

Inhalt

Vorwort	XXV
Teil A: PowerShell-Basiswissen	1
1 Fakten zur PowerShell	3
1.1 Was ist die PowerShell?	3
1.2 Geschichte der PowerShell	4
1.3 Welche Varianten und Versionen der PowerShell gibt es?	6
1.4 Windows PowerShell versus PowerShell Core versus PowerShell 7.0	7
1.5 Motivation zur PowerShell	8
1.6 Betriebssysteme mit vorinstallierter PowerShell	11
1.7 Einflussfaktoren auf die Entwicklung der PowerShell	12
1.8 Anbindung an Klassenbibliotheken	14
1.9 PowerShell versus WSH	14
2 Erste Schritte mit der PowerShell	17
2.1 Windows PowerShell herunterladen und auf anderen Windows-Betriebssystemen installieren	17
2.2 Die Windows PowerShell testen	21
2.3 Woher kommen die PowerShell-Befehle?	30
2.4 PowerShell Community Extensions (PSCX) herunterladen und installieren	31
2.5 Den Windows PowerShell-Editor „ISE“ verwenden	34
2.6 PowerShell 7 installieren und testen	36
3 Einzelbefehle der PowerShell	45
3.1 Commandlets	45
3.2 Aliase	58
3.3 Ausdrücke	66
3.4 Externe Befehle (klassische Kommandozeilenbefehle)	67
3.5 Dateinamen	69

4	Hilfefunktionen	71
4.1	Auflisten der verfügbaren Befehle	71
4.2	Praxistipp: Den Standort eines Kommandozeilenbefehls suchen	72
4.3	Anzahl der Befehle	73
4.4	Volltextsuche	75
4.5	Erläuterungen zu den Befehlen	76
4.6	Hilfe zu Parametern	77
4.7	Hilfe mit Show-Command	78
4.8	Hilfefenster	80
4.9	Allgemeine Hilfetexte	81
4.10	Aktualisieren der Hilfedateien	82
4.11	Online-Hilfe	84
4.12	Fehlende Hilfetexte	85
4.13	Dokumentation der .NETKlassen	86
5	Objektorientiertes Pipelining	89
5.1	Befehlsübersicht	89
5.2	Pipeline-Operator	90
5.3	.NET-Objekte in der Pipeline	91
5.4	Pipeline Processor	92
5.5	Pipelining von Parametern	94
5.6	Pipelining von klassischen Befehlen	96
5.7	Zeilenumbrüche in Pipelines	98
5.8	Schleifen	99
5.9	Zugriff auf einzelne Objekte aus einer Menge	102
5.10	Zugriff auf einzelne Werte in einem Objekt	103
5.11	Methoden ausführen	105
5.12	Analyse des Pipeline-Inhalts	107
5.13	Filtern	122
5.14	Zusammenfassung von Pipeline-Inhalten	127
5.15	„Kastrierung“ von Objekten in der Pipeline	127
5.16	Sortieren	128
5.17	Duplikate entfernen	129
5.18	Gruppierung	130
5.19	Objekte verbinden mit Join-String	136
5.20	Berechnungen	137
5.21	Zwischenschritte in der Pipeline mit Variablen	138
5.22	Verzweigungen in der Pipeline	139
5.23	Vergleiche zwischen Objekten	141
5.24	Weitere Praxislösungen	142

6	PowerShell-Skripte	145
6.1	Skriptdateien	145
6.2	Start eines Skripts	147
6.3	Aliase für Skripte verwenden	149
6.4	Parameter für Skripte	149
6.5	Skripte dauerhaft einbinden (Dot Sourcing)	151
6.6	Das aktuelle Skriptverzeichnis	151
6.7	Sicherheitsfunktionen für PowerShell-Skripte	152
6.8	Skripte mit vollen Rechten (Elevation)	154
6.9	Blockierte PowerShell-Skripte	155
6.10	PowerShell-Skripte im Kontextmenü des Windows Explorers	156
6.11	Anforderungsdefinitionen von Skripten	158
6.12	Skripte anhalten	158
6.13	Versionierung und Versionsverwaltung von Skripten	158
7	PowerShell-Skriptsprache	161
7.1	Hilfe zur PowerShell-Skriptsprache	161
7.2	Befehlstrennung	162
7.3	Kommentare	162
7.4	Variablen	163
7.5	Variablenbedingungen	176
7.6	Zahlen	177
7.7	Zeichenketten (Strings)	179
7.8	Reguläre Ausdrücke	189
7.9	Datum und Uhrzeit	195
7.10	Objekte	196
7.11	Arrays	197
7.12	ArrayList	200
7.13	Assoziative Arrays (Hash-Tabellen)	201
7.14	Operatoren	202
7.15	Überblick über die Kontrollkonstrukte	206
7.16	Schleifen	207
7.17	Bedingungen	212
7.18	Unterrouinen (Prozedur/Funktionen)	214
7.19	Eingebaute Funktionen	220
7.20	Fehlerausgabe	220
7.21	Fehlerbehandlung	221
7.22	Laufzeitfehler erzeugen	233
7.23	Objektorientiertes Programmieren mit Klassen	233

8	Ausgaben	237
8.1	Ausgabe-Commandlets	237
8.2	Benutzerdefinierte Tabellenformatierung	240
8.3	Benutzerdefinierte Listenausgabe	242
8.4	Mehrspaltige Ausgabe	242
8.5	Out-GridView	243
8.6	Standardausgabe	245
8.7	Einschränkung der Ausgabe	248
8.8	Seitenweise Ausgabe	248
8.9	Ausgabe einzelner Werte	249
8.10	Details zum Ausgabeoperator	252
8.11	Ausgabe von Methodenergebnissen und Unterobjekten in Pipelines	255
8.12	Ausgabe von Methodenergebnissen und Unterobjekten in Zeichenketten	255
8.13	Unterdrückung der Ausgabe	256
8.14	Ausgaben an Drucker	257
8.15	Ausgaben in Dateien	257
8.16	Umleitungen (Redirection)	258
8.17	Fortschrittsanzeige	258
8.18	Sprachausgabe	259
9	Das PowerShell-Navigationsmodell (PowerShell Provider)	261
9.1	Einführungsbeispiel: Navigation in der Registrierungsdatenbank	261
9.2	Provider und Laufwerke	262
9.3	Navigationsbefehle	265
9.4	Pfadangaben	265
9.5	Beispiel	267
9.6	Eigene Laufwerke definieren	268
10	Fernausführung (Remoting)	269
10.1	RPC-Fernabfrage ohne WS-Management	270
10.2	Anforderungen an PowerShell Remoting	271
10.3	Rechte für PowerShell-Remoting	272
10.4	Einrichten von PowerShell Remoting	273
10.5	Überblick über die Fernausführungs-Commandlets	275
10.6	Interaktive Fernverbindungen im Telnet-Stil	276
10.7	Fernausführung von Befehlen	277
10.8	Parameterübergabe an die Fernausführung	281
10.9	Fernausführung von Skripten	282
10.10	Ausführung auf mehreren Computern	283
10.11	Sitzungen	284

10.12	Implizites Remoting	289
10.13	Zugriff auf entfernte Computer außerhalb der eigenen Domäne	290
10.14	Verwaltung des WS-Management-Dienstes	294
10.15	PowerShell Direct für Hyper-V	295
10.16	Praxislösung zu PowerShell Direct	297
11	PowerShell-Werkzeuge	301
11.1	PowerShell-Standardkonsole	301
11.2	Windows Terminal	311
11.3	PowerShell Integrated Scripting Environment (ISE)	313
11.4	PowerShell Script Analyzer	325
11.5	PowerShell Analyzer	330
11.6	PowerShell Tools for Visual Studio	331
11.7	PowerShell Pro Tools for Visual Studio	333
11.8	NuGet Package Manager	333
11.9	Visual Studio Code mit PowerShell-Erweiterung	334
11.10	PowerShell-Erweiterungen für andere Editoren	337
11.11	PowerShell Web Access (PSWA)	337
11.12	Azure Cloud Shell	343
11.13	ISE Steroids	343
11.14	PowerShellPlus	344
11.15	PoshConsole	347
11.16	PowerGUI	348
11.17	PrimalScript	349
11.18	PowerShell Help	351
11.19	CIM Explorer for PowerShell ISE	351
11.20	PowerShell Help Reader	352
11.21	PowerShell Remoting	353
12	Windows PowerShell Core 5.1 in Windows Nano Server	355
12.1	Installation	355
12.2	PowerShell-Skriptsprache	355
12.3	Werkzeuge	356
12.4	Fehlende Funktionen	356
13	PowerShell 7 für Windows, Linux und macOS	357
13.1	Motivation für den Einsatz der PowerShell 7 auf Linux und macOS	357
13.2	Basis der PowerShell 7	358
13.3	Identifizierung der PowerShell 7	359
13.4	Funktionsumfang der PowerShell 7	359
13.5	Entfallene Befehle in PowerShell 7	362

13.6	Erweiterungsmodule nutzen in PowerShell 7	367
13.7	Geänderte Funktionen in PowerShell 7	372
13.8	Neue Funktionen der PowerShell 7	374
13.9	PowerShell 7-Konsole	378
13.10	VSCoDe-PowerShell als Editor für PowerShell 7	378
13.11	Verwendung von PowerShell 7 auf Linux und macOS	383
13.12	PowerShell-Remoting via SSH	389
13.13	Dokumentation zur PowerShell 7	393
13.14	Quellcode zur PowerShell 7	394
Teil B: PowerShell-Aufbauwissen		397
14 Verwendung von .NET-Klassen		399
14.1	.NET versus .NET Core	399
14.2	Ermitteln der verwendeten .NET-Version	400
14.3	.NET-Bibliotheken	401
14.4	Microsoft Developer Network (MSDN)	403
14.5	Überblick über die Verwendung von .NET-Klassen	404
14.6	Erzeugen von Instanzen	404
14.7	Parameterbehaftete Konstruktoren	406
14.8	Initialisierung von Objekten	408
14.9	Nutzung von Attributen und Methoden	408
14.10	Statische Mitglieder in .NET-Klassen und statische .NET-Klassen	410
14.11	Generische Klassen nutzen	414
14.12	Zugriff auf bestehende Objekte	415
14.13	Laden von Assemblies	415
14.14	Liste der geladenen Assemblies	418
14.15	Verwenden von Nuget-Assemblies	419
14.16	Objektanalyse	421
14.17	Aufzählungstypen (Auflistungen/Enumerationen)	422
15 Verwendung von COM-Klassen		427
15.1	Unterschiede zwischen COM und .NET	427
15.2	Erzeugen von COM-Instanzen	428
15.3	Abruf der Metadaten	428
15.4	Nutzung von Attributen und Methoden	429
15.5	Liste aller COM-Klassen	430
15.6	Holen bestehender COM-Instanzen	431
15.7	Distributed COM (DCOM)	431

16	Zugriff auf die Windows Management Instrumentation (WMI) ..	433
16.1	Einführung in WMI	433
16.2	WMI in der PowerShell	459
16.3	Open Management Infrastructure (OMI)	461
16.4	Abruf von WMI-Objektmengen	461
16.5	Fernzugriffe	462
16.6	Filtern und Abfragen	463
16.7	Liste aller WMI-Klassen	466
16.8	Hintergrundwissen: WMI-Klassenprojektion mit dem PowerShell-WMI-Objektadapter	467
16.9	Beschränkung der Ausgabeliste bei WMI-Objekten	471
16.10	Zugriff auf einzelne Mitglieder von WMI-Klassen	472
16.11	Werte setzen in WMI-Objekten	473
16.12	Umgang mit WMI-Datumsangaben	475
16.13	Methodenaufrufe	476
16.14	Neue WMI-Instanzen erzeugen	477
16.15	Instanzen entfernen	478
16.16	Commandlet Definition XML-Datei (CDXML)	479
17	Dynamische Objekte	483
17.1	Erweitern bestehender Objekte	483
17.2	Komplett dynamische Objekte	485
18	Einbinden von C# und Visual Basic .NET	487
19	Win32-API-Aufrufe	489
20	Benutzereingaben	493
20.1	Read-Host	493
20.2	Benutzerauswahl	494
20.3	Grafischer Eingabedialog	495
20.4	Dialogfenster	496
20.5	Authentifizierungsdialog	496
20.6	Zwischenablage (Clipboard)	498
21	Fehlersuche	499
21.1	Detailinformationen	499
21.2	Einzelschrittmodus	500
21.3	Zeitmessung	501
21.4	Ablaufverfolgung (Tracing)	502
21.5	Erweiterte Protokollierung aktivieren	503

21.6	Script-Debugging in der ISE	505
21.7	Kommandozeilenbasiertes Script-Debugging	505
22	Transaktionen	507
22.1	Commandlets für Transaktionen	507
22.2	Start und Ende einer Transaktion	508
22.3	Zurücksetzen der Transaktion	509
22.4	Mehrere Transaktionen	510
23	Standardeinstellungen ändern mit Profilskripten	511
23.1	Profilpfade	511
23.2	Ausführungsreihenfolge	513
23.3	Beispiel für eine Profildatei	513
23.4	Starten der PowerShell ohne Profilskripte	514
24	Digitale Signaturen für PowerShell-Skripte	515
24.1	Zertifikat erstellen	515
24.2	Skripte signieren	517
24.3	Verwenden signierter Skripte	519
24.4	Mögliche Fehlerquellen	519
25	Hintergrundaufträge („Jobs“)	521
25.1	Voraussetzungen	521
25.2	Architektur	521
25.3	Starten eines Hintergrundauftrags	522
25.4	Hintergrundaufträge abfragen	523
25.5	Warten auf einen Hintergrundauftrag	524
25.6	Abbrechen und Löschen von Aufträgen	524
25.7	Analyse von Fehlermeldungen	524
25.8	Fernausführung von Hintergrundaufträgen	525
25.9	Praxislösung: Einen Job auf mehreren Computern starten	525
26	Geplante Aufgaben und zeitgesteuerte Jobs	527
26.1	Geplante Aufgaben (Scheduled Tasks)	527
26.2	Zeitgesteuerte Jobs	531
27	PowerShell-Workflows	537
27.1	Ein erstes Beispiel	537
27.2	Unterschiede zu einer Function bzw. einem Skript	542
27.3	Einschränkungen bei Workflows	542
27.4	Workflows in der Praxis	544
27.5	Workflows in Visual Studio erstellen	551

28	Ereignissystem	569
28.1	WMI-Ereignisse	569
28.2	WMI-Ereignisabfragen	569
28.3	WMI-Ereignisse seit PowerShell 1.0	571
28.4	Registrieren von WMI-Ereignisquellen seit PowerShell 2.0	572
28.5	Auslesen der Ereignisliste	573
28.6	Reagieren auf Ereignisse	575
28.7	WMI-Ereignisse ab PowerShell-Version 3.0	577
28.8	Registrieren von .NET-Ereignissen	577
28.9	Erzeugen von Ereignissen	578
29	Datenbereiche und Datendateien	581
29.1	Datenbereiche	581
29.2	Datendateien	583
29.3	Mehrsprachigkeit/Lokalisierung	584
30	Desired State Configuration (DSC)	587
30.1	Grundprinzipien	588
30.2	DSC für Linux	588
30.3	Ressourcen	589
30.4	Verfügbare DSC-Ressourcen	589
30.5	Eigenschaften einer Ressource	592
30.6	Aufbau eines DSC-Dokuments	592
30.7	Commandlets für die Arbeit mit DSC	593
30.8	Ein erstes DSC-Beispiel	593
30.9	Kompilieren und Anwendung eines DSC-Dokuments	594
30.10	Variablen in DSC-Dateien	596
30.11	Parameter für DSC-Dateien	597
30.12	Konfigurationsdaten	598
30.13	Entfernen einer DSC-Konfiguration	601
30.14	DSC Pull Server	604
30.15	DSC-Praxislösung 1: IIS installieren	611
30.16	DSC-Praxislösung 2: Software installieren	613
30.17	DSC-Praxislösung 3: Software deinstallieren	615
30.18	Realisierung einer DSC-Ressource	616
30.19	Weitere Möglichkeiten	616
31	PowerShell-Snap-Ins	617
31.1	Einbinden von Snap-Ins	617
31.2	Liste der Commandlets	621

32	PowerShell-Module	623
32.1	Überblick über die Commandlets	623
32.2	Modularchitektur	624
32.3	Aufbau eines Moduls	625
32.4	Module aus dem Netz herunterladen und installieren mit PowerShellGet	626
32.5	Module manuell installieren	632
32.6	Doppeldeutige Namen	633
32.7	Auflisten der verfügbaren Module	634
32.8	Importieren von Modulen	635
32.9	Entfernen von Modulen	638
33	Ausgewählte PowerShell-Erweiterungen	639
33.1	PowerShell-Module in Windows 8.0 und Windows Server 2012	640
33.2	PowerShell-Module in Windows 8.1 und Windows Server 2012 R2	642
33.3	PowerShell-Module in Windows 10 und Windows Server 2019	644
33.4	PowerShell Community Extensions (PSCX)	648
33.5	PowerShellPack	652
33.6	<i>www.IT-Visions.de</i> : PowerShell Extensions	654
33.7	Quest Management Shell for Active Directory	654
33.8	Microsoft Exchange Server	656
33.9	System Center Virtual Machine Manager	657
33.10	PowerShell Management Library for Hyper-V (pshyperv)	657
33.11	Powershell Outlook Account Manager	658
33.12	PowerShell Configurator (PSConfig)	659
33.13	Weitere Erweiterungen	660
34	Delegierte Administration/Just Enough Administration (JEA)	661
34.1	JEA-Konzept	661
34.2	PowerShell-Sitzungskonfiguration erstellen	661
34.3	Sitzungskonfiguration nutzen	664
34.4	Delegierte Administration per Webseite	666
35	Unit Tests mit Pester	667
35.1	Einführung in das Konzept des Unit Testing	667
35.2	Pester installieren	668
35.3	Befehle in Pester	668
35.4	Testen einer PowerShell-Funktion	669
35.5	Testgenerierung	670
35.6	Tests starten	671

35.7	Prüf-Operationen	672
35.8	Mock-Objekte	672
35.9	Test von Dateisystemoperationen	673
36	Tipps und Tricks zur PowerShell	677
36.1	Alle Anzeigen löschen	677
36.2	Befehlsgeschichte	677
36.3	System- und Hostinformationen	678
36.4	Anpassen der Eingabeaufforderung (Prompt)	679
36.5	PowerShell-Befehle aus anderen Anwendungen heraus starten	680
36.6	ISE erweitern	681
36.7	PowerShell für Gruppenrichtlinienskripte	682
36.8	Einblicke in die Interna der Pipeline-Verarbeitung	685
	Teil C: PowerShell im Praxiseinsatz	687
37	Dateisystem	689
37.1	Laufwerke	690
37.2	Ordnerinhalte	695
37.3	Dateieigenschaften verändern	699
37.4	Eigenschaften ausführbarer Dateien	700
37.5	Kurznamen	702
37.6	Lange Pfade	702
37.7	Dateisystemoperationen	703
37.8	Praxislösungen: Dateien umorganisieren	703
37.9	Praxislösung: Zufällige Dateisystemstruktur erzeugen	705
37.10	Praxislösung: Leere Ordner löschen	706
37.11	Einsatz von Robocopy	707
37.12	Dateisystemkataloge	710
37.13	Papierkorb leeren	711
37.14	Dateieigenschaften lesen	711
37.15	Praxislösung: Fotos nach Aufnahmedatum sortieren	712
37.16	Datei-Hash	713
37.17	Finden von Duplikaten	714
37.18	Verknüpfungen im Dateisystem	716
37.19	Komprimierung	721
37.20	Dateisystemfreigaben	724
37.21	Überwachung des Dateisystems	735
37.22	Dateiversionsverlauf	736
37.23	Windows Explorer öffnen	737
37.24	Windows Server Backup	737

38	Festplattenverschlüsselung mit BitLocker	739
38.1	Übersicht über das BitLocker-Modul	740
38.2	Verschlüsseln eines Laufwerks	741
39	Dokumente	743
39.1	Textdateien	743
39.2	CSV-Dateien	745
39.3	Analysieren von Textdateien	748
39.4	INI-Dateien	751
39.5	XML-Dateien	751
39.6	HTML- und Markdown-Dateien	761
39.7	JSON-Dateien	764
39.8	Binärdateien	774
39.9	Grafikdateien	775
40	Microsoft Office	777
40.1	Allgemeine Informationen zur Office-Automatisierung per PowerShell	777
40.2	Praxislösung: Terminserien aus Textdateien anlegen in Outlook	778
40.3	Praxislösung: Outlook-Termine anhand von Suchkriterien löschen	780
40.4	Praxislösung: Grafiken aus einem Word-Dokument (DOCX) extrahieren	781
41	Datenbanken	785
41.1	ADO.NET-Grundlagen	785
41.2	Beispieldatenbank	791
41.3	Datenzugriff mit den Bordmitteln der PowerShell	792
41.4	Hilfsfrountinen für den Datenbankzugriff (DBUtil.ps1)	804
41.5	Datenzugriff mit den PowerShell-Erweiterungen	807
41.6	Datenbankzugriff mit SQLPS	811
41.7	Datenbankzugriff mit SQLPSX	811
42	Microsoft-SQL-Server-Administration	813
42.1	PowerShell-Integration im SQL Server Management Studio	814
42.2	SQL-Server-Laufwerk „SQLSERVER:“	815
42.3	Die SQLPS-Commandlets	818
42.4	Die SQL Server Management Objects (SMO)	820
42.5	SQLPSX	823
42.6	Microsoft-SQL-Server-Administration mit der PowerShell in der Praxis	831
43	ODBC-Datenquellen	837
43.1	ODBC-Treiber und -Datenquellen auflisten	838
43.2	Anlegen einer ODBC-Datenquelle	839
43.3	Zugriff auf eine ODBC-Datenquelle	840

44	Registrierungsdatenbank (Registry)	843
44.1	Schlüssel auslesen	843
44.2	Schlüssel anlegen und löschen	844
44.3	Laufwerke definieren	844
44.4	Werte anlegen und löschen	845
44.5	Werte auslesen	846
44.6	Praxislösung: Windows-Explorer-Einstellungen	846
44.7	Praxislösung: Massenanlegen von Registry-Schlüsseln	847
45	Computer- und Betriebssystemverwaltung	849
45.1	Computerinformationen	849
45.2	Versionsnummer des Betriebssystems	851
45.3	Zeitdauer seit dem letzten Start des Betriebssystems	851
45.4	BIOS- und Startinformationen	852
45.5	Windows-Produktaktivierung	853
45.6	Umgebungsvariablen	853
45.7	Schriftarten	856
45.8	Computername und Domäne	857
45.9	Herunterfahren und Neustarten	857
45.10	Windows Updates installieren	858
45.11	Wiederherstellungspunkte verwalten	862
46	Windows Defender	863
47	Hardwareverwaltung	865
47.1	Hardwarebausteine	865
47.2	Plug-and-Play-Geräte	867
47.3	Druckerverwaltung (ältere Betriebssysteme)	867
47.4	Druckerverwaltung (seit Windows 8 und Windows Server 2012)	869
48	Softwareverwaltung	871
48.1	Softwareinventarisierung	871
48.2	Installation von Anwendungen	875
48.3	Deinstallation von Anwendungen	875
48.4	Praxislösung: Installationstest	876
48.5	Praxislösung: Installierte .NET Core SDKs aufräumen	877
48.6	Windows 10 Apps verwalten	883
48.7	Installationen mit PowerShell Package Management („OneGet“)	885
48.8	Versionsnummer ermitteln	888
48.9	Servermanager	890
48.10	Windows-Features installieren auf Windows-Clientbetriebssystemen	900

48.11	Praxislösung: IIS-Installation	903
48.12	Softwareeinschränkungen mit dem PowerShell-Modul „AppLocker“	905
49	Prozessverwaltung	911
49.1	Prozesse auflisten	911
49.2	Prozesse starten	912
49.3	Prozesse mit vollen Administratorrechten starten	913
49.4	Prozesse unter einem anderen Benutzerkonto starten	914
49.5	Prozesse beenden	915
49.6	Warten auf das Beenden einer Anwendung	916
50	Windows-Systemdienste	917
50.1	Dienste auflisten	917
50.2	Dienstzustand ändern	920
50.3	Diensteigenschaften ändern	920
50.4	Dienste hinzufügen	921
50.5	Dienste entfernen	922
51	Netzwerk	923
51.1	Netzwerkconfiguration	923
51.2	DNS-Client-Konfiguration	928
51.3	DNS-Namensauflösung	932
51.4	Erreichbarkeit prüfen (Ping)	933
51.5	Windows Firewall	934
51.6	Remote Desktop (RDP) einrichten	941
51.7	E-Mails senden (SMTP)	942
51.8	Auseinandernehmen von E-Mail-Adressen	943
51.9	Abruf von Daten von einem HTTP-Server	943
51.10	Praxislösung: Linkprüfer für eine Website	949
51.11	Aufrufe von SOAP-Webdiensten	953
51.12	Aufruf von REST-Diensten	955
51.13	Aufrufe von OData-Diensten	957
51.14	Hintergrunddatentransfer mit BITS	958
52	Ereignisprotokolle (Event Log)	963
52.1	Protokolleinträge auslesen	963
52.2	Ereignisprotokolle erzeugen	965
52.3	Protokolleinträge erzeugen	965
52.4	Protokollgröße festlegen	965
52.5	Protokolleinträge löschen	965

53	Leistungsdaten (Performance Counter)	967
53.1	Zugriff auf Leistungsindikatoren über WMI	967
53.2	Get-Counter	968
54	Sicherheitseinstellungen	971
54.1	Aktueller Benutzer	971
54.2	Grundlagen	972
54.3	Zugriffsrechtlisten auslesen	977
54.4	Einzelne Rechteeinträge auslesen	978
54.5	Besitzer auslesen	980
54.6	Benutzer und SID	980
54.7	Hinzufügen eines Rechteeintrags zu einer Zugriffsrechtliste	983
54.8	Entfernen eines Rechteeintrags aus einer Zugriffsrechtliste	985
54.9	Zugriffsrechtliste übertragen	987
54.10	Zugriffsrechtliste über SDDL setzen	987
54.11	Zertifikate verwalten	988
55	Optimierungen und Problemlösungen	991
55.1	PowerShell-Modul „TroubleshootingPack“	991
55.2	PowerShell-Modul „Best Practices“	995
56	Active Directory	997
56.1	Benutzer- und Gruppenverwaltung mit WMI	998
56.2	Einführung in System.Directory Services	999
56.3	Basiseigenschaften	1010
56.4	Benutzer- und Gruppenverwaltung im Active Directory	1012
56.5	Verwaltung der Organisationseinheiten	1020
56.6	Suche im Active Directory	1021
56.7	Navigation im Active Directory mit den PowerShell Extensions	1028
56.8	Verwendung der Active-Directory-Erweiterungen von www.IT-Visions.de	1029
56.9	PowerShell-Modul „Active Directory“ (ADPowerShell)	1031
56.10	PowerShell-Modul „ADDSDeployment“	1060
56.11	Informationen über die Active Directory-Struktur	1063
57	Gruppenrichtlinien	1067
57.1	Verwaltung der Gruppenrichtlinien	1068
57.2	Verknüpfung der Gruppenrichtlinien	1069
57.3	Gruppenrichtlinienberichte	1071
57.4	Gruppenrichtlinienvererbung	1073
57.5	Weitere Möglichkeiten	1074

58	Lokale Benutzer und Gruppen	1075
58.1	Modul „Microsoft.PowerShell.LocalAccounts“	1075
58.2	Lokale Benutzerverwaltung in älteren PowerShell-Versionen	1077
59	Microsoft Exchange Server	1079
59.1	Daten abrufen	1079
59.2	Postfächer verwalten	1080
59.3	Öffentliche Ordner verwalten	1081
60	Internet Information Services (IIS)	1083
60.1	Überblick	1083
60.2	Navigationsprovider	1085
60.3	Anlegen von Websites	1087
60.4	Praxislösung: Massenanlegen von Websites	1088
60.5	Ändern von Website-Eigenschaften	1091
60.6	Anwendungspool anlegen	1091
60.7	Virtuelle Verzeichnisse und IIS-Anwendungen	1092
60.8	Website-Zustand ändern	1093
60.9	Anwendungspools starten und stoppen	1093
60.10	Löschen von Websites	1094
61	Virtuelle Systeme mit Hyper-V	1095
61.1	Das Hyper-V-Modul von Microsoft	1096
61.2	Die ersten Schritte mit dem Hyper-V-Modul	1098
61.3	Virtuelle Maschinen anlegen	1102
61.4	Umgang mit virtuellen Festplatten	1108
61.5	Konfiguration virtueller Maschinen	1111
61.6	Dateien kopieren in virtuelle Systeme	1115
61.7	PowerShell Management Library for Hyper-V (für ältere Betriebssysteme)	1116
62	Windows Nano Server	1119
62.1	Das Konzept von Nano Server	1119
62.2	Einschränkungen von Nano Server	1121
62.3	Varianten des Nano Servers	1123
62.4	Installation eines Nano Servers	1123
62.5	Docker-Image	1124
62.6	Fernverwaltung mit PowerShell	1125
62.7	Windows Update auf einem Nano Server	1127
62.8	Nachträgliche Paketinstallation	1127
62.9	Abgespeckter IIS unter Nano Server	1129
62.10	Nano-Serververwaltung aus der Cloud heraus	1130

63	Docker-Container	1131
63.1	Docker-Varianten für Windows	1132
63.2	Docker-Installation auf Windows 10	1133
63.3	Docker-Installation auf Windows Server	1135
63.4	Installation von „Docker for Windows“	1136
63.5	Docker-Registries	1138
63.6	Docker-Images laden	1138
63.7	Container starten	1138
63.8	Container-Identifikation	1140
63.9	Container mit Visual Studio	1141
63.10	Befehle in einem Container ausführen	1143
63.11	Ressourcenbeschränkungen für Container	1146
63.12	Dateien zwischen Container und Host kopieren	1146
63.13	Dockerfile	1146
63.14	Docker-Netzwerke	1147
63.15	Container anlegen, ohne sie zu starten	1148
63.16	Container starten und stoppen	1148
63.17	Container beenden und löschen	1148
63.18	Images löschen	1149
63.19	Images aus Containern erstellen	1149
63.20	.NET Core-Container	1150
63.21	Images verbreiten	1152
63.22	Azure Container Service (ACS)	1154
64	Microsoft Azure	1155
64.1	Azure Konzepte	1155
64.2	Kommandozeilenwerkzeuge für die Azure-Verwaltung	1156
64.3	Benutzeranmeldung und Informationsabfrage	1160
64.4	Azure Ressourcen-Gruppen	1161
64.5	Azure Web Apps	1161
64.6	Azure SQL Server	1163
64.7	Azure Kubernetes Services (AKS)	1164
64.8	Azure DevOps	1188
65	Grafische Benutzeroberflächen (GUI)	1203
65.1	Einfache Nachfragedialoge	1203
65.2	Einfache Eingabe mit Inputbox	1205
65.3	Komplexere Eingabemasken	1205
65.4	Universelle Objektdarstellung	1207
65.5	WPF PowerShell Kit (WPK)	1209
65.6	Direkte Verwendung von WPF	1217

Teil D: Profiwissen – Erweitern der PowerShell	1219
66 Entwicklung von Commandlets in der PowerShell-Skriptsprache	1221
66.1 Aufbau eines skriptbasierten Commandlets	1221
66.2 Verwendung per Dot Sourcing	1223
66.3 Parameterfestlegung	1224
66.4 Fortgeschrittene Funktion (Advanced Function)	1230
66.5 Mehrere Parameter und Parametersätze	1234
66.6 Unterstützung für Sicherheitsabfragen (-whatif und -confirm)	1236
66.7 Kaufmännisches Beispiel: Test-CustomerID	1237
66.8 Erweitern bestehender Commandlets durch Proxy-Commandlets	1241
66.9 Dokumentation	1246
67 Entwicklung eigener Commandlets mit C#	1249
67.1 Technische Voraussetzungen	1250
67.2 Grundkonzept der .NET-basierten Commandlets	1251
67.3 Schrittweise Erstellung eines minimalen Commandlets	1253
67.4 Erstellung eines Commandlets mit einem Rückgabeobjekt	1262
67.5 Erstellung eines Commandlets mit mehreren Rückgabeobjekten	1263
67.6 Erstellen eines Commandlets mit Parametern	1268
67.7 Verarbeiten von Pipeline-Eingaben	1270
67.8 Verkettung von Commandlets	1273
67.9 Fehlersuche in Commandlets	1277
67.10 Statusinformationen	1280
67.11 Unterstützung für Sicherheitsabfragen (-whatif und -confirm)	1285
67.12 Festlegung der Hilfeinformationen	1287
67.13 Erstellung von Commandlets für den Zugriff auf eine Geschäftsanwendung	1292
67.14 Konventionen für Commandlets	1293
67.15 Weitere Möglichkeiten	1295
68 PowerShell-Module erstellen	1297
68.1 Erstellen eines Skriptmoduls	1297
68.2 Praxislösung: Umwandlung einer Skriptdatei in ein Modul	1299
68.3 Erstellen eines Moduls mit Binärdateien	1299
68.4 Erstellen eines Moduls mit Manifest	1300
68.5 Erstellung eines Manifest-Moduls mit Visual Studio	1307
69 Hosting der PowerShell	1309
69.1 Voraussetzungen für das Hosting	1310
69.2 Hosting mit PSHost	1311
69.3 Vereinfachtes Hosting seit PowerShell 2.0	1314

Anhang A: Crashkurs Objektorientierung	1317
Anhang B: Crashkurs .NET	1325
B.1 Was ist das .NET Framework?	1327
B.2 Was ist .NET Core?	1328
B.3 Eigenschaften von .NET	1329
B.4 .NET-Klassen	1330
B.5 Namensgebung von .NET-Klassen (Namensräume)	1330
B.6 Namensräume und Softwarekomponenten	1332
B.7 Bestandteile einer .NET-Klasse	1333
B.8 Vererbung	1334
B.9 Schnittstellen	1334
Anhang C: Literatur	1335
Anhang D: Weitere Informationen im Internet	1339
Anhang E: Abkürzungsverzeichnis	1341
Stichwortverzeichnis	1367