

Inhaltsverzeichnis

Einleitung.....	XVII
PowerShell installieren.....	XVIII
Die richtige PowerShell-Version herunterladen	XIX
PowerShell installieren (und deinstallieren)	XIX
Der »rote Faden« in diesem Buch	XXI
Allgemeiner Teil	XXI
Spezieller Teil: PowerShell in Aktion	XXIII
Nützliches PowerShell-Zubehör.....	XXIII
PowerShell-Hilfdatei.....	XXIII
Alternative PowerShell-Hosts und Editoren.....	XXIV
Grafische Werkzeuge.....	XXVII
Befehlsweiterungen.....	XXVII
Vorschau auf PowerShell Version 2.0	XXIX
Neue Funktionen.....	XXIX
1 Die PowerShell-Konsole	1
PowerShell starten.....	2
Erste Schritte in der Konsole	4
Unvollständige und mehrzeilige Eingaben.....	5
Wichtige Tastenkombinationen	7
Text markieren und einfügen.....	10
Die Konsole angenehm einrichten.....	11
Die Konsolen-Eigenschaften öffnen.....	12
Optionen festlegen	12
Schriftart und Schriftgröße festlegen.....	13
Fenster- und Puffergröße festlegen	14
Farben auswählen.....	15
Änderungen direkt in PowerShell festlegen.....	16
Änderungen speichern	16
Weiterleiten und Umlenken.....	17
Weiterleiten: Informationen seitenweise ausgeben	17
Umleitung: Informationen in Dateien speichern.....	18
Zusammenfassung.....	19
2 PowerShell interaktiv.....	21
PowerShell als Taschenrechner.....	22
Mit Zahlensystemen und Einheiten rechnen.....	23

Externe Befehle ausführen	26
Die »alte« Konsole starten	27
Nützliche Konsolenbefehle entdecken	27
Sicherheitseinschränkungen beim Programmstart	30
Vertrauenswürdige Ordner	31
Cmdlets – die echten PowerShell-Befehle	33
Parameter verwenden.....	36
Alias-Namen: Befehlen andere Namen geben	40
Aliasnamen auflösen	41
Eigene Alias-Namen erfinden.....	44
Alias entfernen – oder dauerhaft behalten.....	44
Alias-Definitionen überschreiben und von Hand löschen.....	46
Funktionen: »Erweiterte« Alias-Namen	46
Befehle mit vorgegebenen Argumenten	46
Kurzformen für gebräuchliche Befehle	47
Dateien und Skripte aufrufen.....	48
Skripte starten.....	49
Zusammenfassung	52
3 Variablen	55
Eigene Variablen.....	56
Variablennamen auswählen	57
Werte zuweisen und abrufen	57
Übersicht über verwendete Variablen	59
Spezielle Variablen-Cmdlets verwenden.....	61
»Automatische« PowerShell-Variablen.....	64
Umgebungsvariablen.....	67
Bestimmte Umgebungsvariablen lesen.....	67
Umgebungsvariablen suchen	68
Neue Umgebungsvariablen anlegen	68
Umgebungsvariablen löschen und ändern	69
Permanente Änderungen an Umgebungsvariablen	70
Direktvariablen.....	71
Direkt auf Dateipfade zugreifen	71
Ad-hoc-Variablen: die Unterausdrücke.....	73
Gültigkeitsbereich von Variablen.....	74
Automatische Abschirmung	74
Abschirmung durchbrechen	75
Vorteil Abschirmung: Eindeutige und klare Startverhältnisse	75
Gültigkeitsbereich einzelner Variablen festlegen.....	77
Variablentypen und strenge Typisierung.....	80
Feste Typen zuweisen.....	82

Variablenverwaltung hinter den Kulissen	85
Variablen-Optionen nachträglich ändern	85
Schreibschutz nachträglich aktivieren	86
Variablen-Typisierung	87
Variableninhalt prüfen und validieren	88
Zusammenfassung.....	91
4 Felder und Hashtables.....	93
PowerShell-Befehle liefern Felder	94
Ergebnisse in Feldern speichern	94
Feldelemente in der Pipeline weiterbearbeiten.....	95
Mit echten Objekten arbeiten.....	96
Neue Felder anlegen.....	97
Polymorphe Felder.....	98
Felder mit nur einem (oder keinem) Element	98
Feldelemente ansprechen	99
Mehrere Elemente aus einem Feld auswählen	100
Elemente zu einem Feld hinzufügen und daraus entfernen	101
Hashtables.....	102
Eine neue Hashtable anlegen	102
Felder in Hashtables speichern	103
Neue Schlüssel in eine bestehende Hashtable einfügen	104
Werte ändern und entfernen	105
Hashtables zur Ausgabeformatierung verwenden	105
Felder und Hashtables kopieren	106
Streng typisierte Felder	107
Zusammenfassung.....	108
5 Die PowerShell-Pipeline	109
Die PowerShell-Pipeline einsetzen.....	110
Objektorientierte Pipeline.....	110
Textumwandlung erst am Schluss	111
Streaming: Echtzeitbearbeitung oder nicht?	112
Blockierende Pipeline-Befehle.....	113
Objekte in Text verwandeln.....	115
Objekteigenschaften sichtbar machen	115
Pipeline-Ergebnisse formatieren.....	117
Pipeline-Ergebnisse sortieren und gruppieren.....	123
Sort-Object und Hashtables.....	124
Informationen gruppieren	124
Gruppen bilden mit den Formatierungs-Cmdlets	127
Pipeline-Ergebnisse filtern	128
Objekte aus der Pipeline herausfiltern	128
Objekteigenschaften auswählen	132

Objektanzahl begrenzen	133
Alle Ergebnisse der Pipeline einzeln bearbeiten.....	135
Doppelgänger entfernen	136
Ergebnisse analysieren und vergleichen	136
Statistische Berechnungen.....	137
Objekte vergleichen	137
Pipeline-Ergebnisse exportieren	144
Ergebnisse unterdrücken	144
Formatierungen in der Pipeline verändern	145
Textdarstellungen erzwingen.....	146
Excel: Objekte exportieren	147
HTML-Ausgaben.....	148
Das Extended Type System (Teil 1).....	149
Textdarstellung bleibt Textdarstellung.....	150
Ihr Wunsch hat Vorrang	150
Formatierung bei bekannten Objekten	151
Unbekannte Objekte	152
Der Fall der verschwundenen Spalte.....	153
Das Extended Type System erweitern	154
Zusammenfassung.....	157
6 Objekte verwenden.....	159
Objekte = Eigenschaften + Methoden	160
Ein neues Objekt anlegen	160
Eigenschaften hinzufügen	161
Methoden hinzufügen.....	162
Eigenschaften: Was das Objekt <i>ist</i>	163
Eigenschaften enthalten Objekte.....	165
Manche Eigenschaften lassen sich ändern	166
Welche Werte sind für Eigenschaften erlaubt?	167
Alle Eigenschaften eines Objekts in der Übersicht.....	170
Methoden: Was das Objekt <i>kann</i>	171
Überflüssige Methoden aussortieren	171
Eine Methode aufrufen	172
Eine Methode mit Argumenten aufrufen.....	173
Eine Methode mit mehreren »Signaturen«.....	175
Mit vorhandenen Objekten arbeiten.....	177
Ergebnisse in Variablen zwischenspeichern	177
Objekte aus Ergebnissen lesen	178
Die Verbindung zum .NET Framework.....	182
Typen, Klassen und Instanzen	185
Statische Methoden verwenden.....	186
Welche Klassen gibt es sonst noch?	192

Neue Objekte anlegen	196
Neue Objekte mit New-Object	196
Neue Objekte durch Umwandlung	198
Welche Klassen und Befehle gibt es sonst noch?.....	201
Weitere Assemblies nachladen: Besserer Internet-Download	201
Windows-Fenster erstellen	203
Type Acceleratoren einsetzen.....	204
Eigene Type Acceleratoren	205
Eigene Abkürzungen für statische Methoden.....	206
Visual Basic-Befehle nachrüsten	207
Die .NET-Umgebung verlassen	210
API-Aufrufe aus PowerShell.....	211
COM-Objekte verwenden.....	215
Das Extended Type System (Teil 2)	220
Ein ausgewähltes Objekt erweitern.....	220
Alle Objekte eines Typs erweitern	222
Exkurs: Doppelte Dateien finden.....	223
PropertySets – die besonderen Eigenschaften	226
Zusammenfassung.....	228
7 Bedingungen.....	229
Bedingungen formulieren	230
Einen Vergleich durchführen.....	231
Unterschiedliche Datentypen miteinander vergleichen	232
Vergleiche umkehren.....	234
Vergleiche kombinieren	235
Vergleiche mit Feldern	236
Where-Object	238
Ergebnisse in der Pipeline filtern	239
Eine Bedingung formulieren	240
Abkürzung verwenden	241
If...Elseif...Else	242
Switch.....	244
Wertebereiche testen	245
Keine zutreffende Bedingung.....	246
Mehrere zutreffende Bedingungen.....	246
Textvergleiche verwenden	247
Mehrere Werte gleichzeitig verarbeiten.....	249
Zusammenfassung.....	250
8 Schleifen.....	251
Foreach-Object	252
Pipeline-Objekte einzeln auswerten	252

Bedingungen integrieren	253
Methoden aufrufen	254
ForEach	255
Do und While	260
Fortsetzungs- und Abbruchbedingungen	260
Variablen als Fortsetzungskriterium verwenden	261
Endlosschleifen ohne Fortsetzungskriterium	262
For	262
For-Schleife: Nur Sonderform der While-Schleife	263
Ungewöhnliche Einsatzbereiche für die For-Schleife	264
Switch	265
Befehlszeilenparameter mit Switch auswerten	266
Dateiinhalte zeilenweise bearbeiten	269
Schleifen vorzeitig verlassen	270
Continue: Schleifendurchläufe überspringen	271
Verschachtelte Schleifen und Sprungmarken	272
Zusammenfassung	273
9 Funktionen	275
Neue Funktionen anlegen	276
Beispiel 1: Abkürzungsfunktionen	276
Beispiel 2: Mehrere Schritte zusammenfassen	277
Beispiel 3: Kapseln und Erweitern	280
Funktionen bearbeiten und verändern	280
Funktionen entfernen	282
Argumente an Funktionen übergeben	282
\$args – beliebige Argumente	283
Den Argument-Parser von \$args verwenden	286
Parameter festlegen	288
Argumente mit fest vorgegebenen Werten	291
Streng typisierte Argumente verwenden	292
Switch-Parameter funktionieren wie Schalter	295
Rückgabewerte einer Funktion festlegen	296
Ein Rückgabewert oder mehrere?	296
Die Return-Anweisung	297
Auf die Rückgabewerte zugreifen	299
Ausgaben aus dem Funktionsergebnis ausschließen	300
Vorhandene Funktionen untersuchen	305
Prompt: Eine bessere Eingabeaufforderung	306
Clear-Host: Den Bildschirmpuffer löschen	310
A:, B:, C: –vordefinierte Funktionen	312
Funktionen, Filter und die Pipeline	313
\$Input – langsamer sequentieller Modus	313

Filter: Schneller Streaming-Modus.....	315
Echte Pipeline-Funktionen entwickeln.....	315
Zusammenfassung.....	317
10 Skripte	319
PowerShell-Skripte verfassen und starten.....	320
Skripte per Umleitung anlegen	320
Skripte mit einem Editor anlegen	320
Skripte starten	321
Argumente an Skripte übergeben	324
\$args liefert alle Argumente.....	324
\$args ist ein Feld	325
Einzelne Argumente in \$args ansprechen	325
Parameter in Skripten einsetzen	326
Scopes: Gültigkeitsbereiche in Skripten.....	328
#requires – Skriptvoraussetzungen.....	330
Skripte übersichtlich gestalten.....	331
Funktionen in Skripten verwenden.....	332
Skripte in Arbeitsskripte und Bibliotheken trennen.....	333
Zentrale Ordner für Bibliotheksskripte	334
Pipeline-Skripte erstellen.....	336
Langsamer sequentieller Modus	336
Schneller Streaming-Modus	336
Pipeline-Ergebnisse schreiben.....	337
Profile: die Autostart-Skripte	339
Vier verschiedene Profil-Skripte.....	339
Ein eigenes persönliches Profil anlegen	340
Ein globales Profil für alle Benutzer anlegen.....	340
Digitale Signaturen für Ihre Skripte.....	341
Ein geeignetes Zertifikat finden	341
Das Codesigning-Zertifikat untersuchen	344
Zertifikate als vertrauenswürdig erklären.....	344
PowerShell-Skripte signieren.....	347
Signierte PowerShell-Skripte überprüfen	349
Eine Mini-PKI aufbauen	353
Zusammenfassung.....	357
11 Fehler finden und vermeiden	359
Was-wäre-wenn-Szenarien	360
Trockentraining: Vorgänge simulieren.....	360
Einzelbestätigung: Einzeln nachfragen	361
Fehlertoleranz festlegen.....	363
Fehler erkennen und darauf reagieren	364

Fehlerstatus in \$?.....	365
Traps verwenden.....	365
Error Records – Details zum Fehler.....	367
Error Records umleiten.....	368
Error Record(s) über den Parameter -ErrorVariable.....	370
Error Records über \$Error.....	371
Error Record über Traps.....	371
Exceptions verstehen.....	371
Bestimmte Ausnahmen behandeln.....	372
Eigene Ausnahmen auslösen.....	372
Fehler in Funktionen und Skripten abfangen.....	373
Code schrittweise ausführen: Haltepunkte.....	377
Tracing: Ausgeführte Anweisungen anzeigen lassen.....	380
Stepping: Code schrittweise ausführen.....	381
Zusammenfassung.....	382
12 CommandDiscovery und Skriptblöcke.....	383
Command Discovery.....	384
Der Call-Operator »&&<<.....	386
Der Call-Operator akzeptiert nur einzelne Befehle.....	386
Der Call-Operator führt CommandInfo-Objekte aus.....	387
Gleichnamige Befehle: Welcher wird ausgeführt?.....	388
Skriptblöcke verwenden.....	391
Ganze Befehlszeilen ausführen.....	391
Aufbau von Skriptblöcken.....	394
Der ExecutionContext.....	397
InvokeCommand.....	397
SessionState.....	399
Zusammenfassung.....	401
13 Texte und reguläre Ausdrücke.....	403
Texte definieren.....	404
Sonderzeichen in Texten.....	404
Here-Strings: Mehrzeiligen Text erfassen.....	406
Mit dem Anwender kommunizieren.....	407
Spezielle Textbefehle verwenden.....	409
Textoperatoren.....	409
Methoden des String-Objekts.....	416
Befehle aus der String-Klasse verwenden.....	419
Einfache Textmustererkennung.....	421
Reguläre Ausdrücke.....	423
Muster beschreiben.....	423
Nach verschiedenen Begriffen gleichzeitig suchen.....	427
Groß- und Kleinschreibung unterscheiden.....	428

Informationen in Texten finden	430
Nach mehreren Schlüsselwörtern suchen	432
Gruppen bilden.....	433
Ergebnisse von Unterausdrücken weiterverwenden.....	435
Gierig oder faul? Ausführliche oder kurze Ergebnisse	436
Wortbereiche finden	438
Texte ersetzen	438
Rückverweise verwenden	439
Zeichen am Zeilenanfang voranstellen	440
Überflüssige Leerzeichen entfernen	441
Doppelte Wörter finden und entfernen.....	441
Zusammenfassung.....	441
14 XML.....	443
Aufbau von XML.....	444
XML-Dateien laden und verarbeiten.....	445
Auf einzelne Knoten zugreifen und Daten ändern	446
Knoten mit SelectNodes() auswählen	446
Auf Attribute zugreifen	448
Neue Knoten hinzufügen	448
Das Extended Type System untersuchen	449
Die XML-Daten des Extended Type Systems	449
Vordefinierte Ansichten finden	450
15 Dateisystem.....	455
Auf Dateien und Ordner zugreifen.....	457
Ordnerinhalte auflisten	457
Rekursiv das gesamte Dateisystem durchsuchen	458
Datei- und Ordnerinhalte beschaffen	460
Dateien an Cmdlets, Funktionen oder Skripte weitergeben.....	461
Im Dateisystem navigieren.....	467
Relative und absolute Pfade	467
Ordnerpositionen speichern.....	470
Spezial-Ordner finden	470
Pfadnamen konstruieren.....	473
Mit Dateien und Ordnern arbeiten.....	474
Neue Ordner anlegen.....	474
Neue Dateien anlegen.....	475
Neue Laufwerke anlegen	477
Den Inhalt von Textdateien lesen	478
Kommaseparierte Listen verarbeiten	478
Ersetzungen in Textdateien vornehmen.....	479
Textinhalte miteinander vergleichen	481

Textinhalte parsen und Informationen extrahieren	482
Binäre Inhalte lesen	484
Dateien und Ordner verschieben und kopieren	486
Dateien und Ordner umbenennen.....	487
Dateien und Ordner löschen.....	489
Berechtigungen verwalten	491
Effektive Sicherheitseinstellungen überprüfen.....	492
Neue Berechtigungen anlegen	493
16 Die Registrierungsdatenbank.....	503
Provider: Orte außerhalb des Dateisystems	505
Verfügbare Provider	505
Laufwerke anlegen	506
Registrierungsdatenbank durchsuchen	508
Rekursive Suche	508
Einzelne Registrierungsdatenbank-Schlüssel	509
Wie PowerShell Registrierungsdatenbank-Schlüssel anspricht.....	510
Anlegen und Löschen von Schlüsseln und Werten.....	513
Schlüssel mit Inhalten löschen.....	514
Werte eines Schlüssels setzen, ändern und löschen.....	515
Beispiel: Kontextmenü erweitern	519
Berechtigungen in der Registrierungsdatenbank.....	521
17 Prozesse, Dienste, Ereignislogbuch.....	529
Prozesse	530
Prozesse starten.....	530
Prozesse überwachen.....	531
Auf Process-Objekte zugreifen	534
Prozesse beenden	534
Dienste.....	535
Dienste auflisten.....	535
Dienste starten, stoppen, anhalten oder fortsetzen.....	535
Ereignislogbuch.....	537
Einträge ins Ereignislogbuch schreiben	539
18 WMI – der Windows-Verwaltungsdienst	541
WMI-Klassen und -Instanzen	542
Instanzen einer Klasse.....	542
Bestimmte Instanzen auswählen	544
Instanzen direkt ansprechen	545
Eigenschaften ändern.....	547
Klassenbeschreibung sichtbar machen.....	548

WMI-Methoden aufrufen	548
Instanzbasierte Methoden.....	548
Statische Methoden.....	550
Hilfe zu Klassen und Methoden.....	551
WMI-Ereignisse	552
Remotenzugriff und Namespaces	553
Auf WMI-Objekte eines anderen Computers zugreifen.....	553
Namespaces – WMI-Erweiterungen	554
WMI und Extended Type System	555
WMI-Datumsformat umwandeln	556
Einen Typkonverter nachrüsten	556
19 Benutzerverwaltung.....	561
Mit einer Domäne verbinden	562
Unter anderem Benutzernamen anmelden.....	562
Auf einen Container zugreifen	564
Den Inhalt eines Containers auflisten.....	565
Auf einzelne Benutzer oder Gruppen zugreifen	566
Eigenschaften lesen und ändern.....	573
Welche Eigenschaften gibt es überhaupt?	573
Eigenschaften lesen	577
Eigenschaften ändern	578
Eigenschaften löschen.....	579
Das Schema der Domäne	581
Eigenschaften setzen, die mehrere Werte haben	583
Methoden aufrufen	583
Kennwort ändern	585
Gruppenmitgliedschaften regeln	585
Neue Objekte anlegen.....	587
Neue Organisationseinheiten anlegen	587
Neue Gruppen anlegen.....	588
Neue Benutzer anlegen.....	588
Anhang A – Kurz-Referenz	591
Stichwortverzeichnis	601