

**Kapitel 1: Grundlagen und erste Schritte**

1.1	Erste Schritte	11
1.1.1	Hilfe in der PowerShell aufrufen	13
1.1.2	Daten abfragen	13
1.1.3	Skripte in der PowerShell erstellen	15
1.1.4	Variablen in der PowerShell verwenden	17
1.1.5	PowerShell über das Netzwerk nutzen	17
1.1.6	Dateioperationen in der PowerShell	18
1.1.7	Systemadministration über Cmdlets	20
1.1.8	Wiederherstellungspunkte nutzen	22
1.1.9	Daten sichern	22
1.1.10	Virtuelle und physische Festplatten verwalten	23
1.1.11	Druckerverwaltung mit der PowerShell	25
1.1.12	Serverrollen und Features verwalten	26
1.1.13	Voraussetzungen für AD-Betrieb testen	27
1.1.14	Desired State Configuration	28
1.1.15	SQL Server 2012 administrieren	32
1.1.16	PowerShell-eigene Laufwerke	33
1.2	Objekte in der PowerShell	36
1.2.1	Arbeiten mit Objekten	36
1.2.2	Detailinformationen der Objekte	37
1.2.3	Noch gezielter, noch detaillierter	38
1.2.4	Daten weiterverarbeiten	40
1.3	PowerShell-Schleifen	42
1.3.1	Bekannt aus Basic: Die For-Schleife	42
1.3.2	Das ist anders bei Foreach und Foreach-Object	44
1.3.3	Schleifen für die Ewigkeit: Do und While	47
1.3.4	Schleifen-Exit: Die continue- und break-Anweisung	49
1.4	Reguläre Ausdrücke und Metazeichen in der PowerShell	50
1.4.1	PowerShell mit Regex-Einsatz	51
1.4.2	Von Suchmustern und Metazeichen	53
1.4.3	Zeichen, Zahlen und die Anzahl davon ermitteln	54
1.4.4	Eingaben prüfen	57
1.5	PowerShell-Skripte optimieren	59
1.5.1	Grundlegende Prinzipien für unternehmenskritische Skripte	60
1.5.2	Stabilen Skript-Rahmen bauen	61
1.5.3	Quelltext sauber formatieren	62

1.5.4	Normalized Skript Notation nutzen	62
1.5.5	Fallstricke bei der Benennung von Funktionen	65
1.5.6	Software ohne Dokumentation ist wertlos	65
1.5.7	Ausnahmen behandeln, nicht Fehler	66
1.6	Kostenlose Editoren	67
1.6.1	Gewöhnungsbedürftig: Emacs und Vim	68
1.6.2	Der Klassiker: Notepad++	69
1.6.3	Der Unbekannte: PSPad	69
1.6.4	Plattformübergreifend: jEdit	71
1.6.5	Geheimtipp: SynWrite	72
1.6.6	Ein Muss für Mac-Anwender: TextWrangler	74
1.6.7	Multi-Plattform-Code Editor: Visual Studio Code	75
1.7	PowerShell Core 6.0	78
1.7.1	Alles neu mit .NET Core	79
1.7.2	Installation von PSCore	80
1.7.3	Die PowerShell unter Linux	80
1.7.4	Die Krux mit dem Alias	82
1.7.5	Gründe für die PowerShell auf Linux	83
1.7.6	Remotezugriffe alt und neu	83
1.8	Smartes Scripting nach dem DTAP-Modell	84
1.8.1	Vierstufiges Konzept	84
1.8.2	Herausforderungen in der Praxis	85
1.8.3	Scripting in der Praxis	86
1.8.4	Wiederverwertung erwünscht	87
1.8.5	Auswirkungen auf die IT-Praxis	88
<b>Kapitel 2: Systemverwaltung</b>		
2.1	PowerShell in KMUs	91
2.1.1	Vielseitige Automatisierungsmöglichkeiten	91
2.1.2	Drucker installieren	91
2.1.3	Clouddienste verwalten	93
2.1.4	Übersicht der Firewallbefehle	96
2.1.5	Effiziente Verarbeitung durch Workflows	97
2.1.6	Alternative: Desired State Configuration	99
2.2	Clientverwaltung in der Domäne mit der PowerShell	102
2.2.1	Administration via CSV-Datei	102
2.2.2	Vertrauen herstellen	103

---

2.2.3	Druckerverwaltung	104
2.2.4	Ermitteln der Gruppenmitgliedschaft	105
2.2.5	DNS-Einträge setzen	107
2.2.6	Windows-Features installieren	108
2.3	Skripte in Gruppenrichtlinien nutzen	108
2.3.1	Klassische Anmeldeskripte weiternutzen	109
2.3.2	Skripte in Gruppenrichtlinien integrieren	110
2.3.3	Kombinieren und parallel ausführen	112
2.3.4	Einstellungen in den Gruppenrichtlinien steuern	112
2.3.5	Loopbackverarbeitung von Gruppenrichtlinien	112
2.3.6	Sicherheit in der PowerShell	114
2.3.7	Fehlerbehebung mit Group Policy Log	114
2.3.8	Remoterechner im Blick	116
2.4	Virtuelle Maschinen mit PowerShell Direct administrieren	118
2.4.1	Der klassische Weg in die VM	119
2.4.2	Direkter Weg mit PowerShell Direct	120
2.4.3	Zugang über VMBus	120
2.4.4	Zwei neue, aber mächtige Parameter	121
2.4.5	Vielfältige Einsatzgebiete	123
2.4.6	Freiwilliger Verzicht	124
2.5	Hyper-V- und vSphere-VMs per Skript sichern	125
2.5.1	PowerCLI einrichten	125
2.5.2	Snapshots mit der PowerCLI verwalten	126
2.5.3	VMs mit Zusatzskript sichern	126
2.5.4	PowerShell für Hyper-V	128
2.5.5	Hyper-V-Snapshots scripten	129
2.5.6	PowerShell Direct nutzen	130
2.6	Datenbankzugriffe mit der PowerShell	131
2.6.1	ADO.NET als Grundlage für DB-Zugriffe	131
2.6.2	Hilfreiche Hilfsroutinen	135
2.7	Oracle-Datenbanken per PowerShell ansprechen	138
2.7.1	Oracle Data Provider mit der PowerShell nutzen	139
2.7.2	Zugriffs-DLL einspielen	140
2.7.3	Alternative Verbindung zur Datenbank aufbauen	141
2.7.4	Zugriff auf Oracle-Datenbanken	142
2.7.5	Erweiterungen der PowerShell für Oracle-Datenbanken	144

**Kapitel 3: Sicherheit**

3.1	Webserver unter Windows Server verwalten und absichern	147
3.1.1	Delegieren der IIS-Verwaltung	147
3.1.2	Einschränkungen festlegen	148
3.1.3	Konfigurieren von HTTP-Umleitungen	149
3.1.4	Tausendsassa PowerShell DSC	150
3.1.5	Absicherung mit Desired State Configuration	151
3.1.6	Konfigurationsdatei erstellen	152
3.1.7	MOF-Datei erstellen	155
3.1.8	MOF-Dateien ausführen, DSC aktivieren	155
3.1.9	Änderungen an der Konfiguration entdecken	156
3.1.10	Konfigurationsdateien erweitern	157
3.1.11	Dienste mit DSC steuern	157
3.2	Security-Module für die PowerShell	159
3.2.1	Windows-Systeme testen	159
3.2.2	Unbekannte Freigaben finden	160
3.2.3	Sichere Passwörter	160
3.2.4	DLLs manipulieren	161
3.2.5	Forensische Analyse nach Angriffen	161
3.3	Ransomware mit der PowerShell aufspüren	162
3.3.1	Grundlegende Schutzmaßnahmen	163
3.3.2	Skript erkennt Ransomware	164
3.3.3	Einsatz im Produktivbetrieb	166
3.4	PowerShell im IT-Support	166
3.4.1	Effizienz für Teams und Kunden	167
3.4.2	PowerShell schafft Abhilfe	168
3.4.3	Alles unter einem Dach	168
3.4.4	Monitoring automatisieren	169
3.5	Sicherheit mit der PowerShell 5	170
3.5.1	Adminrechte beschränken	171
3.5.2	Sprachbeschränkung mit \$ExecutionContext	172
3.5.3	Aktionen protokollieren	172
3.5.4	Schwachstellentests durchführen	173