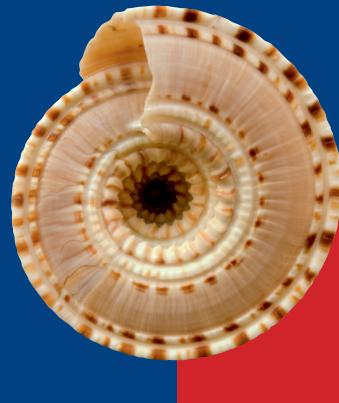


holger SCHWICHTENBERG

WINDOWS PowerShell 5.0



DAS PRAXISBUCH

HANSER



Im Internet: [Codebeispiele](#), [Forum](#),
[PowerShell-Kurzreferenz](#)

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 5.0

Das Praxisbuch

HANSER

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. Autoren 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 Autoren 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 dürfen.



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.

© 2016 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sylvia Hasselbach

Copy editing: Petra Kienle, Fürstenfeldbruck

Herstellung: Irene Weilhart

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: 978-3-446-44643-4

E-Book-ISBN: 978-3-446-44815-5

Inhalt

Vorwort zur fünften Auflage Januar 2016	XIX
Wer bin ich?	XIX
Wer sind Sie?	XX
Was ist neu in diesem Buch?	XX
Sind in diesem Buch alle Features der PowerShell beschrieben?	XXI
Wem ist zu danken?	XXI
Woher bekommen Sie die Beispiele aus diesem Buch?	XXI
Wo kann ich mich schulen oder beraten lassen?	XXII
Zum Schluss des Vorworts	XXII
Über den Autor Dr. Holger Schwichtenberg	XXIII
Über den Co-Autor Peter Monadjemi	XXIV
Teil A: PowerShell-Basiswissen	1
1 Erste Schritte mit der Windows PowerShell	3
1.1 Was ist die Windows PowerShell?	3
1.2 Geschichte der Windows PowerShell	4
1.3 Motivation zur Windows PowerShell	5
1.4 Betriebssysteme mit vorinstallierter PowerShell	8
1.5 PowerShell und Linux	9
1.6 PowerShell herunterladen und auf anderen Betriebssystemen installieren	10
1.7 Die Windows PowerShell testen	13
1.8 Woher kommen die Commandlets?	22
1.9 PowerShell Community Extensions (PSCX) herunterladen und installieren ...	22
1.10 Den PowerShell-Editor „ISE“ verwenden	24
2 Architektur der Windows PowerShell	29

3	Einzelbefehle der PowerShell	33
3.1	Commandlets	33
3.2	Aliase	42
3.3	Ausdrücke	49
3.4	Externe Befehle	50
3.5	Dateinamen	51
4	Hilfefunktionen	53
4.1	Auflisten der verfügbaren Befehle	53
4.2	Erläuterungen zu den Befehlen	56
4.3	Hilfe zu Parametern	57
4.4	Hilfe mit Show-Command	57
4.5	Hilfefenster	58
4.6	Aktualisieren der Hilfedateien	60
4.7	Online-Hilfe	61
4.8	Dokumentation der .NET-Klassen	62
5	Objektorientiertes Pipelining	65
5.1	Pipeline-Operator	65
5.2	.NET-Objekte in der Pipeline	66
5.3	Pipeline Processor	68
5.4	Pipelining von Parametern	69
5.5	Pipelining von klassischen Befehlen	71
5.6	Anzahl der Objekte in der Pipeline	72
5.7	Zugriff auf einzelne Objekte aus einer Menge	72
5.8	Zugriff auf einzelne Werte in einem Objekt	74
5.9	Methoden ausführen	75
5.10	Analyse des Pipeline-Inhalts	77
5.11	Filtern	88
5.12	Zusammenfassung von Pipeline-Inhalten	91
5.13	„Kastrierung“ von Objekten in der Pipeline	92
5.14	Sortieren	93
5.15	Duplikate entfernen	93
5.16	Gruppierung	94
5.17	Berechnungen	96
5.18	Zwischenschritte in der Pipeline mit Variablen	96
5.19	Verzweigungen in der Pipeline	97
5.20	Vergleiche zwischen Objekten	99
5.21	Zusammenfassung	100
5.22	Praxisbeispiele	100

6	PowerShell-Skripte	103
6.1	Skriptdateien	103
6.2	Start eines Skripts	105
6.3	Aliase für Skripte verwenden	106
6.4	Parameter für Skripte	107
6.5	Skripte dauerhaft einbinden (Dot Sourcing)	108
6.6	Sicherheitsfunktionen für PowerShell-Skripte	109
6.7	Skripte anhalten	113
7	PowerShell-Skriptsprache	115
7.1	Hilfe zur PowerShell-Skriptsprache	115
7.2	Befehlstrennung	116
7.3	Kommentare	116
7.4	Variablen	117
7.5	Variablenbedingungen	124
7.6	Zahlen	125
7.7	Zeichenketten	127
7.8	Reguläre Ausdrücke	133
7.9	Datum und Uhrzeit	140
7.10	Arrays	141
7.11	Assoziative Arrays (Hash-Tabellen)	143
7.12	Operatoren	145
7.13	Überblick über die Kontrollkonstrukte	146
7.14	Schleifen	147
7.15	Bedingungen	152
7.16	Unterroutinen (Prozedur/Funktionen)	154
7.17	Eingebaute Funktionen	159
7.18	Fehlerbehandlung	160
7.19	Objektorientiertes Programmieren mit Klassen	167
8	Ausgaben	171
8.1	Ausgabe-Commandlets	171
8.2	Out-GridView	174
8.3	Standardausgabe	175
8.4	Einschränkung der Ausgabe	178
8.5	Seitenweise Ausgabe	178
8.6	Ausgabe einzelner Werte	179
8.7	Details zum Ausgabeoperator	181
8.8	Benutzerdefinierte Tabellenformatierung	184
8.9	Ausgabe von Methodenergebnissen und Unterobjekten in Pipelines	185

8.10	Ausgabe von Methodenergebnissen und Unterobjekten in Zeichenketten	186
8.11	Unterdrückung der Ausgabe	186
8.12	Ausgaben an Drucker	187
8.13	Ausgaben in Dateien	187
8.14	Umleitungen (Redirection)	188
8.15	Sprachausgabe	189
9	Benutzereingaben	191
9.1	Read-Host	191
9.2	Grafischer Eingabedialog	192
9.3	Dialogfenster	192
9.4	Authentifizierungsdialog	193
10	Das PowerShell-Navigationsmodell	195
10.1	Einführungsbeispiel: Navigation in der Registrierungsdatenbank	195
10.2	Provider und Laufwerke	196
10.3	Navigationsbefehle	198
10.4	Pfadangaben	199
10.5	Beispiel	201
10.6	Eigene Laufwerke definieren	202
11	PowerShell-Werkzeuge	203
11.1	PowerShell-Standardkonsole	203
11.2	PowerShell Integrated Scripting Environment (ISE)	210
11.3	PowerShell Script Analyzer	216
11.4	CIM Explorer for PowerShell ISE	222
11.5	PowerShell Web Access (PSWA)	223
11.6	ISE Steroids	229
11.7	PowerShellPlus	230
11.8	PoshConsole	233
11.9	PowerGUI	234
11.10	PowerShell Analyzer	235
11.11	PrimalScript	236
11.12	PowerShell Help	238
11.13	PowerShell Help Reader	238
11.14	PowerTab	239
11.15	NuGet Package Manager	239
11.16	PowerShell Tools for Visual Studio	240
11.17	PowerShell Remoting	241
11.18	Vergleich der Skripteditoren	241

Teil B: PowerShell-Aufbauwissen	243
12 Ferausführung (Remoting)	245
12.1 RPC-Fernabfrage ohne WS-Management	246
12.2 Anforderungen an PowerShell Remoting	247
12.3 Rechte für PowerShell-Remoting	248
12.4 Einrichten von PowerShell Remoting	249
12.5 Überblick über die Ferausführungs-Commandlets	251
12.6 Interaktive Fernverbindungen im Telnet-Stil	252
12.7 Ferausführung von Befehlen	253
12.8 Parameterübergabe an die Ferausführung	257
12.9 Ferausführung von Skripten	258
12.10 Ausführung auf mehreren Computern	259
12.11 Sitzungen	260
12.12 Implizites Remoting	264
12.13 Zugriff auf entfernte Computer außerhalb der eigenen Domäne	265
12.14 Verwaltung des WS-Management-Dienstes	268
12.15 PowerShell Direct für HyperV	270
12.16 Praxisbeispiel zu PowerShell Direct	272
13 Verwendung von .NET-Klassen	275
13.1 Microsoft Developer Network (MSDN)	275
13.2 Erzeugen von Instanzen	276
13.3 Parameterbehaftete Konstruktoren	278
13.4 Initialisierung von Objekten	279
13.5 Nutzung von Attributen und Methoden	280
13.6 Statische Mitglieder in .NET-Klassen und statische .NET-Klassen	282
13.7 Generische Klassen nutzen	284
13.8 Zugriff auf bestehende Objekte	285
13.9 Laden von Assemblies	286
13.10 Objektanalyse	287
13.11 Auflistungen (Enumerationen)	288
13.12 Verknüpfen von Aufzählungswerten	289
14 Verwendung von COM-Klassen	291
14.1 Erzeugen von Instanzen	291
14.2 Nutzung von Attributen und Methoden	292
14.3 Holen bestehender Instanzen	293
15 Zugriff auf die Windows Management Instrumentation (WMI)	295
15.1 Einführung in WMI	295

15.2	WMI in der Windows PowerShell	321
15.3	Abruf von WMI-Objektmengen	322
15.4	Fernzugriffe	323
15.5	Filtern und Abfragen	323
15.6	Liste aller WMI-Klassen	328
15.7	Hintergrundwissen: WMI-Klassenprojektion mit dem PowerShell-WMI-Objektadapter	328
15.8	Beschränkung der Ausgabeliste bei WMI-Objekten	332
15.9	Zugriff auf einzelne Mitglieder von WMI-Klassen	333
15.10	Werte setzen in WMI-Objekten	334
15.11	Umgang mit WMI-Datumsangaben	336
15.12	Methodenaufrufe	337
15.13	Neue WMI-Instanzen erzeugen	338
15.14	Instanzen entfernen	339
15.15	Commandlet Definition XML-Datei (CDXML)	340
16	Dynamische Objekte	343
16.1	Erweitern bestehender Objekte	343
16.2	Komplett dynamische Objekte	345
17	Einbinden von C# und VB.NET	347
18	Win32-API-Aufrufe	349
19	Fehlersuche	353
19.1	Detailinformationen	353
19.2	Einzelschrittmodus	354
19.3	Zeitmessung	355
19.4	Ablaufverfolgung	356
19.5	Kommandozeilenbasiertes Script-Debugging	356
20	Transaktionen	359
20.1	Commandlets für Transaktionen	359
20.2	Start und Ende einer Transaktion	360
20.3	Zurücksetzen der Transaktion	361
20.4	Mehrere Transaktionen	362
21	Hintergrundaufträge („Jobs“)	363
21.1	Voraussetzungen	363
21.2	Architektur	364
21.3	Starten eines Hintergrundauftrags	364

21.4	Hintergrundaufträge abfragen	365
21.5	Warten auf einen Hintergrundauftrag	366
21.6	Abbrechen und Löschen von Aufträgen	366
21.7	Analyse von Fehlermeldungen	367
21.8	Fernausführung von Hintergrundaufträgen	367
21.9	Praxisbeispiel	367
22	Geplante Aufgaben und zeitgesteuerte Jobs	369
22.1	Geplante Aufgaben (Scheduled Tasks)	369
22.2	Zeitgesteuerte Jobs	373
23	PowerShell-Workflows	379
23.1	Ein erstes Beispiel	379
23.2	Unterschiede zu einer Function bzw. einem Skript	384
23.3	Einschränkungen bei Workflows	384
23.4	Workflows in der Praxis	386
23.5	Workflows in Visual Studio erstellen	393
24	Ereignissystem	411
24.1	WMI-Ereignisse	411
24.2	WMI-Ereignisabfragen	411
24.3	WMI-Ereignisse seit PowerShell 1.0	413
24.4	Registrieren von WMI-Ereignisquellen seit PowerShell 2.0	414
24.5	Auslesen der Ereignisliste	415
24.6	Reagieren auf Ereignisse	417
24.7	WMI-Ereignisse ab PowerShell-Version 3.0	419
24.8	Registrieren von .NET-Ereignissen	419
24.9	Erzeugen von Ereignissen	420
25	Datenbereiche und Datendateien	423
25.1	Datenbereiche	423
25.2	Datendateien	425
25.3	Mehrsprachigkeit/Lokalisierung	426
26	Desired State Configuration (DSC)	429
26.1	Grundprinzipien	430
26.2	DSC für Linux	430
26.3	Ressourcen	431
26.4	Verfügbare DSC-Ressourcen	431
26.5	Eigenschaften einer Ressource	434
26.6	Aufbau eines DSC-Dokuments	434

26.7	Commandlets für die Arbeit mit DSC	435
26.8	Ein erstes DSC-Beispiel	435
26.9	Kompilieren und Anwendung eines DSC-Dokuments	436
26.10	Variablen in DSC-Dateien	438
26.11	Parameter für DSC-Dateien	439
26.12	Konfigurationsdaten	440
26.13	Entfernen einer DSC-Konfiguration	443
26.14	DSC Pull Server	446
26.15	DSC-Praxisbeispiel 1: IIS installieren	453
26.16	DSC-Praxisbeispiel 2: Software installieren	454
26.17	DSC-Praxisbeispiel 3: Software deinstallieren	457
26.18	Realisierung einer DSC-Ressource	457
26.19	Weitere Möglichkeiten	458
27	PowerShell-Snap-Ins	459
27.1	Einbinden von Snap-Ins	459
27.2	Liste der Commandlets	463
27.3	Doppeldeutige Namen	464
28	PowerShell-Module	465
28.1	Überblick über die Commandlets	465
28.2	Modulararchitektur	466
28.3	Module aus dem Netz herunterladen und installieren mit PowerShellGet	467
28.4	Module manuell installieren	472
28.5	Auflisten der verfügbaren Module	473
28.6	Importieren von Modulen	474
28.7	Entfernen von Modulen	477
29	Ausgewählte PowerShell-Erweiterungen	479
29.1	PowerShell-Module in Windows 7 und Windows Server 2008 R2	479
29.2	PowerShell-Module in Windows 8.0 und Windows Server 2012	481
29.3	PowerShell-Module in Windows 8.1 und Windows Server 2012 R2	483
29.4	PowerShell-Module in Windows 10 und Windows Server 2016	485
29.5	Windows PowerShell Community Extensions (PSCX)	490
29.6	PowerShellPack	493
29.7	<i>www.IT-Visions.de</i> PowerShell Extensions	495
29.8	Quest Management Shell for Active Directory	496
29.9	Microsoft Exchange Server	497
29.10	System Center Virtual Machine Manager	498
29.11	PowerShell Management Library for Hyper-V (pshyperv)	499
29.12	Powershell Outlook Account Manager	500

29.13 PowerShell Configurator (PSConfig)	500
29.14 Weitere Erweiterungen	501
30 Tipps und Tricks zur PowerShell	503
30.1 Befehlsgeschichte	503
30.2 System- und Hostinformationen	504
30.3 Alle Anzeigen löschen	505
30.4 Anpassen der Eingabeaufforderung (Prompt)	505
30.5 Standardeinstellungen ändern mit Profilskripten	506
30.6 ISE erweitern	511
30.7 PowerShell für Gruppenrichtlinienskripte	512
30.8 Einblicke in die Interna der Pipeline-Verarbeitung	514
Teil C: PowerShell im Praxiseinsatz	517
31 Dateisystem	519
31.1 Laufwerke	520
31.2 Ordnerinhalte	525
31.3 Kurznamen	526
31.4 Dateisystemoperationen	527
31.5 Praxisbeispiel: Leere Ordner löschen	528
31.6 Papierkorb leeren	529
31.7 Dateieigenschaften lesen	529
31.8 Praxisbeispiel: Fotos nach Aufnahmedatum sortieren	530
31.9 Datei-Hash	531
31.10 Finden von Duplikaten	532
31.11 Dateieigenschaften verändern	534
31.12 Verknüpfungen im Dateisystem	536
31.13 Komprimierung	541
31.14 Dateisystemfreigaben	544
31.15 Überwachung des Dateisystems	555
31.16 Dateiversionsverlauf	556
31.17 Windows Server Backup	557
32 Festplattenverschlüsselung mit BitLocker	559
32.1 Übersicht über das BitLocker-Modul	560
32.2 Verschlüsseln eines Laufwerks	561
33 Dokumente	563
33.1 Textdateien	563
33.2 CSV-Dateien	564

33.3	Analysieren von Textdateien	567
33.4	INI-Dateien	570
33.5	XML-Dateien	570
33.6	HTML-Dateien	579
33.7	Binärdateien	579
34	Datenbanken	581
34.1	ADO.NET-Grundlagen	581
34.2	Beispieldatenbank	587
34.3	Datenzugriff mit den Bordmitteln der PowerShell	588
34.4	Datenzugriff mit den PowerShell-Erweiterungen	599
34.5	Datenbankzugriff mit SQLPS	602
34.6	Datenbankzugriff mit SQLPSX	602
35	Microsoft-SQL-Server-Administration	603
35.1	PowerShell-Integration im SQL Server Management Studio	604
35.2	SQL-Server-Laufwerk „SQLSERVER:“	605
35.3	Die SQLPS-Commandlets	608
35.4	Die SQL Server Management Objects (SMO)	610
35.5	SQLPSX	613
35.6	Microsoft-SQL-Server-Administration mit der PowerShell in der Praxis	621
36	ODBC-Datenquellen	627
36.1	ODBC-Treiber und -Datenquellen auflisten	628
36.2	Anlegen einer ODBC-Datenquelle	629
36.3	Zugriff auf eine ODBC-Datenquelle	630
37	Registrierungsdatenbank (Registry)	633
37.1	Schlüssel auslesen	633
37.2	Schlüssel anlegen und löschen	634
37.3	Laufwerke definieren	634
37.4	Werte anlegen und löschen	635
37.5	Werte auslesen	636
37.6	Praxisbeispiel: Windows-Explorer-Einstellungen	636
37.7	Praxisbeispiel: Massenanlegen von Registry-Schlüsseln	637
38	Computerverwaltung	639
38.1	Computerinformationen	639
38.2	Computername und Domäne	641
38.3	Herunterfahren und Neustarten	642
38.4	Wiederherstellungspunkte verwalten	643

39 Hardwareverwaltung	645
39.1 Hardwarebausteine	645
39.2 Plug-and-Play-Geräte	647
39.3 Druckerverwaltung (ältere Betriebssysteme)	647
39.4 Druckerverwaltung (seit Windows 8 und Windows Server 2012)	648
40 Softwareverwaltung	651
40.1 Softwareinventarisierung	651
40.2 Installation von Anwendungen	654
40.3 Deinstallation von Anwendungen	655
40.4 Praxisbeispiel: Installationstest	655
40.5 Installationen mit PowerShell Package Management („OneGet“)	656
40.6 Versionsnummer ermitteln	659
40.7 Servermanager	660
40.8 Softwareeinschränkungen mit dem PowerShell-Modul „AppLocker“	671
41 Prozessverwaltung	677
41.1 Prozesse auflisten	677
41.2 Prozesse starten	678
41.3 Prozesse beenden	680
41.4 Warten auf das Beenden einer Anwendung	680
42 Systemdienste	681
42.1 Dienste auflisten	681
42.2 Dienstzustand ändern	683
42.3 Diensteigenschaften ändern	684
43 Netzwerk	685
43.1 Netzwerkkonfiguration (ältere Betriebssysteme)	685
43.2 Netzwerkkonfiguration (ab Windows 8 und Windows Server 2012)	687
43.3 DNS-Client-Konfiguration	690
43.4 DNS-Namensauflösung	693
43.5 Erreichbarkeit prüfen (Ping)	695
43.6 Windows Firewall	696
43.7 Remote Desktop (RDP) einrichten	702
43.8 E-Mails senden (SMTP)	703
43.9 Abruf von Daten von einem HTTP-Server	705
43.10 Aufrufe von SOAP-Webdiensten	706
43.11 Aufrufe von OData-Diensten	709
43.12 BITSTransfer	709

44 Ereignisprotokolle	711
14.1 Protokolleinträge auslesen	711
14.2 Ereignisprotokolle erzeugen	713
14.3 Protokolleinträge erzeugen	713
14.4 Protokollgröße festlegen	713
14.5 Protokolleinträge löschen	713
45 Leistungsdaten	715
45.1 Zugriff auf Leistungsindikatoren über WMI	715
45.2 Get-Counter	716
46 Sicherheitseinstellungen	719
46.1 Grundlagen	719
46.2 Zugriffsrechitelisten auslesen	724
46.3 Einzelne Rechteeinträge auslesen	726
46.4 Besitzer auslesen	727
46.5 Benutzer und SID	727
46.6 Hinzufügen eines Rechteeintrags zu einer Zugriffsrechtleiste	731
46.7 Entfernen eines Rechteintrags aus einer Zugriffsrechtleiste	733
46.8 Zugriffsrechtleiste übertragen	735
46.9 Zugriffsrechtleiste über SDDL setzen	735
47 Active Directory	737
47.1 Benutzer- und Gruppenverwaltung mit WMI	738
47.2 Einführung in System.DirectoryServices	739
47.3 Basiseigenschaften	750
47.4 Benutzer- und Gruppenverwaltung im Active Directory	752
47.5 Verwaltung der Organisationseinheiten	760
47.6 Suche im Active Directory	761
47.7 Navigation im Active Directory mit den PowerShell Extensions	768
47.8 Verwendung der Active-Directory-Erweiterungen von <i>www.IT-Visions.de</i>	769
47.9 PowerShell-Modul „Active Directory“ (ADPowerShell)	771
47.10 PowerShell-Modul „ADDSDeployment“	794
47.11 Informationen über die Active-Directory-Struktur	796
48 Gruppenrichtlinien	799
48.1 Verwaltung der Gruppenrichtlinien	800
48.2 Verknüpfung der Gruppenrichtlinien	801
48.3 Gruppenrichtlinienberichte	803
48.4 Gruppenrichtlinienvererbung	804
48.5 Weitere Möglichkeiten	805

49 Virtuelle Systeme mit Hyper-V	807
49.1 Das Hyper-V-Modul von Microsoft	808
49.2 Die ersten Schritte mit dem Hyper-V-Modul	810
49.3 Virtuelle Maschinen anlegen	814
49.4 Umgang mit virtuellen Festplatten	820
49.5 Konfiguration virtueller Maschinen	823
49.6 Dateien kopieren in virtuelle Systeme	827
49.7 PowerShell Management Library for Hyper-V (für ältere Betriebssysteme)	828
50 Internet Information Server (IIS)	833
50.1 Überblick	833
50.2 Navigationsprovider	835
50.3 Anlegen von Websites	837
50.4 Praxisbeispiel: Massenanlegen von Websites	838
50.5 Ändern von Eigenschaften von Websites	841
50.6 Anwendungspool anlegen	841
50.7 Virtuelle Verzeichnisse und IIS-Anwendungen	842
50.8 Website-Zustand ändern	843
50.9 Anwendungspools starten und stoppen	843
50.10 Löschen von Websites	844
51 Microsoft Exchange Server	845
51.1 Daten abrufen	845
51.2 Postfächer verwalten	846
51.3 Öffentliche Ordner verwalten	847
52 Optimierungen und Problemlösungen	849
52.1 PowerShell-Modul „TroubleshootingPack“	849
52.2 PowerShell-Modul „Best Practices“	853
53 Grafische Benutzeroberflächen	855
53.1 Eingabemasken	855
53.2 Universelle Objektdarstellung	857
53.3 Zwischenablage	859
53.4 WPF PowerShell Kit (WPK)	860
Teil D: Profiwissen – Erweitern der PowerShell	871
54 Entwicklung von Commandlets in der PowerShell-Sprache	873
54.1 Aufbau eines skriptbasierten Commandlets	873
54.2 Parameterfestlegung	876

54.3 Auszeichnung der Parameterdefinitionen	881
54.4 Dokumentation	884
55 Entwicklung eigener Commandlets mit C#	887
55.1 Technische Voraussetzungen	887
55.2 Grundkonzept der .NET-basierten Commandlets	889
55.3 Schrittweise Erstellung eines minimalen Commandlets	890
55.4 Erstellung eines Commandlets mit einem Rückgabeobjekt	898
55.5 Erstellung eines Commandlets mit mehreren Rückgabeobjekten	900
55.6 Erstellen eines Commandlets mit Parametern	904
55.7 Verarbeiten von Pipeline-Eingaben	906
55.8 Verkettung von Commandlets	909
55.9 Fehlersuche in Commandlets	913
55.10 Statusinformationen	916
55.11 Unterstützung für Sicherheitsabfragen	921
55.12 Festlegung der Hilfeinformationen	923
55.13 Erstellung von Commandlets für den Zugriff auf eine Geschäftsanwendung ...	927
55.14 Konventionen für Commandlets	928
55.15 Weitere Möglichkeiten	930
56 Hosting der Windows PowerShell	931
56.1 Voraussetzungen für das Hosting	932
56.2 Hosting mit PSHost	933
56.3 Vereinfachtes Hosting seit PowerShell 2.0	936
57 PowerShell-Module erstellen	939
57.1 Erstellen eines Skriptmoduls	939
57.2 Erstellen eines Moduls mit Binärdateien	941
57.3 Erstellen eines Moduls mit Manifest	941
Anhang	949
A Crashkurs „Objektorientierung“	951
B Crashkurs „.NET Framework“	959
C Literatur	969
D Weitere Informationen im Internet	973
Stichwortverzeichnis	975

Vorwort zur fünften Auflage

Liebe Leserin, lieber Leser,

willkommen zur fünften Auflage dieses PowerShell-Buchs! Das vor Ihnen liegende Buch behandelt die Windows-PowerShell-Version 5.0 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 oder 4.0 einsetzen. Welche Funktionen neu hinzugekommen sind, wird jeweils erwähnt.

■ Wer bin ich?

Mein Name ist Holger Schwichtenberg, ich bin derzeit 43 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- 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 Unternehmen.

Es ist mein Hobby und „Nebenberuf“, IT-Fachbücher zu schreiben. Dieses Buch ist, unter Mitzählung aller nennenswerten Neuauflagen, das 63. Buch, das ich allein oder mit Co-Autoren geschrieben habe. Meine weiteren Hobbys sind Mountain Biking, Lauf-Sport, Fotografie und Reisen.

Natürlich versteh ich das Bücherschreiben auch als Werbung für die Arbeit unserer Unternehmen und wir hoffen, dass der ein oder andere von Ihnen uns beauftragen wird, Sie 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 oder 5.0 unterstützten Betriebssystemen? (Windows 7/8/8.1/10, Windows Server 2008/2008 R2/2012/2012 R2/2016)	<input type="radio"/> Ja <input type="radio"/> Nein
Sie besitzen zumindest rudimentäre Grundkenntnisse im Bereich des (objekt-orientierten) 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 1000 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 4.0 wurde das Buch um die neuen Funktionen in PowerShell 5.0 erweitert und inhaltlich optimiert. Zudem wurden die bestehenden Inhalte des Buchs an einigen Stellen erweitert und ich habe Praxisbeispiele aus meinen Arbeiten für Kunden und an unserer eigenen IT-Infrastruktur aus den letzten Jahren hinzugefügt.

■ Sind in diesem Buch alle Features der PowerShell beschrieben?

Die PowerShell umfasst mittlerweile über 1400 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 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 jemals eingesetzt. Ich beschreibe in diesem Buch, was ich selbst in der Praxis verwende.

■ 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-Auflage dieses Buchs beigetragen hat (Themen: Workflows, Bitlocker, ODBC, Hyper-V, DNS-Client, Firewall und SQL-Server-Administration),
- Frau Bauer-Schiewek und Frau Hasselbach, die dieses Buchprojekt beim Carl Hanser Verlag koordinieren und vermarkten,
- Frau Petra Kienle, 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 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 **The Man in the High Castle** angeben.

■ Wo kann ich mich schulen oder beraten lassen?

Unter der E-Mail-Adresse *buero@ITVisions.de* stehen mein Team und ich für Anfragen bezüglich Schulung, Beratung und Entwicklungstätigkeiten zur Verfügung.

■ Zum Schluss des Vorworts ...

wünsche ich Ihnen viel Spaß und Erfolg mit der PowerShell 5.0!

Dr. Holger Schwichtenberg

Essen, im Januar 2016

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 60 Fachbücher beim Carl Hanser Verlag, bei Microsoft Press und Addison-Wesley sowie mehr als 850 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, VS One, NRW.Conf, Net.Object Days, Windows Forum)
- Zertifikate und Auszeichnungen von Microsoft:
 - Microsoft Most Valuable Professional (MVP)
 - Microsoft Certified Solution Developer (MCSD)
- Thematische Schwerpunkte:
 - Microsoft .NET Framework, Visual Studio, C#, Visual Basic
 - .NET-Architektur/Auswahl von .NET-Technologien

- Einführung von .NET Framework und Visual Studio/Migration auf .NET
- Webanwendungsentwicklung mit IIS, ASP.NET und JavaScript/AJAX
- Enterprise .NET, verteilte Systeme/Webservices mit .NET
- Relationale Datenbanken, XML, Datenzugriffsstrategien
- Objektrelationales Mapping (ORM), insbesondere ADO.NET Entity Framework
- Windows PowerShell (WPS) und Windows Management Instrumentation (WMI)
- Ehrenamtliche Community-Tätigkeiten:
 - Vortragender für die International .NET Association (INETA)
 - Betrieb diverser Community-Websites: www.dotnetframework.de, www.entwicklerlexikon.de, www.windows-scripting.de, www.aspnetdev.de u. a.
- Firmenwebsites: <http://www.IT-Visions.de> und <http://www.5minds.de>
- Weblog: <http://www.dotnet-doktor.de>
- Kontakt: buero@IT-Visions.de sowie Telefon 0201-6495 90-0

Über den Co-Autor Peter Monadjemi



Peter Monadjemi ist freiberuflicher Trainer und Autor mit den Schwerpunkten PowerShell, Windows-Server-Administration und Softwareentwicklung auf der Grundlage des .NET Frameworks. Er kennt die PowerShell bereits seit der Zeit, in der sie als Vorabversion noch unter dem Projektnamen „Monad“ lief – ein Codename, der für ihn eine Art „Wink des Schicksals“ war. Seitdem hat er viel Zeit mit der PowerShell verbracht und Hunderte von Skripten geschrieben. Er lebt in Esslingen am Neckar.

A

Teil A: PowerShell-Basiswissen

Dieser Buchteil informiert über die Basiskonzepte der PowerShell, insbesondere Commandlets, Pipelines, Navigation und Skripte. Außerdem werden am Ende dieses Teils Werkzeuge vorgestellt.

1

Erste Schritte mit der Windows PowerShell

Das DOS-ähnliche Kommandozeilenfenster hat viele Windows-Versionen in beinahe unveränderter Form überlebt. Mit der Windows PowerShell (WPS) besitzt Microsoft nun endlich einen Nachfolger, der es mit den Unix-Shells aufnehmen kann und diese in Hinblick auf Eleganz und Robustheit in einigen Punkten auch überbieten kann. Die PowerShell ist eine Adaption des Konzepts von Unix-Shells auf Windows unter Verwendung des .NET Frameworks und mit Anbindung an die Windows Management Instrumentation (WMI).

■ 1.1 Was ist die Windows PowerShell?

In einem Satz: Die Windows PowerShell (WPS) ist eine neue, .NET-basierte Umgebung für interaktive Systemadministration und Scripting auf der Windows-Plattform.

Die Kernfunktionen der PowerShell sind:

- Zahlreiche eingebaute Befehle, die „Commandlets“ genannt werden
- Zugang zu allen Systemobjekten, die durch COM-Bibliotheken, das .NET Framework und die Windows Management Instrumentation (WMI) bereitgestellt werden
- Robuster Datenaustausch zwischen Commandlets durch Pipelines basierend auf typisierten Objekten
- Ein einheitliches Navigationsparadigma für verschiedene Speicher (z.B. Dateisystem, Registrierungsdatenbank, Zertifikatsspeicher, Active Directory und Umgebungsvariablen)
- Eine einfach zu erlernende, aber mächtige Skriptsprache mit wahlweise schwacher oder starker Typisierung
- Ein Sicherheitsmodell, das die Ausführung unerwünschter Skripte unterbindet
- Integrierte Funktionen für Ablaufverfolgung und Debugging
- Die PowerShell kann um eigene Befehle erweitert werden.
- Die PowerShell kann in eigene Anwendungen integriert werden (Hosting).

■ 1.2 Geschichte der Windows PowerShell

Das Active Scripting ist einigen Administratoren zu komplex, weil es viel Wissen über objektorientiertes Programmieren und das Component Object Model (COM) voraussetzt. Die vielen Ausnahmen und Ungereimtheiten im Active Scripting erschweren das Erlernen von Windows Script Host (WSH) und der zugehörigen Komponentenbibliotheken.

Schon im Zuge der Entwicklung des Windows Server 2003 gab Microsoft zu, dass man Unix-Administratoren zum Interview über ihr tägliches Handwerkszeug gebeten hatte. Das kurzfristige Ergebnis war eine große Menge zusätzlicher Kommandozeilenwerkzeuge. Langfristig setzt Microsoft jedoch auf eine Ablösung des DOS-ähnlichen Konsolenfensters durch eine neue Scripting-Umgebung.

Mit dem Erscheinen des .NET Frameworks im Jahre 2002 wurde lange über einen WSH.NET spekuliert. Microsoft stellte jedoch die Neuentwicklung des WSH für das .NET Framework ein, als abzusehen war, dass die Verwendung von .NET-basierten Programmiersprachen wie C# und Visual Basic .NET dem Administrator nur noch mehr Kenntnisse über objektorientierte Softwareentwicklung abverlangen würde.

Microsoft beobachtete in der Unix-Welt eine hohe Zufriedenheit mit den dortigen Kommandozeilen-Shells und entschloss sich daher, das Konzept der Unix-Shells, insbesondere das Pipelining, mit dem .NET Framework zusammenzubringen und daraus eine .NET-basierte Windows Shell zu entwickeln. Diese ist so einfach wie eine Unix-Shell, aber kann so mächtig wie das .NET Framework sein.

In einer ersten Beta-Version wurde die neue Shell schon unter dem Codenamen „Monad“ auf der Professional Developer Conference (PDC) im Oktober 2003 in Los Angeles vorgestellt. Nach den Zwischenstufen „Microsoft Shell (MSH)“ und „Microsoft Command Shell“ trägt die neue Skriptumgebung seit Mai 2006 den Namen „Windows PowerShell“.

Die PowerShell 1.0 erschien am 6.11.2006 zeitgleich mit Windows Vista, war aber dort nicht enthalten, sondern musste heruntergeladen und nachinstalliert werden.

Die PowerShell 2.0 ist zusammen mit Windows 7/Windows Server 2008 R2 erschienen am 22.7.2009.

Die PowerShell 3.0 ist zusammen mit Windows 8/Windows Server 2012 erschienen am 15.8.2012.

Die PowerShell 4.0 ist zusammen mit Windows 8.1/Windows Server 2012 R2 am 9.9.2013 erschienen.

Die PowerShell 5.0 ist als Teil von Windows 10 erschienen am 29.7.2015. Abweichend von den bisherigen Gepflogenheiten ist die PowerShell 5.0 als Erweiterung für Windows Server 2008 R2 (mit Service Pack 1) und Windows Server 2012/2012 R2 erst deutlich später am 16.12.2015 erschienen. Für Windows 7 und Windows 8.1 sollte es erst gar keine Version mehr geben. Doch am 18.12.2015 hatte Microsoft ein Einsehen mit den Kunden und lieferte die PowerShell 5.0 auch für diese Betriebssysteme nach. Kurioserweise musste Microsoft den Download dann am 23.12. wegen eines gravierenden Fehlers vor einigen Wochen vom Netz nehmen. Auch der Windows Server 2016, der zum Redaktionsschluss dieses Buchs noch in der Preview-Phase ist, enthält auf dem derzeitigen Stand die PowerShell 5.0.



HINWEIS: Mit Windows 10 hat Microsoft das Auslieferungsverfahren auf „Windows as a Service“ umgestellt. Dies bedeutet, dass Microsoft über Windows Update im Sinne der neuen „agilen“ Strategie nun auch ständig neue Funktionen ausliefern. Dies betrifft ebenso die Windows PowerShell, die dann zukünftig auch auf diesem Wege häufigere Aktualisierungen erfahren kann. Wie häufig dies sein wird, ist zum Reaktionsschluss dieses Buchs noch offen.

■ 1.3 Motivation zur Windows PowerShell

Falls Sie eine Motivation brauchen, sich mit der PowerShell zu beschäftigen, wird dieses Kapitel sie Ihnen liefern. Es stellt die Lösung für eine typische Scripting-Aufgabe sowohl im „alten“ Windows Script Host (WSH) als auch in der „neuen“ Windows PowerShell vor.

Zur Motivation, sich mit der Windows PowerShell zu beschäftigen, soll folgendes Beispiel aus der Praxis dienen. Es soll ein Inventarisierungsskript für Software erstellt werden, das die installierten MSI-Pakete mit Hilfe der Windows Management Instrumentation (WMI) von mehreren Computern ausliest und die Ergebnisse in einer CSV-Datei (*softwareinventar.csv*) zusammenfasst. Die Namen (oder IP-Adressen) der abzufragenden Computer sollen in einer Textdatei (*computernamen.txt*) stehen.

Die Lösung mit dem WSH benötigt 90 Codezeilen (inklusive Kommentare und Parametrisierungen). In der Windows PowerShell lässt sich das Gleiche in nur 13 Zeilen ausdrücken. Wenn man auf die Kommentare und die Parametrisierung verzichtet, dann reicht sogar genau eine Zeile.

Listing 1.1 Softwareinventarisierung – Lösung 1 mit dem WSH

[3_Einsatzgebiete/Software/Software_Inventory.vbs]

```
' -----  
' Skriptname: Software_inventar.vbs  
' Autor: Dr. Holger Schwichtenberg 2004-2007  
' -----  
' Dieses Skript erstellt eine Liste  
' der installierten Software  
' Version 1.02 (28.1.2007)  
' -----  
Option Explicit  
  
' --- Vorgabewerte  
Const Trennzeichen = ";" ' Trennzeichen für Spalten in der Ausgabedatei  
Const Eingabedateiname = "computernamen.txt"  
Const Ausgabedateiname = "softwareinventar.csv"  
Const Bedingung = "SELECT * FROM Win32_Product where not Vendor like '%Microsoft%'"  
  
Dim objFSO' Dateisystem-Objekt  
Dim objTX    ' Textdatei-Objekt für die Liste der zu durchsuchenden computer  
Dim i      ' Zähler für Computer  
Dim computer ' Name des aktuellen computers
```