschaftsinformatik promoviert. Ich lebe (in Essen, im Herzen des Ruhrgebiets) davon, dass mein Team und ich im Rahmen unserer Firma [www.IT-Visions.de](http://www.IT-Visions.de) anderen Unternehmen bei der Entwicklung von .NET-, Web- und PowerShell-Anwendungen beratend und schulend zur Seite stehen. Zudem entwickeln wir im Rahmen der 5Minds IT-Solutions GmbH & Co. KG Software ([www.5Minds.de](http://www.5Minds.de)) im Auftrag von Kunden in zahlreichen Branchen.

Es ist mein Hobby und Nebenberuf, IT-Fachbücher zu schreiben. Dieses Buch ist, unter Mitzählung aller nennenswerten Neuauflagen, das 68. Buch, das ich allein oder mit Co-Autoren geschrieben habe. Meine weiteren Hobbys sind Mountain Biking, Lauf-Sport, Fotografie und Reisen.

Natürlich verstehe ich das Bücherschreiben auch als Werbung für die Arbeit unserer Unternehmen und wir hoffen, dass der ein oder andere von Ihnen uns beauftragen wird, Ihre Organisation durch Beratung, Schulung und Auftragsentwicklung zu unterstützen.

## ■ Wer sind Sie?

Damit Sie den optimalen Nutzen aus diesem Buch ziehen können, möchte ich – so genau es mir möglich ist – beschreiben, an wen sich dieses Buch richtet. Hierzu habe ich einen Fragebogen ausgearbeitet, mit dem Sie schnell erkennen können, ob das Buch für Sie geeignet ist.

Sind Sie Systemadministrator in einem Windows-Netzwerk?	<input type="radio"/> Ja	<input type="radio"/> Nein
Laufen die für Sie relevanten Computer mit den von PowerShell 3.0, 4.0, 5.x oder 6.x unterstützten Betriebssystemen? (Windows 7/8/8.1/10, Windows Server 2008/2008 R2/2012/2012 R2/2016) Hinweis: Die PowerShell Core 6.0 für Linux und MacOS wird nur als Randthema kurz in diesem Buch behandelt, da es hier bislang kaum Befehle für die PowerShell gibt!	<input type="radio"/> Ja	<input type="radio"/> Nein
Sie besitzen zumindest rudimentäre Grundkenntnisse im Bereich des (objektorientierten) Programmierens?	<input type="radio"/> Ja	<input type="radio"/> Nein
Wünschen Sie einen kompakten Überblick über die Architektur, Konzepte und Anwendungsfälle der PowerShell?	<input type="radio"/> Ja	<input type="radio"/> Nein
Sie können auf Schritt-für-Schritt-Anleitungen verzichten?	<input type="radio"/> Ja	<input type="radio"/> Nein
Sie können auf formale Syntaxbeschreibungen verzichten und lernen lieber an aussagekräftigen Beispielen?	<input type="radio"/> Ja	<input type="radio"/> Nein
Sie erwarten nicht, dass in diesem Buch <b>alle</b> Möglichkeiten der PowerShell detailliert beschrieben werden?	<input type="radio"/> Ja	<input type="radio"/> Nein
Sind Sie, nachdem Sie ein Grundverständnis durch dieses Buch gewonnen haben, bereit, Detailfragen in der Dokumentation der PowerShell, von .NET und WMI nachzuschlagen, da das Buch auf 1200 Seiten nicht alle Details erläutern kann?	<input type="radio"/> Ja	<input type="radio"/> Nein

Wenn Sie alle obigen Fragen mit „Ja“ beantwortet haben, ist das Buch richtig für Sie. In anderen Fällen sollten Sie sich erst mit einführender Literatur beschäftigen.

Die PowerShell Core 6.0, die zum letzten Redaktionsschluss noch auf dem Beta-Stand war, ist inzwischen erschienen. Das Buch wurde auf die RTM-Version 6.0 sowie die Release Candidate-Version von PowerShell Core 6.1 aktualisiert. Zudem wurde ein Unterkapitel zur Installation von Windows-Features in Client-Betriebssystemen im Kapitel „Teil C/Softwareverwaltung“ sowie ein Praxisbeispiel zur Installation des Webservers „IIS“ ergänzt. Außerdem wurde das Kapitel „Verwendung von .NET-Klassen“ erweitert.

## ■ Was ist neu in diesem Buch?

Gegenüber der vorherigen Auflage zur PowerShell 5.0 wurde das Buch um die neuen Funktionen in Windows PowerShell 5.1 sowie PowerShell Core 6.0 erweitert und inhaltlich optimiert. Praxiseinsatzkapitel wurden ergänzt zu Windows Update, Windows Nano Server und Docker-Containern. Zudem wurden die bestehenden Inhalte des Buchs an vielen Stellen erweitert und didaktisch optimiert.

## ■ Sind in diesem Buch alle Features der PowerShell beschrieben?

Die PowerShell umfasst mittlerweile über 1500 Commandlets mit jeweils zahlreichen Optionen. Zudem gibt es unzählige Erweiterungen mit vielen hundert weiteren Commandlets. Zudem existieren zahlreiche Zusatzwerkzeuge. Es ist allein schon aufgrund der Vorgaben des Verlags für den Umfang des Buchs nicht möglich, alle Commandlets und Parameter hier auch nur zu erwähnen. Zudem habe ich – obwohl ich selbst fast jede Woche mit der PowerShell in der Praxis arbeite – immer noch nicht alle Commandlets und alle Parameter jemals eingesetzt. Ich beschreibe in diesem Buch, was ich selbst in der Praxis, in meinen Schulungen und bei Kundeneinsätzen verwende. Es macht auch keinen Sinn, jedes Detail der PowerShell hier zu dokumentieren. Stattdessen gebe ich Ihnen **Hilfe zur Selbsthilfe**, damit Sie die Konzepte gut verstehen und sich dann Sonderfälle selbst erarbeiten können.

## ■ Wie aktuell ist dieses Buch?

Die Informationstechnik hat sich immer schon schnell verändert. Seit aber auch Microsoft das Themen „Agilität“ und „Open Source“ für sich entdeckt hat, ist die Entwicklung nicht mehr schnell, sondern zum Teil rasant:

- Es erscheinen in kurzer Abfolge immer neue Produkte.
- Produkte erscheinen schon in frühen Produktstadien als „Preview“ mit Versionsnummern wie 0.1.
- Produkte ändern sich häufig. Aufwärts- und Abwärtskompatibilität ist kein Ziel mehr. Es wird erwartet, dass Sie Ihre Lösungen ständig den neuen Gegebenheiten anpassen.
- Produkte werden nicht mehr so ausführlich dokumentiert wie früher. Teilweise erscheint Dokumentation erst deutlich nach dem Erscheinen der Software.
- Produkte werden schnell auch wieder abgekündigt, wenn sie sich aus der Sicht der Hersteller bzw. aufgrund des Nutzerfeedbacks nicht bewährt haben.

Unter diesen neuen Einflüssen steht natürlich auch dieses etablierte Buch. Leider kann man ein Buch nicht so schnell ändern wie Software.

Daher kann es passieren, dass – auch schon kurz nach dem Erscheinen dieses Buchs – einzelne Informationen in diesem Buch nicht mehr zu neueren Versionen passen. Wenn Sie so einen Fall feststellen, schreiben Sie bitte eine Nachricht an mich im Leser-Portal (siehe unten). Ich werde dies dann in Neuauflagen des Buchs berücksichtigen.

## ■ Wem ist zu danken?

Folgenden Personen möchte ich meinen Dank für ihre Mitwirkung an diesem Buch aussprechen:

- meinem Kollegen und Freund Peter Monadjemi, der rund 100 Seiten mit Beispielen zu der Vor-Vor-Vor-Auflage dieses Buchs beigetragen hat (Themen: Workflows, Bitlocker, ODBC, Hyper-V, DNS-Client, Firewall und SQL-Server-Administration),
- Frau Sylvia Hasselbach, die mich schon seit 20 Jahren als Lektorin begleitet und die dieses Buchprojekt beim Carl Hanser Verlag koordiniert und vermarktet,
- Frau Sandra Gottmann, die meine Tippfehler gefunden und sprachliche Ungenauigkeiten eliminiert hat,
- meiner Frau und meinen Kindern dafür, dass sie mir das Umfeld geben, um neben meinem Hauptberuf an Büchern wie diesem zu arbeiten.

## ■ Woher bekommen Sie die Beispiele aus diesem Buch?

Unter <http://www.powershell-doktor.de/leser> biete ich ein **ehrenamtlich betriebenes** Webportal für Leser meiner Bücher an. In diesem Portal können Sie

- die Codebeispiele aus diesem Buch in einem Archiv herunterladen,
- eine PowerShell-Kurzreferenz „Cheat Sheet“ (zwei DIN-A4-Seiten als Hilfe für die tägliche Arbeit) kostenlos herunterladen,
- Feedback zu diesem Buch geben (Bewertung abgeben und Fehler melden) und
- technische Fragen in einem Webforum stellen.

Alle registrierten Leser erhalten auch Einladungen zu kostenlosen Community-Veranstaltungen sowie Vergünstigungen bei unseren öffentlichen Seminaren zu .NET und zur PowerShell. Bei der Registrierung müssen Sie das Kennwort **Rogue One** angeben.

## ■ Wie sind die Programmcodebeispiele organisiert?

Die Beispiele sind im Archiv organisiert nach den Buchteilen und innerhalb der Buchteile nach Kapitelnamen (verkürzt). In diesem Buch wird für den Zugriff auf die Beispieldateien das X:-Laufwerk verwendet. Dies müssen Sie auf Ihre Situation anpassen!

```
PS T:\> dir x:\

Verzeichnis: x:\

Mode                LastWriteTime         Length Name
----                -
d-r---             29.06.2017   23:56         1_Basiswissen
d-r---             28.06.2017   17:09         2_Aufbauwissen
d-r---             02.06.2017   10:38         3_Einsatzgebiete
d-r---             30.06.2017   17:22         4_Profiwissen

PS T:\> dir x:\1_Basiswissen\

Verzeichnis: x:\1_Basiswissen

Mode                LastWriteTime         Length Name
----                -
d-----             29.06.2017   23:56         Aliase
d-r---             24.04.2017   09:52         Ausgaben
d-r---             30.05.2017   00:28         Commandlets
d-----             26.06.2017   10:40         ErsteSchritte
d-r---             29.06.2017   23:34         Hilfe
d-----             30.05.2017   20:59         Module
d-r---             26.03.2014   12:49         Navigation
d-r---             04.06.2017   11:21         Pipelining
d-----             30.05.2017   21:15         PowerShellLanguage
d-----             29.05.2017   23:57         PowerShell00P
d-----             30.06.2017   18:47         PSCore
d-r---             30.05.2017   20:46         Scripting
d-r---             26.03.2014   12:49         TippsAndTricks
d-r---             26.03.2014   12:49         Werkzeuge
d-r---             26.03.2014   12:49         WPS versus VBS
d-----             03.05.2016   14:12         Zeichenkettenbearbeitung
```

## ■ Wo können Sie sich schulen oder beraten lassen?

Unter der E-Mail-Adresse [kundenteam@IT-Visions.de](mailto:kundenteam@IT-Visions.de) stehen mein Team und ich für Anfragen bezüglich Schulung, Beratung und Entwicklungstätigkeiten zur Verfügung – nicht nur zum Thema PowerShell und .NET/.NET Core, sondern zu fast allen modernen Techniken der Entwicklung und des Betriebs von Software in großen Unternehmen. Wir besuchen Sie gerne in Ihrem Unternehmen an einem beliebigen Standort.

## ■ Zum Schluss des Vorworts ...

... wünsche ich Ihnen viel Spaß und Erfolg mit der PowerShell!

*Dr. Holger Schwichtenberg*

*Essen, im Oktober 2018*

# Über den Autor

## Dr. Holger Schwichtenberg



- Studienabschluss Diplom-Wirtschaftsinformatik an der Universität Essen
- Promotion an der Universität Essen im Gebiet komponentenbasierter Softwareentwicklung
- Seit 1996 selbstständig als unabhängiger Berater, Dozent, Softwarearchitekt und Fachjournalist
- Leiter des Berater- und Dozententeams bei *www.IT-Visions.de*



- Leitung der Softwareentwicklung im Bereich Microsoft/.NET bei der 5Minds IT-Solutions GmbH & Co. KG (*www.5minds.de*)



- Über 65 Fachbücher beim Carl Hanser Verlag, bei O'Reilly, Microsoft Press und Addison-Wesley sowie mehr als 950 Beiträge in Fachzeitschriften
- Gutachter in den Wettbewerbsverfahren der EU gegen Microsoft (2006–2009)
- Ständiger Mitarbeiter der Zeitschriften iX (seit 1999), dotnetpro (seit 2000) und Windows Developer (seit 2010) sowie beim Online-Portal *heise.de* (seit 2008)
- Regelmäßiger Sprecher auf nationalen und internationalen Fachkonferenzen (z.B. Microsoft TechEd, Microsoft Summit, Microsoft IT Forum, BASTA, BASTA-on-Tour, .NET Architecture Camp, Advanced Developers Conference, Developer Week, OOP, DOTNET Cologne, MD DevDays, Community in Motion, DOTNET-Konferenz, VS One, NRW.Conf, Net.Object Days, Windows Forum)Zertifikate und Auszeichnungen von Microsoft:
  - Bereits 14 mal ausgezeichnet als Microsoft Most Valuable Professional (MVP)
  - Zertifiziert als Microsoft Certified Solution Developer (MCSD)
- Thematische Schwerpunkte:
  - Softwarearchitektur, mehrschichtige Softwareentwicklung, Softwarekomponenten, SOA