

holger SCHWICHTENBERG

Für
PowerShell
2 bis 6

WINDOWS

PowerShell 5 und PowerShell Core 6

2. Auflage



DAS PRAXISBUCH



»Sehr gut« dotnetpro zur 1. Auflage



Im Internet: Codebeispiele, Forum,
PowerShell-Kurzreferenz

HANSER

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



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



Holger Schwichtenberg

Windows PowerShell 5 und PowerShell Core 6

Das Praxisbuch

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.



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.

© 2017 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sylvia Hasselbach

Copy editing: Sandra Gottmann, Nienberge

Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

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

Printed in Germany

Print-ISBN: 978-3-446-45331-9

E-Book-ISBN: 978-3-446-45397-5

Inhalt

Vorwort zur sechsten Auflage	XXIII
Über den Autor Dr. Holger Schwichtenberg	XXIX
Teil A: PowerShell-Basiswissen	1
1 Erste Schritte mit der PowerShell	3
1.1 Was ist die PowerShell?	3
1.2 Windows PowerShell versus PowerShell Core	4
1.3 Geschichte der PowerShell	4
1.4 Motivation zur PowerShell	6
1.5 Betriebssysteme mit vorinstallierter PowerShell	9
1.6 Windows PowerShell herunterladen und auf anderen Windows- Betriebssystemen installieren	10
1.7 Die Windows PowerShell testen	14
1.8 PowerShell Core installieren und testen	23
1.9 Woher kommen die Commandlets?	26
1.10 PowerShell Community Extensions (PSCX) herunterladen und installieren	27
1.11 Den Windows PowerShell-Editor „ISE“ verwenden	29
2 Architektur der PowerShell	33
3 Einzelbefehle der PowerShell	37
3.1 Commandlets	37
3.2 Aliase	50
3.3 Ausdrücke	58
3.4 Externe Befehle	59
3.5 Dateinamen	60

4	Hilfefunktionen	63
4.1	Auflisten der verfügbaren Befehle	63
4.2	Volltextsuche	65
4.3	Erläuterungen zu den Befehlen	66
4.4	Hilfe zu Parametern	67
4.5	Hilfe mit Show-Command	69
4.6	Hilfefenster	70
4.7	Allgemeine Hilfetexte	72
4.8	Aktualisieren der Hilfsdateien	73
4.9	Online-Hilfe	75
4.10	Fehlende Hilfetexte	76
4.11	Dokumentation der .NET-Klassen	77
5	Objektorientiertes Pipelining	81
5.1	Pipeline-Operator	81
5.2	.NET-Objekte in der Pipeline	82
5.3	Pipeline Processor	84
5.4	Pipelining von Parametern	85
5.5	Pipelining von klassischen Befehlen	88
5.6	Anzahl der Objekte in der Pipeline	89
5.7	Zeilenumbrüche in Pipelines	90
5.8	Zugriff auf einzelne Objekte aus einer Menge	90
5.9	Zugriff auf einzelne Werte in einem Objekt	92
5.10	Methoden ausführen	93
5.11	Analyse des Pipeline-Inhalts	95
5.12	Filtern	107
5.13	Zusammenfassung von Pipeline-Inhalten	110
5.14	„Kastrierung“ von Objekten in der Pipeline	111
5.15	Sortieren	112
5.16	Duplikate entfernen	113
5.17	Gruppierung	114
5.18	Berechnungen	116
5.19	Zwischenschritte in der Pipeline mit Variablen	116
5.20	Verzweigungen in der Pipeline	117
5.21	Vergleiche zwischen Objekten	119
5.22	Zusammenfassung	120
5.23	Praxisbeispiele	121
6	PowerShell-Skripte	123
6.1	Skriptdateien	123

6.2	Start eines Skripts.	125
6.3	Aliase für Skripte verwenden	126
6.4	Parameter für Skripte.	127
6.5	Skripte dauerhaft einbinden (Dot Sourcing)	128
6.6	Das aktuelle Skriptverzeichnis	129
6.7	Sicherheitsfunktionen für PowerShell-Skripte	129
6.8	Anforderungsdefinitionen von Skripten	132
6.9	Skripte anhalten	132
6.10	Versionierung und Versionsverwaltung von Skripten	133
7	PowerShell-Skriptsprache	135
7.1	Hilfe zur PowerShell-Skriptsprache	135
7.2	Befehlstrennung	136
7.3	Kommentare	136
7.4	Variablen	137
7.5	Variablenbedingungen	147
7.6	Zahlen	148
7.7	Zeichenketten (Strings)	150
7.8	Reguläre Ausdrücke	159
7.9	Datum und Uhrzeit.	166
7.10	Arrays	167
7.11	ArrayList	170
7.12	Assoziative Arrays (Hash-Tabellen)	171
7.13	Operatoren	172
7.14	Überblick über die Kontrollkonstrukte	177
7.15	Schleifen	177
7.16	Bedingungen	182
7.17	Unterroutinen (Prozedur/Funktionen)	184
7.18	Eingebaute Funktionen	191
7.19	Fehlerbehandlung.	191
7.20	Objektorientiertes Programmieren mit Klassen	199
8	Ausgaben	203
8.1	Ausgabe-Commandlets.	203
8.2	Benutzerdefinierte Tabellenformatierung	206
8.3	Benutzerdefinierte Listenausgabe.	208
8.4	Mehrpaltige Ausgabe	208
8.5	Out-GridView.	209
8.6	Standardausgabe	211
8.7	Einschränkung der Ausgabe	213

8.8	Seitenweise Ausgabe	213
8.9	Ausgabe einzelner Werte	214
8.10	Details zum Ausgabeoperator	216
8.11	Ausgabe von Methodenergebnissen und Unterobjekten in Pipelines	220
8.12	Ausgabe von Methodenergebnissen und Unterobjekten in Zeichenketten	220
8.13	Unterdrückung der Ausgabe	221
8.14	Ausgaben an Drucker	222
8.15	Ausgaben in Dateien	222
8.16	Umleitungen (Redirection)	223
8.17	Fortschrittsanzeige	223
8.18	Sprachausgabe	224
9	Das PowerShell-Navigationsmodell	227
9.1	Einführungsbeispiel: Navigation in der Registrierungsdatenbank	227
9.2	Provider und Laufwerke	228
9.3	Navigationsbefehle	231
9.4	Pfadangaben	231
9.5	Beispiel	233
9.6	Eigene Laufwerke definieren	234
10	Fernausführung (Remoting)	235
10.1	RPC-Fernabfrage ohne WS-Management	236
10.2	Anforderungen an PowerShell Remoting	237
10.3	Rechte für PowerShell-Remoting	238
10.4	Einrichten von PowerShell Remoting	239
10.5	Überblick über die Fernausführungs-Commandlets	241
10.6	Interaktive Fernverbindungen im Telnet-Stil	242
10.7	Fernausführung von Befehlen	243
10.8	Parameterübergabe an die Fernausführung	247
10.9	Fernausführung von Skripten	248
10.10	Ausführung auf mehreren Computern	249
10.11	Sitzungen	250
10.12	Implizites Remoting	255
10.13	Zugriff auf entfernte Computer außerhalb der eigenen Domäne	256
10.14	Verwaltung des WS-Management-Dienstes	259
10.15	PowerShell Direct für Hyper-V	261
10.16	Praxisbeispiel zu PowerShell Direct	263
11	PowerShell-Werkzeuge	267
11.1	PowerShell-Standardkonsole	267

11.2	PowerShell Integrated Scripting Environment (ISE)	276
11.3	PowerShell Script Analyzer	286
11.4	PowerShell Analyzer	292
11.5	PowerShell Tools for Visual Studio	293
11.6	PowerShell Pro Tools for Visual Studio	294
11.7	NuGet Package Manager	295
11.8	PowerShell-Erweiterung für Visual Studio Code	295
11.9	PowerShell Web Access (PSWA)	298
11.10	Azure Cloud Shell	304
11.11	ISE Steroids	304
11.12	PowerShellPlus	305
11.13	PoshConsole	308
11.14	PowerGUI	309
11.15	PrimalScript	310
11.16	PowerShell Help	312
11.17	CIM Explorer for PowerShell ISE	312
11.18	PowerShell Help Reader	313
11.19	PowerShell Remoting	314
12	Windows PowerShell Core 5.1 in Windows Nano Server	315
13	PowerShell Core 6.x für Windows, Linux und MacOS	317
13.1	Funktionsumfang der PowerShell Core	318
13.2	PowerShell Core-Konsole	324
13.3	VSCoDe-PowerShell	325
13.4	Verwendung auf Linux und MacOS	326
13.5	PowerShell-Remoting via SSH	330
Teil B: PowerShell-Aufbauwissen		333
14	Verwendung von .NET-Klassen	335
14.1	Microsoft Developer Network (MSDN)	335
14.2	Erzeugen von Instanzen	336
14.3	Parameterbehaftete Konstruktoren	338
14.4	Initialisierung von Objekten	339
14.5	Nutzung von Attributen und Methoden	340
14.6	Statische Mitglieder in .NET-Klassen und statische .NET-Klassen	342
14.7	Generische Klassen nutzen	346
14.8	Zugriff auf bestehende Objekte	347
14.9	Laden von Assemblies	347
14.10	Objektanalyse	350

14.11	Auflistungen (Enumerationen)	350
14.12	Verknüpfen von Aufzählungswerten	351
15	Verwendung von COM-Klassen	353
15.1	Erzeugen von COM-Instanzen	353
15.2	Nutzung von Attributen und Methoden	354
15.3	Liste aller COM-Klassen	355
15.4	Holen bestehender COM-Instanzen	356
15.5	Distributed COM (DCOM)	356
16	Zugriff auf die Windows Management Instrumentation (WMI) ...	357
16.1	Einführung in WMI	357
16.2	WMI in der PowerShell	384
16.3	Open Management Infrastructure (OMI)	386
16.4	Abruf von WMI-Objektmengen	386
16.5	Fernzugriffe	387
16.6	Filtern und Abfragen	387
16.7	Liste aller WMI-Klassen	391
16.8	Hintergrundwissen: WMI-Klassenprojektion mit dem PowerShell-WMI-Objektadapter	392
16.9	Beschränkung der Ausgabeliste bei WMI-Objekten	396
16.10	Zugriff auf einzelne Mitglieder von WMI-Klassen	398
16.11	Werte setzen in WMI-Objekten	398
16.12	Umgang mit WMI-Datumsangaben	400
16.13	Methodenaufrufe	401
16.14	Neue WMI-Instanzen erzeugen	402
16.15	Instanzen entfernen	403
16.16	Commandlet Definition XML-Datei (CDXML)	404
17	Dynamische Objekte	407
17.1	Erweitern bestehender Objekte	407
17.2	Komplett dynamische Objekte	409
18	Einbinden von C# und Visual Basic .NET	411
19	Win32-API-Aufrufe	413
20	Benutzereingaben	417
20.1	Read-Host	417
20.2	Benutzerauswahl	418
20.3	Grafischer Eingabedialog	419
20.4	Dialogfenster	420

20.5	Authentifizierungsdialog	420
20.6	Zwischenablage (Clipboard)	422
21	Fehlersuche	425
21.1	Detailinformationen	425
21.2	Einzelstschrittmodus	426
21.3	Zeitmessung	427
21.4	Ablaufverfolgung (Tracing)	428
21.5	Erweiterte Protokollierung aktivieren	429
21.6	Script-Debugging in der ISE	431
21.7	Kommandozeilenbasiertes Script-Debugging	431
22	Transaktionen	433
22.1	Commandlets für Transaktionen	433
22.2	Start und Ende einer Transaktion	434
22.3	Zurücksetzen der Transaktion	435
22.4	Mehrere Transaktionen	436
23	Standardeinstellungen ändern mit Profilskripten	437
23.1	Profilpfade	437
23.2	Ausführungsreihenfolge	439
23.3	Beispiel für eine Profildatei	439
23.4	Starten der PowerShell ohne Profilskripte	440
24	Digitale Signaturen für PowerShell-Skripte	441
24.1	Zertifikat erstellen	441
24.2	Skripte signieren	443
24.3	Verwenden signierter Skripte	444
24.4	Mögliche Fehlerquellen	445
25	Hintergrundaufträge („Jobs“)	447
25.1	Voraussetzungen	447
25.2	Architektur	448
25.3	Starten eines Hintergrundauftrags	448
25.4	Hintergrundaufträge abfragen	449
25.5	Warten auf einen Hintergrundauftrag	450
25.6	Abbrechen und Löschen von Aufträgen	450
25.7	Analyse von Fehlermeldungen	451
25.8	Fernausführung von Hintergrundaufträgen	451
25.9	Praxisbeispiel	452

26	Geplante Aufgaben und zeitgesteuerte Jobs	455
26.1	Geplante Aufgaben (Scheduled Tasks)	455
26.2	Zeitgesteuerte Jobs	459
27	PowerShell-Workflows	465
27.1	Ein erstes Beispiel	465
27.2	Unterschiede zu einer Function bzw. einem Skript	470
27.3	Einschränkungen bei Workflows	470
27.4	Workflows in der Praxis	472
27.5	Workflows in Visual Studio erstellen	479
28	Ereignissystem	497
28.1	WMI-Ereignisse	497
28.2	WMI-Ereignisabfragen	497
28.3	WMI-Ereignisse seit PowerShell 1.0	499
28.4	Registrieren von WMI-Ereignisquellen seit PowerShell 2.0	500
28.5	Auslesen der Ereignisliste	501
28.6	Reagieren auf Ereignisse	503
28.7	WMI-Ereignisse ab PowerShell-Version 3.0	505
28.8	Registrieren von .NET-Ereignissen	505
28.9	Erzeugen von Ereignissen	506
29	Datenbereiche und Datendateien	509
29.1	Datenbereiche	509
29.2	Datendateien	511
29.3	Mehrsprachigkeit/Lokalisierung	512
30	Desired State Configuration (DSC)	515
30.1	Grundprinzipien	516
30.2	DSC für Linux	516
30.3	Ressourcen	517
30.4	Verfügbare DSC-Ressourcen	517
30.5	Eigenschaften einer Ressource	520
30.6	Aufbau eines DSC-Dokuments	520
30.7	Commandlets für die Arbeit mit DSC	521
30.8	Ein erstes DSC-Beispiel	521
30.9	Kompilieren und Anwendung eines DSC-Dokuments	522
30.10	Variablen in DSC-Dateien	524
30.11	Parameter für DSC-Dateien	525
30.12	Konfigurationsdaten	526
30.13	Entfernen einer DSC-Konfiguration	529

30.14 DSC Pull Server	532
30.15 DSC-Praxisbeispiel 1: IIS installieren	540
30.16 DSC-Praxisbeispiel 2: Software installieren	541
30.17 DSC-Praxisbeispiel 3: Software deinstallieren	543
30.18 Realisierung einer DSC-Ressource	544
30.19 Weitere Möglichkeiten	545
31 PowerShell-Snap-Ins	547
31.1 Einbinden von Snap-Ins	547
31.2 Liste der Commandlets	551
32 PowerShell-Module	553
32.1 Überblick über die Commandlets	553
32.2 Modulararchitektur	554
32.3 Module aus dem Netz herunterladen und installieren mit PowerShellGet	555
32.4 Module manuell installieren	562
32.5 Doppeldeutige Namen	562
32.6 Auflisten der verfügbaren Module	563
32.7 Importieren von Modulen	565
32.8 Entfernen von Modulen	568
33 Ausgewählte PowerShell-Erweiterungen	569
33.1 PowerShell-Module in Windows 7 und Windows Server 2008 R2	570
33.2 PowerShell-Module in Windows 8.0 und Windows Server 2012	571
33.3 PowerShell-Module in Windows 8.1 und Windows Server 2012 R2	573
33.4 PowerShell-Module in Windows 10 und Windows Server 2016	576
33.5 PowerShell Community Extensions (PSCX)	580
33.6 PowerShellPack	584
33.7 www.IT-Visions.de: PowerShell Extensions	585
33.8 Quest Management Shell for Active Directory	586
33.9 Microsoft Exchange Server	587
33.10 System Center Virtual Machine Manager	588
33.11 PowerShell Management Library for Hyper-V (pshyperv)	589
33.12 Powershell Outlook Account Manager	590
33.13 PowerShell Configurator (PSConfig)	591
33.14 Weitere Erweiterungen	592
34 Delegierte Administration/Just Enough Administration (JEA) ...	593
34.1 JEA-Konzept	593
34.2 PowerShell-Sitzungskonfiguration erstellen	593
34.3 Sitzungskonfiguration nutzen	597
34.4 Delegierte Administration per Webseite	598

35	Tipps und Tricks zur PowerShell	599
35.1	Alle Anzeigen löschen	599
35.2	Befehlsgeschichte	599
35.3	System- und Hostinformationen	600
35.4	Anpassen der Eingabeaufforderung (Prompt)	601
35.5	PowerShell-Befehle aus anderen Anwendungen heraus starten	602
35.6	ISE erweitern	603
35.7	PowerShell für Gruppenrichtlinienskripte	604
35.8	Einblicke in die Interna der Pipeline-Verarbeitung	606
Teil C: PowerShell im Praxiseinsatz		609
36	Dateisystem	611
36.1	Laufwerke	612
36.2	Ordnerinhalte	617
36.3	Dateieigenschaften verändern	619
36.4	Eigenschaften ausführbarer Dateien	620
36.5	Kurznamen	622
36.6	Lange Pfade	622
36.7	Dateisystemoperationen	623
36.8	Praxisbeispiel: Zufällige Dateisystemstruktur erzeugen	624
36.9	Praxisbeispiel: Leere Ordner löschen	625
36.10	Einsatz von Robocopy	626
36.11	Dateisystemkataloge	629
36.12	Papierkorb leeren	630
36.13	Dateieigenschaften lesen	630
36.14	Praxisbeispiel: Fotos nach Aufnahmedatum sortieren	631
36.15	Datei-Hash	632
36.16	Finden von Duplikaten	633
36.17	Verknüpfungen im Dateisystem	635
36.18	Komprimierung	640
36.19	Dateisystemfreigaben	642
36.20	Überwachung des Dateisystems	653
36.21	Dateiversionsverlauf	654
36.22	Windows Explorer öffnen	655
36.23	Windows Server Backup	655
37	Festplattenverschlüsselung mit BitLocker	659
37.1	Übersicht über das BitLocker-Modul	660
37.2	Verschlüsseln eines Laufwerks	661

38	Dokumente	663
38.1	Textdateien	663
38.2	CSV-Dateien	664
38.3	Analysieren von Textdateien	667
38.4	INI-Dateien	670
38.5	XML-Dateien	671
38.6	HTML-Dateien	679
38.7	Binärdateien	679
39	Datenbanken	681
39.1	ADO.NET-Grundlagen	681
39.2	Beispieldatenbank	687
39.3	Datenzugriff mit den Bordmitteln der PowerShell	688
39.4	Datenzugriff mit den PowerShell-Erweiterungen	699
39.5	Datenbankzugriff mit SQLPS	702
39.6	Datenbankzugriff mit SQLPSX	702
40	Microsoft-SQL-Server-Administration	703
40.1	PowerShell-Integration im SQL Server Management Studio	704
40.2	SQL-Server-Laufwerk „SQLSERVER:“	705
40.3	Die SQLPS-Commandlets	708
40.4	Die SQL Server Management Objects (SMO)	710
40.5	SQLPSX	713
40.6	Microsoft-SQL-Server-Administration mit der PowerShell in der Praxis	721
41	ODBC-Datenquellen	727
41.1	ODBC-Treiber und -Datenquellen auflisten	728
41.2	Anlegen einer ODBC-Datenquelle	729
41.3	Zugriff auf eine ODBC-Datenquelle	730
42	Registrierungsdatenbank (Registry)	733
42.1	Schlüssel auslesen	733
42.2	Schlüssel anlegen und löschen	734
42.3	Laufwerke definieren	734
42.4	Werte anlegen und löschen	735
42.5	Werte auslesen	736
42.6	Praxisbeispiel: Windows-Explorer-Einstellungen	736
42.7	Praxisbeispiel: Massenanlegen von Registry-Schlüsseln	737
43	Computer- und Betriebssystemverwaltung	739
43.1	Computerinformationen	739

43.2	Versionsnummer des Betriebssystems	741
43.3	Zeitdauer seit dem letzten Start des Betriebssystems	741
43.4	BIOS- und Startinformationen	742
43.5	Windows-Produktaktivierung	742
43.6	Umgebungsvariablen	742
43.7	Schriftarten	745
43.8	Computersname und Domäne	746
43.9	Herunterfahren und Neustarten	746
43.10	Windows Updates installieren	747
43.11	Wiederherstellungspunkte verwalten	751
44	Windows Defender	753
45	Hardwareverwaltung	755
45.1	Hardwarebausteine	755
45.2	Plug-and-Play-Geräte	757
45.3	Druckerverwaltung (ältere Betriebssysteme)	757
45.4	Druckerverwaltung (seit Windows 8 und Windows Server 2012)	758
46	Softwareverwaltung	761
46.1	Softwareinventarisierung	761
46.2	Installation von Anwendungen	764
46.3	Deinstallation von Anwendungen	765
46.4	Praxisbeispiel: Installationstest	765
46.5	Installationen mit PowerShell Package Management („OneGet“)	766
46.6	Versionsnummer ermitteln	769
46.7	Servermanager	770
46.8	Softwareeinschränkungen mit dem PowerShell-Modul „AppLocker“	781
47	Prozessverwaltung	787
47.1	Prozesse auflisten	787
47.2	Prozesse starten	788
47.3	Prozesse mit vollen Administratorrechten starten	789
47.4	Prozesse unter einem anderen Benutzerkonto starten	790
47.5	Prozesse beenden	791
47.6	Warten auf das Beenden einer Anwendung	792
48	Systemdienste	793
48.1	Dienste auflisten	793
48.2	Dienstzustand ändern	795
48.3	Diensteigenschaften ändern	796

49	Netzwerk	797
49.1	Netzwerkconfiguration (ältere Betriebssysteme)	797
49.2	Netzwerkconfiguration (ab Windows 8 und Windows Server 2012)	799
49.3	DNS-Client-Konfiguration	802
49.4	DNS-Namensauflösung	805
49.5	Erreichbarkeit prüfen (Ping)	807
49.6	Windows Firewall	808
49.7	Remote Desktop (RDP) einrichten	814
49.8	E-Mails senden (SMTP)	815
49.9	Auseinandernehmen von E-Mail-Adressen	817
49.10	Abruf von Daten von einem HTTP-Server	817
49.11	Praxisbeispiel: Linkprüfer für eine Website	819
49.12	Aufrufe von SOAP-Webdiensten	822
49.13	Aufruf von REST-Diensten	824
49.14	Aufrufe von OData-Diensten	826
49.15	Hintergrunddatentransfer mit BITS	827
50	Ereignisprotokolle (Event Log)	831
51	Leistungsdaten (Performance Counter)	835
51.1	Zugriff auf Leistungsindikatoren über WMI	835
51.2	Get-Counter	836
52	Sicherheitseinstellungen	839
52.1	Aktueller Benutzer	839
52.2	Grundlagen	840
52.3	Zugriffsrechtelisten auslesen	845
52.4	Einzelne Rechteinträge auslesen	846
52.5	Besitzer auslesen	848
52.6	Benutzer und SID	848
52.7	Hinzufügen eines Rechteintrags zu einer Zugriffsrechteliste	851
52.8	Entfernen eines Rechteintrags aus einer Zugriffsrechteliste	854
52.9	Zugriffsrechteliste übertragen	855
52.10	Zugriffsrechteliste über SDDL setzen	856
52.11	Zertifikate verwalten	857
53	Optimierungen und Problemlösungen	861
53.1	PowerShell-Modul „TroubleshootingPack“	861
53.2	PowerShell-Modul „Best Practices“	865

54	Active Directory	867
54.1	Benutzer- und Gruppenverwaltung mit WMI	868
54.2	Einführung in System.DirectoryServices	869
54.3	Basiseigenschaften	880
54.4	Benutzer- und Gruppenverwaltung im Active Directory	882
54.5	Verwaltung der Organisationseinheiten	890
54.6	Suche im Active Directory	891
54.7	Navigation im Active Directory mit den PowerShell Extensions	898
54.8	Verwendung der Active-Directory-Erweiterungen von www.IT-Visions.de	899
54.9	PowerShell-Modul „Active Directory“ (ADPowerShell)	901
54.10	PowerShell-Modul „ADDSDeployment“	924
54.11	Informationen über die Active-Directory-Struktur	926
55	Gruppenrichtlinien	929
55.1	Verwaltung der Gruppenrichtlinien	930
55.2	Verknüpfung der Gruppenrichtlinien	931
55.3	Gruppenrichtlinienberichte	933
55.4	Gruppenrichtlinienvererbung	934
55.5	Weitere Möglichkeiten	935
56	Lokale Benutzer und Gruppen	937
56.1	Modul „Microsoft.PowerShell.LocalAccounts“	937
56.2	Lokale Benutzerverwaltung in älteren PowerShell-Versionen	939
57	Microsoft Exchange Server	941
57.1	Daten abrufen	941
57.2	Postfächer verwalten	942
57.3	Öffentliche Ordner verwalten	943
58	Internet Information Server (IIS)	945
58.1	Überblick	945
58.2	Navigationsprovider	947
58.3	Anlegen von Websites	949
58.4	Praxisbeispiel: Massenanlegen von Websites	950
58.5	Ändern von Website-Eigenschaften	953
58.6	Anwendungspool anlegen	953
58.7	Virtuelle Verzeichnisse und IIS-Anwendungen	954
58.8	Website-Zustand ändern	955
58.9	Anwendungspools starten und stoppen	955
58.10	Löschen von Websites	956

59	Virtuelle Systeme mit Hyper-V	957
59.1	Das Hyper-V-Modul von Microsoft	958
59.2	Die ersten Schritte mit dem Hyper-V-Modul	960
59.3	Virtuelle Maschinen anlegen	964
59.4	Umgang mit virtuellen Festplatten	970
59.5	Konfiguration virtueller Maschinen	973
59.6	Dateien kopieren in virtuelle Systeme	977
59.7	PowerShell Management Library for Hyper-V (für ältere Betriebssysteme)	979
60	Windows Nano Server	983
60.1	Das Konzept von Nano Server	983
60.2	Einschränkungen von Nano Server	985
60.3	Varianten des Nano Servers	987
60.4	Installation eines Nano Servers	987
60.5	Docker-Image	989
60.6	Fernverwaltung mit PowerShell	989
60.7	Windows Update auf einem Nano Server	992
60.8	Nachträgliche Paketinstallation	992
60.9	Abgespeckter IIS unter Nano Server	994
60.10	Nano-Serververwaltung aus der Cloud heraus	995
61	Docker-Container	997
61.1	Docker-Varianten für Windows	998
61.2	Docker-Installation auf Windows 10	999
61.3	Docker-Installation auf Windows Server 2016	1001
61.4	Installation von „Docker for Windows“	1002
61.5	Docker-Registries	1004
61.6	Docker-Images laden	1004
61.7	Container starten	1005
61.8	Container-Identifikation	1006
61.9	Container mit Visual Studio	1007
61.10	Befehle in einem Container ausführen	1009
61.11	Ressourcenbeschränkungen für Container	1011
61.12	Dateien zwischen Container und Host kopieren	1011
61.13	Dockerfile	1011
61.14	Docker-Netzwerke	1012
61.15	Container anlegen, ohne sie zu starten	1013
61.16	Container starten und stoppen	1013
61.17	Container beenden und löschen	1013
61.18	Images löschen	1014

61.19	Images aus Containern erstellen	1014
61.20	.NET Core-Container	1014
61.21	Images verbreiten	1017
61.22	Azure Container Service (ACS)	1019
62	Grafische Benutzeroberflächen (GUI)	1021
62.1	Einfache Nachfragedialoge	1021
62.2	Einfache Eingabe mit Inputbox	1023
62.3	Komplexere Eingabemasken	1024
62.4	Universelle Objektdarstellung	1026
62.5	WPF PowerShell Kit (WPK)	1027
62.6	Direkte Verwendung von WPF	1035
Teil D: Profiwissen – Erweitern der PowerShell		1037
63	Entwicklung von Commandlets in der PowerShell- Skriptsprache	1039
63.1	Aufbau eines skriptbasierten Commandlets	1039
63.2	Verwendung per Dot Sourcing	1041
63.3	Parameterfestlegung	1042
63.4	Fortgeschrittene Funktion (Advanced Function)	1048
63.5	Mehrere Parameter und Parametersätze	1050
63.6	Unterstützung für Sicherheitsabfragen (-whatif und -confirm)	1052
63.7	Kaufmännisches Beispiel: Test-CustomerID	1054
63.8	Erweitern bestehender Commandlets durch Proxy-Commandlets	1057
63.9	Dokumentation	1063
64	Entwicklung eigener Commandlets mit C#	1067
64.1	Technische Voraussetzungen	1068
64.2	Grundkonzept der .NET-basierten Commandlets	1069
64.3	Schrittweise Erstellung eines minimalen Commandlets	1071
64.4	Erstellung eines Commandlets mit einem Rückgabeobjekt	1079
64.5	Erstellung eines Commandlets mit mehreren Rückgabeobjekten	1081
64.6	Erstellen eines Commandlets mit Parametern	1085
64.7	Verarbeiten von Pipeline-Eingaben	1087
64.8	Verkettung von Commandlets	1090
64.9	Fehlersuche in Commandlets	1094
64.10	Statusinformationen	1097
64.11	Unterstützung für Sicherheitsabfragen (-whatif und -confirm)	1102
64.12	Festlegung der Hilfeinformationen	1104
64.13	Erstellung von Commandlets für den Zugriff auf eine Geschäftsanwendung ...	1108

64.14 Konventionen für Commandlets	1109
64.15 Weitere Möglichkeiten	1111
65 PowerShell-Module erstellen	1113
65.1 Erstellen eines Skriptmoduls	1113
65.2 Praxisbeispiel: Umwandlung einer Skriptdatei in ein Modul	1115
65.3 Erstellen eines Moduls mit Binärdateien	1115
65.4 Erstellen eines Moduls mit Manifest	1116
65.5 Erstellung eines Manifest-Moduls mit Visual Studio	1123
66 Hosting der PowerShell	1125
66.1 Voraussetzungen für das Hosting	1126
66.2 Hosting mit PSHost	1127
66.3 Vereinfachtes Hosting seit PowerShell 2.0	1130
Anhang A: Crashkurs „Objektorientierung“	1133
Anhang B: Crashkurs .NET	1141
B.1 Was ist das .NET Framework?	1143
B.2 Was ist .NET Core?	1144
B.3 Eigenschaften von .NET	1145
B.4 .NET-Klassen	1146
B.5 Namensgebung von .NET-Klassen (Namensräume)	1146
B.6 Namensräume und Softwarekomponenten	1148
B.7 Bestandteile einer .NET-Klasse	1149
B.8 Vererbung	1150
B.9 Schnittstellen	1151
Anhang C: Literatur	1152
Anhang D: Weitere Informationen im Internet	1155
Anhang E: Abkürzungsverzeichnis	1157
Stichwortverzeichnis	1183

Vorwort zur sechsten Auflage

Liebe Leserin, lieber Leser,

willkommen zur fünften Auflage dieses PowerShell-Buchs! Das vor Ihnen liegende Buch behandelt die Windows PowerShell in der Version 5.1 sowie die PowerShell Core in der Version 6.0 (zum Redaktionsschluss dieses Buchs noch Beta-Version) 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 44 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 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) 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 67. 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 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 alle 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.

■ 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 Einflusstströmen steht natürlich auch dieses etablierte Buch. Leider kann man ein gedrucktes Buch nicht so schnell ändern wie Software. Verlage definieren erhebliche Mindestauflagen, die abverkauft werden müssen, bevor neu gedruckt werden darf. Das E-Book ist keine Alternative. Die Verkaufszahlen zeigen, dass nur eine verschwindend kleine Menge von Lesern technischer Literatur ein E-Book statt eines gedruckten Buchs kauft. Das E-Book wird offenbar nur gerne als Ergänzung genommen. Das kann ich gut verstehen, denn ich selbst lese auch lieber gedruckte Bücher und nutze E-Books nur für eine Volltextsuche.

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     PowerShellOOP
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 stehen mein Team und ich für Anfragen bezüglich Schulung, Beratung und Entwicklungstätigkeiten zur Verfügung – nicht nur zum Thema PowerShell und .NET, sondern zu fast allen modernen Techniken der Entwicklung und des Betriebs von Software. 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 Juni 2017

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*


Dr. Holger Schwichtenberg

- 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