

Vorwort	XV
----------------------	-----------

Über den Autor Dr. Holger Schwichtenberg	XXI
---	------------

1 Erste Schritte mit der PowerShell	1
--	----------

1.1 Was ist die PowerShell?	1
1.2 Windows PowerShell versus PowerShell Core	2
1.3 Windows PowerShell herunterladen und auf anderen Windows- Betriebssystemen installieren	2
1.4 Die Windows PowerShell testen	7
1.4.1 PowerShell im interaktiven Modus	7
1.4.2 Installierte Version ermitteln	10
1.4.3 PowerShell im Skriptmodus	11
1.4.4 Skript eingeben	11
1.4.5 Skript starten	12
1.4.6 Skriptausführungsrichtlinie ändern	13
1.4.7 Farben ändern	16
1.5 Woher kommen die Commandlets?	17
1.6 PowerShell Community Extensions (PSCX) herunterladen und installieren .	18
1.7 Den Windows PowerShell-Editor „ISE“ verwenden	20

2 Fakten zur PowerShell	23
--------------------------------------	-----------

2.1 Geschichte der PowerShell	23
2.2 Warum PowerShell einsetzen?	24
2.3 Einflussfaktoren auf die Entwicklung der PowerShell	28
2.4 Betriebssysteme mit vorinstallierter PowerShell	29
2.5 Anbindung an Klassenbibliotheken	31
2.6 PowerShell versus WSH	31

3	Einzelbefehle der PowerShell	35
3.1	Commandlets	35
3.1.1	Aufbau eines Commandlets	35
3.1.2	Aufruf von Commandlets	36
3.1.3	Commandlet-Parameter	36
3.1.4	Platzhalter bei den Parameterwerten	39
3.1.5	Abkürzungen für Parameter	40
3.1.6	Allgemeine Parameter (Common Parameters)	41
3.1.7	Dynamische Parameter	45
3.1.8	Zeilenumbrüche	45
3.1.9	PowerShell-Module	46
3.1.10	Prozessmodell	47
3.1.11	Aufruf von Commandlets aus anderen Prozessen heraus	47
3.1.12	Namenskonventionen	48
3.2	Aliase	48
3.2.1	Aliase auflisten	49
3.2.2	Neue Aliase anlegen	53
3.2.3	Aliase für Eigenschaften	54
3.3	Ausdrücke	56
3.4	Externe Befehle	57
3.5	Dateinamen	59
3.6	Aufgaben zu diesem Kapitel	59
4	Hilfefunktionen	61
4.1	Auflisten der verfügbaren Befehle	61
4.2	Volltextsuche	63
4.3	Erläuterungen zu den Befehlen	64
4.4	Hilfe zu Parametern	65
4.5	Hilfe mit Show-Command	67
4.6	Hilfefenster	69
4.7	Allgemeine Hilfetexte	70
4.8	Aktualisieren der Hilfedateien	70
4.9	Online-Hilfe	72
4.10	Fehlende Hilfetexte	73
4.11	Dokumentation der .NET-Klassen	75
4.12	Aufgaben zu diesem Kapitel	78
5	Objektorientiertes Pipelining	79
5.1	Pipeline-Operator	79
5.2	.NET-Objekte in der Pipeline	80
5.3	Pipeline Processor	82
5.4	Pipelining von Parametern	83

5.5	Pipelining von klassischen Befehlen	86
5.6	Anzahl der Objekte in der Pipeline	87
5.7	Zeilenumbrüche in Pipelines	88
5.8	Zugriff auf einzelne Objekte aus einer Menge	88
5.9	Zugriff auf einzelne Werte in einem Objekt	90
5.10	Methoden ausführen	92
5.11	Analyse des Pipeline-Inhalts	94
5.12	Filtern	105
5.13	Zusammenfassung von Pipeline-Inhalten	108
5.14	„Kastrierung“ von Objekten in der Pipeline	109
5.15	Sortieren	110
5.16	Duplikate entfernen	111
5.17	Gruppierung	112
5.18	Berechnungen	114
5.19	Zwischenschritte in der Pipeline mit Variablen	114
5.20	Verzweigungen in der Pipeline	115
5.21	Vergleiche zwischen Objekten	117
5.22	Zusammenfassung	118
5.23	Aufgaben zu diesem Kapitel	119
6	PowerShell-Skripte	121
6.1	Skriptdateien	121
6.2	Start eines Skripts	123
6.3	Aliase für Skripte verwenden	125
6.4	Parameter für Skripte	125
6.5	Skripte dauerhaft einbinden (Dot Sourcing)	127
6.6	Das aktuelle Skriptverzeichnis	127
6.7	Sicherheitsfunktionen für PowerShell-Skripte	128
6.8	Anforderungsdefinitionen von Skripten	130
6.9	Skripte anhalten	131
6.10	Versionierung und Versionsverwaltung von Skripten	131
	6.10.1 Versionierung	131
	6.10.2 Versionsverwaltung (Versionskontrolle)	132
6.11	Aufgaben zu diesem Kapitel	133
7	Die PowerShell-Skriptsprache	135
7.1	Hilfe zur PowerShell-Skriptsprache	135
7.2	Befehlstrennung	136
7.3	Kommentare	136
7.4	Variablen	137
	7.4.1 Variablen in der PowerShell	137

7.4.2	Typisierung	138
7.4.3	Datentypen/Typbezeichner in PowerShell	139
7.4.4	Typisierungszwang	141
7.4.5	Typkonvertierung (Typumwandlung)	142
7.4.6	Gültigkeitsbereiche (Scope)	143
7.4.7	Variablen leeren oder löschen	144
7.4.8	Variablentyp ermitteln	145
7.4.9	Vordefinierte Variablen	145
7.5	Variablenbedingungen	147
7.6	Zahlen	148
7.7	Zeichenketten (Strings)	150
7.7.1	Zeichenkettenlitterale	150
7.7.2	Zeichenketten zusammensetzen	151
7.7.3	Variablenuflösung in Zeichenketten	151
7.7.4	Wiederholte Zeichenketten	153
7.7.5	Leere Zeichenketten	153
7.7.6	Sonderzeichen in Zeichenketten	154
7.7.7	Bearbeitungsmöglichkeiten für Zeichenketten	155
7.7.8	Zeichenketten ersetzen	157
7.7.9	Zeichenketten trennen und verbinden	158
7.8	Reguläre Ausdrücke	159
7.8.1	Mustervergleichsoperatoren	159
7.8.2	Allgemeiner Aufbau von regulären Ausdrücken	161
7.9	Datum und Uhrzeit	165
7.10	Arrays	167
7.10.1	Deklaration	167
7.10.2	Arrayoperationen	167
7.10.3	Array auflisten	169
7.10.4	Arrays verbinden	169
7.10.5	Mehrdimensionale Arrays	170
7.11	ArrayList	170
7.12	Assoziative Arrays (Hash-Tabellen)	171
7.13	Operatoren	172
7.13.1	Vergleichsoperatoren	172
7.13.2	Arithmetische Operatoren	172
7.13.3	Zuweisungsoperator	173
7.13.4	Bit-Operatoren	175
7.13.5	Aufrufoperator	175
7.14	Überblick über die Kontrollkonstrukte	176
7.15	Schleifen	177
7.16	Bedingungen	182
7.17	Unterrouinen (Prozedur/Funktionen)	184
7.17.1	Prozedur versus Funktion	184

7.17.2 Prozeduren	185
7.17.3 Funktionen (mit Rückgabewerten)	185
7.17.4 Art der Rückgabewerte	188
7.17.5 Parameterübergabe	188
7.18 Eingebaute Funktionen	190
7.19 Fehlerbehandlung	191
7.20 Objektorientiertes Programmieren mit Klassen	200
7.21 Aufgaben zu diesem Kapitel	202
8 Ausgaben	203
8.1 Ausgabe-Commandlets	203
8.2 Benutzerdefinierte Tabellenformatierung	206
8.3 Benutzerdefinierte Listenausgabe	209
8.4 Mehrspaltige Ausgabe	209
8.5 Out-GridView	210
8.6 Standardausgabe	212
8.7 Einschränkung der Ausgabe	215
8.8 Seitenweise Ausgabe	216
8.9 Ausgabe einzelner Werte	217
8.10 Details zum Ausgabeoperator	220
8.11 Ausgabe von Methodenergebnissen und Unterobjekten in Pipelines	223
8.12 Ausgabe von Methodenergebnissen und Unterobjekten in Zeichenketten ...	224
8.13 Unterdrückung der Ausgabe	224
8.14 Ausgaben an Drucker	225
8.15 Ausgaben in Dateien	225
8.16 Umleitungen (Redirection)	226
8.17 Fortschrittsanzeige	227
8.18 Sprachausgabe	227
8.19 Aufgaben zu diesem Kapitel	228
9 Benutzereingaben	229
9.1 Read-Host	229
9.2 Benutzerauswahl	230
9.3 Grafischer Eingabedialog	231
9.4 Dialogfenster	232
9.5 Authentifizierungsdialog	232
9.6 Zwischenablage (Clipboard)	234
9.7 Aufgaben zu diesem Kapitel	235

10 Das PowerShell-Navigationsmodell (PowerShell Provider)	237
10.1 Einführungsbeispiel: Navigation in der Registrierungsdatenbank	237
10.2 Provider und Laufwerke	239
10.3 Navigationsbefehle	241
10.4 Pfadangaben.	241
10.5 Beispiel.	243
10.6 Eigene Laufwerke definieren	244
10.7 Aufgaben zu diesem Kapitel	245
 11 Fernausführung (Remoting)	 249
11.1 RPC-Fernabfrage ohne WS-Management	250
11.2 Anforderungen an PowerShell Remoting	251
11.3 Rechte für PowerShell-Remoting	252
11.4 Einrichten von PowerShell Remoting	253
11.5 Überblick über die Fernausführungs-Commandlets	255
11.6 Interaktive Fernverbindungen im Telnet-Stil	256
11.7 Fernausführung von Befehlen	257
11.8 Parameterübergabe an die Fernausführung	261
11.9 Fernausführung von Skripten	262
11.10 Ausführung auf mehreren Computern	263
11.11 Sitzungen	264
11.11.1 Commandlets zur Sitzungsverwaltung	265
11.11.2 Sitzungen erstellen	266
11.11.3 Kopieren von Dateien in Sitzungen	266
11.11.4 Schließen von Sitzungen	267
11.11.5 Sitzungskonfigurationen	267
11.11.6 Zugriffsrechte für Fernaufrufe	267
11.12 Implizites Remoting	269
11.13 Zugriff auf entfernte Computer außerhalb der eigenen Domäne	270
11.14 Verwaltung des WS-Management-Dienstes	274
11.15 PowerShell Direct für Hyper-V	275
11.16 Praxisbeispiel zu PowerShell Direct	277
11.17 Aufgaben zu diesem Kapitel	279
 12 Verwendung von .NET-Klassen	 281
12.1 Microsoft Developer Network (MSDN)	281
12.2 Erzeugen von Instanzen	282
12.3 Parameterbehaftete Konstruktoren	284
12.4 Initialisierung von Objekten	286
12.5 Nutzung von Attributen und Methoden	286
12.6 Statische Mitglieder in .NET-Klassen und statische .NET-Klassen	289

12.7	Generische Klassen nutzen	293
12.8	Zugriff auf bestehende Objekte	294
12.9	Laden von Assemblies	294
12.10	Objektanalyse	297
12.11	Auflistungen (Enumerationen)	298
12.12	Verknüpfen von Aufzählungswerten	299
12.13	Aufgaben zu diesem Kapitel	299
13	Verwendung von COM-Klassen	301
13.1	Erzeugen von COM-Instanzen	301
13.2	Nutzung von Attributen und Methoden	302
13.3	Liste aller COM-Klassen	303
13.4	Holen bestehender COM-Instanzen	304
13.5	Distributed COM (DCOM)	304
13.6	Aufgaben zu diesem Kapitel	305
14	Zugriff auf die Windows Management Instrumentation (WMI) ..	307
14.1	WMI in der PowerShell	307
14.2	Open Management Infrastructure (OMI)	309
14.3	Abruf von WMI-Objektmengen	309
14.4	Fernzugriffe	310
14.5	Filtern und Abfragen	311
14.6	Liste aller WMI-Klassen	315
14.7	Hintergrundwissen: WMI-Klassenprojektion mit dem PowerShell-WMI-Objektadapter	315
14.8	Beschränkung der Ausgabeliste bei WMI-Objekten	320
14.9	Zugriff auf einzelne Mitglieder von WMI-Klassen	322
14.10	Werte setzen in WMI-Objekten	322
14.11	Umgang mit WMI-Datumsangaben	324
14.12	Methodenaufrufe	325
14.13	Neue WMI-Instanzen erzeugen	326
14.14	Instanzen entfernen	327
14.15	Commandlet Definition XML-Datei (CDXML)	328
14.16	Aufgaben zu diesem Kapitel	330
15	Fehlersuche	333
15.1	Detailinformationen	333
15.2	Einzel schrittmodus	335
15.3	Zeitmessung	336
15.4	Ablaufverfolgung (Tracing)	336
15.4.1	Tracesources	336

15.4.2	Verfolgung eines Einzelbefehls	337
15.4.3	Generelle Ablaufverfolgung	337
15.5	Erweiterte Protokollierung aktivieren	338
15.6	Script-Debugging in der ISE	339
15.7	Kommandozeilenbasiertes Script-Debugging	340
16	Standardeinstellungen ändern mit Profilskripten	343
16.1	Profilpfade	343
16.2	Ausführungsreihenfolge	345
16.3	Beispiel für eine Profildatei	345
16.4	Starten der PowerShell ohne Profilskripte	346
17	PowerShell-Module	347
17.1	Überblick über die Commandlets	347
17.2	Modularchitektur	348
17.3	Aufbau eines Moduls	349
17.4	Module aus dem Netz herunterladen und installieren mit PowerShellGet	350
17.5	Module manuell installieren	357
17.6	Doppeldeutige Namen	357
17.7	Importieren von Modulen	360
17.8	Entfernen von Modulen	363
18	Praxislösungen	365
18.1	Leere Ordner löschen	365
18.2	Fotos nach Aufnahmedatum sortieren	366
18.3	Zufällige Dateisystemstruktur erzeugen	368
18.4	Freigaben anlegen	369
18.4.1	WMI-Klassen	369
18.4.2	Freigaben auflisten	371
18.4.3	Freigaben anlegen (mit WMI)	371
18.4.4	Berechtigungen auf Freigaben setzen	373
18.4.5	Freigaben anlegen (mit PowerShell-Commandlets)	376
18.4.6	Freigaben anlegen auf Basis einer XML-Datei	379
18.5	Netzwerkconfiguration	381
18.6	Massenanlegen von Registry-Schlüsseln	383
18.7	Massenanlegen von Active-Directory-Benutzerkonten	384
18.7.1	Benutzer und Gruppen anlegen aus einer Datenbank (Microsoft Access oder Microsoft SQL Server)	385
18.7.2	Benutzer und Gruppen anlegen aus einer CSV-Datei	399
18.8	Massenanlegen von IIS-Websites	402

18.9	Softwareinstallation	404
18.10	Virtuelles System in Hyper-V anlegen	406
19	PowerShell Core 6.0 für Windows, Linux und macOS	409
19.1	Geschichte der PowerShell Core	409
19.2	Vergleich zwischen Windows PowerShell und PowerShell Core	410
19.3	Motivation für den Einsatz der PowerShell Core auf Linux und macOS ...	411
19.4	PowerShell Core installieren und testen	413
19.4.1	Installation und Test auf Windows	413
19.4.2	Installation und Test auf Ubuntu Linux	415
19.4.3	Installation und Test auf macOS	417
19.5	Funktionsumfang der PowerShell Core	418
19.6	Entfallene Commandlets in PowerShell Core	420
19.6.1	PowerShell-Kern-Befehle, die in PowerShell Core fehlen	420
19.6.2	Befehle, die zusätzlich unter Linux und macOS fehlen	423
19.7	Erweiterungsmodule nutzen in PowerShell Core	425
19.7.1	Eingebaute Module in PowerShell Core	425
19.7.2	Windows-Module in PowerShell Core nutzen	427
19.7.3	Module der PowerShell Gallery in PowerShell Core	427
19.8	Geänderte Funktionen in PowerShell Core	428
19.8.1	Änderungen der Parameter von pwsh.exe	428
19.8.2	Geänderte Pfade	428
19.8.3	Weitere Änderungen	430
19.9	Neue Funktionen der PowerShell Core	430
19.9.1	Neue eingebaute Variablen	430
19.9.2	Neue Commandlets	431
19.9.3	Sonstige Verbesserungen in PowerShell Core	431
19.10	PowerShell-Core-Konsole	432
19.11	VSCoDe-PowerShell als Editor für PowerShell Core	433
19.11.1	PowerShell Core zur Skriptausführung konfigurieren	434
19.11.2	PowerShell Core für das Terminalfenster in VSCoDe festlegen ...	436
19.12	Verwendung auf Linux und macOS	438
19.12.1	Praxisbeispiel: Linux-Benutzerliste auswerten	440
19.12.2	Praxisbeispiel: Offene Ports auswerten	441
19.12.3	Praxisbeispiel: Dateien unter Linux und macOS verstecken und versteckte Dateien auflisten	443
19.13	PowerShell-Remoting via SSH	444
19.13.1	OpenSSH auf Windows	444
19.13.2	PowerShell Remoting mit SSH	446
19.14	Dokumentation zur PowerShell Core	448
19.15	Quellcode zur PowerShell Core	450

20	Weiterführende Literatur.....	453
21	Lösungen.....	457
Index		473