PowerShell 7.4

Grundlagen und Verwaltung des Active Directory

Andreas Dittfurth

1. Ausgabe, März 2024

ISBN 978-3-98569-165-4

POSH74

Plus

HERDT

Bevor Sie beginnen …

HERDT BuchPlus – unser Konzept:

Problemlos einsteigen – Effizient lernen – Zielgerichtet nachschlagen

Nutzen Sie dabei unsere maßgeschneiderten, im Internet frei verfügbaren Medien:

Wie Sie schnell auf diese BuchPlus-Medien zugreifen können, erfahren Sie unter *www.herdt.com/BuchPlus*

Hinweise zu Soft- und Hardware

Die Windows PowerShell ist fest in die modernen Betriebssysteme Windows 10/11 und Windows Server 2019/2022 integriert. Ihr Rechner muss in der Lage sein, mit den Betriebssystemen zu arbeiten bzw. diese Betriebssysteme im Rahmen einer Virtualisierungssoftware zu betreiben.

Alle im Buch vorgestellten Aktionen können Sie ohne weitere Software durchführen. Damit Sie alle vorgestellten Übungen mit der PowerShell direkt an Ihrem Rechner durchführen können, benötigen Sie die folgenden Betriebssysteme:

- Windows 10 Enterprise oder Windows 11 Enterprise, Download unter *https://www.microsoft.com/de-de/evalcenter*
- Windows Server 2019 oder Windows Server 2022, Download unter *https://www.microsoft.com/de-de/evalcenter*
- Optional: Oracle VirtualBox (freie Virtualisierungssoftware, falls Sie Hyper-V nicht einsetzen können oder wollen), Download unter *https://virtualbox.org/wiki/Downloads*

Auch wenn Sie bereits die genannten Betriebssysteme einsetzen, ist der Aufbau einer virtuellen Testumgebung empfehlenswert, damit Sie nicht in Ihrer Produktivumgebung testen müssen. Die PowerShell ist sehr mächtig, deshalb ist in der Lernphase eine geschützte Testumgebung vorzuziehen. Installations- und Konfigurationshinweise zur Testumgebung mit Microsoft Hyper-V finden Sie in unter den ergänzenden Lerninhalten.

Typographische Konventionen

Kursivschrift: alle von Programmen vorgegebenen Bezeichnungen (Schaltflächen, Dialogfenster, Symbolleisten, Menüs und Menüpunkte (z. B. *Datei - Schließen*) sowie alle vom Anwender zugewiesene Namen. Schriftart `Courier New`: Programmtext und -namen, Funktions- und Variablennamen, Datentypen, Operatoren, zitierte Programmausgaben etc. *`Courier New`*: Kursive Passagen in dieser Schriftart kennzeichnen optionalen Code. Bei Darstellungen der Syntax einer Programmiersprache sowie PowerShell-Sprachelementen kennzeichnen eckige Klammern [ ] optionale Angaben. Alternative Elemente in der Syntax werden durch einen Schrägstrich / voneinander getrennt, spitze Klammern <> kennzeichnen Platzhalter.

1

Die PowerShell kennenlernen

1.1 Grundlagen zur PowerShell

Was ist die PowerShell?

Die Entwicklung der PowerShell beginnt mit der *Windows PowerShell*, einer jungen Befehlszeilenkonsole von Microsoft und Skriptsprache für die System- und Netzwerkverwaltung und -automatisierung. Sie wurde erstmalig Ende 2006 mit dem Microsoft Exchange Server 2007 ausgeliefert.

Microsoft bezeichnet die Windows PowerShell als Basis des Betriebssystems. Administrative Aufgaben, die Sie in der grafischen Oberfläche durchführen, werden im Hintergrund in PowerShell-Befehle übersetzt. Das Konzept wurde konsequent umgesetzt, sodass die administrativen Aufgaben mithilfe der Windows PowerShell durchgeführt werden können. Das ist vor allem dann interessant, wenn es keine grafische Alternative zur Durchführung der Aufgabe gibt oder Sie eine Automatisierung anstreben.

Die Windows PowerShell arbeitet objektorientiert und basiert auf dem Microsoft .NET-Framework. Es handelt sich trotz ähnlichen Aussehens nicht um eine Weiterentwicklung der Windows-Eingabeaufforderung (*cmd.exe*), sondern basiert auf einer anderen Technik. Die Windows PowerShell nimmt Konzepte von Unix-Shells auf, arbeitet mit Befehlsverbindungen und Filtern, bietet aber auch die Möglichkeit, eigene Skripte zu schreiben und damit Abläufe zu automatisieren. Letztlich folgt die Windows PowerShell mit ihrer integrierten Skriptumgebung (ISE) in erster Linie dem Ansatz der objektorientierten Programmierung.

Windows PowerShell-Versionen

Windows Server 2008 war das erste Betriebssystem, das mit der PowerShell Version 1 ausgeliefert wurde. Nachträglich wurden Downloads für Windows Vista, Windows XP und Windows Server 2003 angeboten. Seit dieser Zeit ist die PowerShell in jedem veröffentlichten Betriebssystem von Microsoft enthalten.

Windows PowerShell Version 2 wurde gegenüber der Version 1 funktional stark erweitert und ist vorinstalliert in den Betriebssystemen Windows 7 und Windows Server 2008 R2.

Windows PowerShell Version 3 ist automatisch in den Editionen der Betriebssysteme Windows 8 und Windows Server 2012 enthalten und damit bei beiden Betriebssystemen identisch. Die Windows PowerShell Version 3 steht als Teil des Windows Management Frameworks 3.0 für ältere Betriebssysteme als Download zur Verfügung.

Windows PowerShell Version 4 wurde im Oktober 2013 als finale Version veröffentlicht. Analog zu Version 3 ist die Windows PowerShell Version 4 bereits in allen Editionen der Betriebssysteme Windows 8.1 und Windows 2012 R2 enthalten (und Bestandteil des Windows Management Frameworks 4.0).

Die Version 5.0 der Windows PowerShell wurde ab April 2014 in einer Vorabversion des Management Frameworks 5.0 veröffentlicht. Wie bei den Vorgängerversionen ist Windows PowerShell 5 in allen Editionen aktueller Windows-Betriebssysteme enthalten – hier Windows 10 und Windows Server 2016.

Das im Juli 2015 erschienene Betriebssystem Windows 10 enthält die Version 5.0 der Windows PowerShell. Nach Installation des **Windows 10 Anniversary Updates** wird die Windows PowerShell auf Version 5.1 aktualisiert. Seit Windows Server 2016 (ab Oktober 2016) wird bis zum heutigen Tage in Windows-Betriebssystemen die Windows PowerShell Version 5.1 vorinstalliert als Teil des Betriebssystems ausgeliefert.

Zwischenzeitlich sorgten Meldungen für Verunsicherung, dass die Windows PowerShell nicht weiterentwickelt werde. Nutzer, denen die Windows PowerShell mittlerweile ans Herz gewachsen war, befürchteten, dass dies eine Abkehr von der PowerShell bedeuten würde. Dies ist keineswegs der Fall. Wer sich die Äußerungen von Microsoft genauer anschaut, bemerkt schnell, dass Microsoft etwas anderes im Sinn hat: Die Entwicklung der PowerShell als neue Sprache wird als abgeschlossen betrachtet. Es wird als vorinstallierte Komponente in Windows-Betriebssystemen keine andere Version mehr geben als die Version 5.1. Der Funktionsumfang der Windows PowerShell orientiert sich ohnehin primär an den installierten Komponenten Ihres Rechners und zusätzlich installierten Modulen aus dem Internet.

PowerShell Core (Version 6.x)

Nach Abschluss der Arbeiten für die so gut in Windows angebundene und auf .NET Framework basierende Windows PowerShell überraschte Microsoft mit einer PowerShell-Variante, die unter der Bezeichnung PowerShell Core 6 veröffentlicht wurde. Hier handelt es sich um keine Weiterentwicklung der Windows PowerShell, sondern um eine plattformübergreifende Open Source-Variante für Windows, macOS und Linux, die auf .NET Core basiert.

PowerShell (Version 7.x)

Im Jahr 2020 erschien PowerShell 7. Das Suffix Core wurde aus dem Namen gestrichen. Die Kompatibilität zu in Windows vorhandenen PowerShell-Modulen wurde verbessert. Mit der Weiterentwicklung der Version 7 (7.1, 7.2, 7.3 und aktuell 7.4 [Oktober 2023]) die Kompatibilität zur Windows PowerShell weiter verbessert, sodass es sich mittlerweile auch innerhalb der Windows-Welt lohnt, komplett von der Windows PowerShell (5.1) auf die PowerShell (7.4) umzusteigen.

Somit hat sich die PowerShell von einem reinen Windows-Werkzeug zu einem plattformübergreifenden Open-Source-Projekt entwickelt, das seit der Veröffentlichung von PowerShell Core auch auf macOS- und Linux-Systemen läuft. Mit der Version 7 haben sich die Beteiligten das Ziel gesetzt, die Welten zusammenzuführen. Das macht die neue PowerShell auch für reine Windows-Anwender sehr interessant.

Im vorliegenden Buch werden sowohl die vorinstallierte Windows PowerShell 5.1 und die PowerShell 7.4 verwendet und verglichen. Zum Lernen und Verstehen der Grundlagen spielt es größtenteils keine Rolle, ob Sie mit der vorinstallierten Windows PowerShell arbeiten und üben oder mit der neuen PowerShell 7.

1.2 Windows PowerShell 5.1 vs. PowerShell 7.x

Ok, Windows PowerShell ist älter, die letzte Version ist 5.1. PowerShell ist neuer, die Versionszählung beginnt bei Version 6.0. Ist dann nicht alles klar? Vorgänger vs. Nachfolger, also nimmt man die neue Version, und alles ist gut.

So einfach ist es nicht. Zum Lernen und Vertrautmachen mit der PowerShell ist es gleich, welche PowerShell Sie verwenden. Oder Sie nehmen einfach beide, um Verhalten und Aussehen zu vergleichen. Aber lassen Sie uns beide Produkte vergleichen.

- Windows PowerShell basiert auf dem .NET-Framework von Microsoft und ist dadurch hervorragend in die Betriebssysteme von Microsoft integriert.
- PowerShell basiert auf dem freien und quelloffenen .NET (ehemals .NET Core). Es werden auch windowsfremde Architekturen wie macOS und verschiedene Linux-Distributionen unterstützt.
- Windows PowerShell: solide und komplette Sprachbasis. Es bietet sehr gute Leistung für die Verwaltung und Automatisierung von Windows-Systemen.
- PowerShell: bietet einige zusätzliche Funktionen, Erweiterungen und Leistungsverbesserungen. Die Stärken der neuen PowerShell liegen im Umgang mit großen Datenmengen und parallelen Vorgängen.
- Windows PowerShell ist vollständig kompatibel mit vorhandenen Skripten und Tools, die in früheren Windows PowerShell-Versionen entwickelt wurden.
- PowerShell ist zwar abwärtskompatibel mit Windows PowerShell 5.x, es gibt jedoch Unterschiede und mögliche Inkompatibilitäten. Manche Befehle funktionieren geringfügig anders oder zeigen Unterschiede in der Syntax.
- ***Fazit**: Für Spezialanwendungen tief in Microsoft-Betriebssystemen hat die Windows PowerShell noch immer die Nase vorn. PowerShell wird besser, holt durch Beseitigung von Inkompatibilitäten auf.*

  Wer rein mit Windows-Systemen arbeitet, ist mit einer Mischung aus beiden Produkten gut beraten, da neueste Entwicklungen und Befehle die PowerShell voraussetzen, da für die Windows PowerShell nicht mehr entwickelt wird.

  Wer rechenintensive Aufgaben erledigt oder plattformübergreifend arbeitet, verwendet PowerShell 7.x.

Beachten Sie, dass PowerShell 7.x nicht als vollständiger Ersatz für Windows PowerShell 5.x betrachtet werden sollte. Die Auswahl der geeigneten PowerShell-Version sollte unter Berücksichtigung der spezifischen Anforderungen eigener Projekte und der eigenen technischen Umgebung erfolgen.

1.3 PowerShell-Konsolen und grafische PowerShell-Umgebungen

Die Windows PowerShell 5.1 steht als interaktive Eingabeaufforderung (Konsole; *powershell.exe*) und als Version mit integrierter Skriptumgebung (PowerShell Integrated Scripting Environment; *powershell_ise.exe*) im Windows-Betriebssystem zur Verfügung.

PowerShell 7 muss erst installiert werden und steht dann als Konsole (*pwsh.exe*) zur Verfügung. Bevorzugen Sie eine grafisch orientiertere Entwicklungsumgebung, steht z. B. Visual Studio Code von Microsoft kostenlos als Download zur Verfügung.

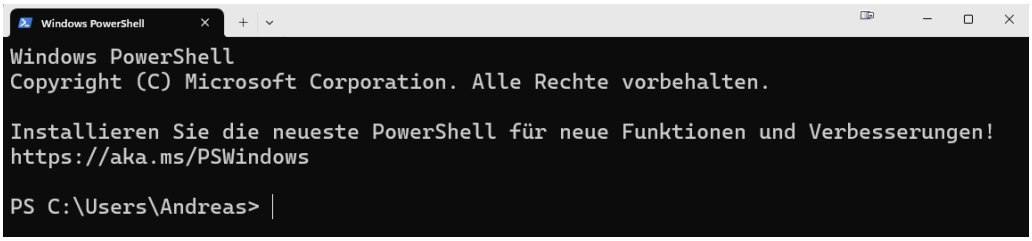
Microsoft zeigt in Anleitungen ausführlich die verschiedenen Möglichkeiten einer Installation von PowerShell 7.x (*https://learn.microsoft.com/de-de/powershell/scripting/install/installing-powershell-on-windows*).

Die PowerShell-Konsole

Die Konsole ähnelt stark den bekannten DOS-Fenstern. Bis auf den weiter unten ausführlich erläuterten QuickEdit-Modus, der das Kopieren und Einfügen von Textbausteinen erleichtert, begegnen Ihnen Aussehen und Handhabung der Windows-Eingabeaufforderung.

Bei einer Anwendung, die Texteingaben von Ihnen erwartet, ist das nicht weiter verwunderlich. Im Laufe der Kapitel werden Sie erkennen, dass ein erster Eindruck täuschen kann und in der PowerShell weit mehr steckt.

In Windows 11 öffnen sich alle Konsolenanwendungen – wie alle PowerShell-Versionen – in der neuen App *Terminal* in eigenen Registerkarten. Für Windows 10 können Sie die Terminal-App nachinstallieren.



PowerShell-Konsolenfenster unter Windows 11 im Windows Terminal

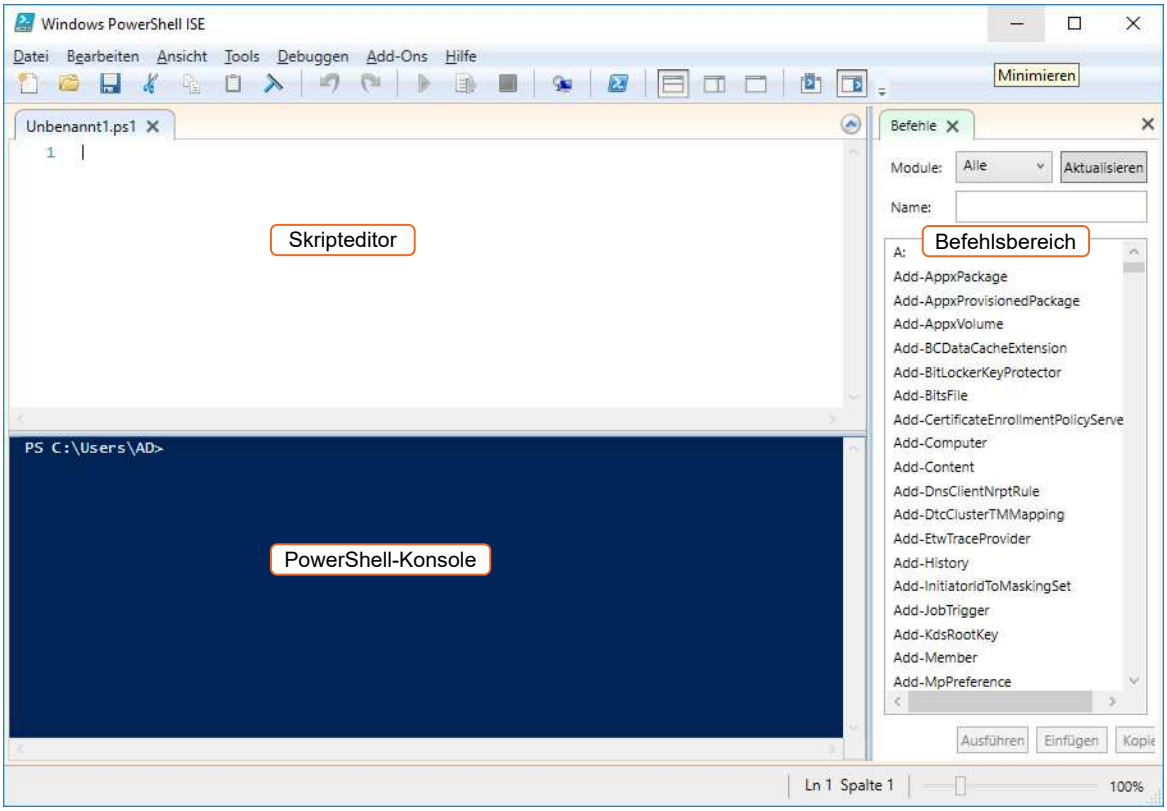
Wofür die PowerShell-Konsole eingesetzt wird

Die PowerShell-Konsole wird häufig für das schnelle Erledigen einer Aufgabe verwendet. Im Vordergrund steht der interaktive Einsatz: Sie geben einen Befehl ein, der sofort ausgeführt wird und – sofern vorhanden – umgehend die entsprechenden Informationen zurückgibt. In der Konsole lassen sich auch Skripte ausführen.

Im Alltag ist sie das primäre Werkzeug als „echte“ Konsole, die wenig Speicherplatz benötigt und ohne grafische Zusätze auskommt.

PowerShell Integrated Scripting Environment (ISE)

PowerShell ISE ist ein Skripteditor, den Microsoft seit PowerShell 2.0 mitliefert. In der PowerShell-Version 3.0 wurde PowerShell ISE stark überarbeitet und mit nützlichen Vereinfachungen für den Anwender ausgestattet. Die PowerShell ISE 5 weist im Vergleich zu ihren Vorgängern ab Version 3 nur marginale Änderungen auf. Eine interaktive Arbeit mit PowerShell ISE ist durch die integrierte Konsole ebenfalls möglich.



PowerShell ISE-Anwendungsfenster

Nach dem Start sehen Sie im kompletten linken Bereich des Fensters die integrierte PowerShell-Konsole sowie den Skripteditor. Im obigen Bild wurde der standardmäßig ausgeblendete Skripteditor bereits ausgeklappt. Im rechten Fensterbereich sehen Sie den Befehlsbereich, der Sie grafisch bei der Auswahl von Befehlen (und dazugehörigen Parametern) unterstützt.

Wofür PowerShell ISE eingesetzt wird

PowerShell ISE wird häufig für das Schreiben und Ausführen eigener Skripte verwendet. Zusätzlich bietet die PowerShell ISE eine ausgefeilte, grafisch orientierte Hilfe, die z. B. von Lernenden gern verwendet wird. Damit bedient PowerShell ISE besonders die Enden des Spektrums: durch Benutzerfreundlichkeit einer Windows-Anwendung PowerShell-Lernende und erfahrene Anwender, die anspruchsvolleren Code entwickeln.

64-Bit-Version und 32-Bit-Version

In 64-Bit-Versionen von Windows gibt es zusätzlich zur 64-Bit-Version der PowerShell eine 32-Bit-Version, die Sie im Verzeichnis *%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0* finden. Das betrifft beide Anwendungen – PowerShell-Konsole und PowerShell ISE.

Die 32-Bit-Version benötigen Sie nur dann, wenn die PowerShell auf Bibliotheken zugreift, für die es ausschließlich eine 32-Bit-Version gibt. Dies ist z. B. beim Zusammenspiel mit Microsoft Access der Fall, begegnet Ihnen im Normalfall aber nicht. Ansonsten verhalten sich die PowerShell-Versionen identisch.

Sie haben richtig gelesen: Die 32-Bit-Version der PowerShell befindet sich unterhalb des Verzeichnisses *SysWOW**64***, während die 64-Bit-Version in der Struktur unterhalb des Verzeichnisses *System**32*** zu finden ist.

1.4 Automatische Vervollständigung

Bei einer textbasierten Umgebung kann jeder Tippfehler verhindern, dass eingegebene Befehle so funktionieren, wie Sie es erwarten. Die PowerShell unterstützt Sie bei der Eingabe und hilft, Eingabefehler zu vermeiden.

Sowohl die PowerShell-Konsole als auch die PowerShell ISE haben Mechanismen zur Erleichterung Ihrer Eingaben:

- Die PowerShell-Konsole hilft Ihnen mit der sogenannten Tabulator-Vervollständigung. Wenn Sie einen Teil eines Befehls oder eines Pfads eingeben, können Sie mit der Tabulatortaste ⇥ eine Vervollständigung des Befehls oder Pfads auf Basis der bisher erfolgten Eingabe erreichen.
- Die PowerShell ISE stellt eine weitergehende Hilfe namens IntelliSense zur Verfügung. Grafisch orientierte Menüs helfen Ihnen automatisch bei der Vervollständigung einer Eingabe. Wird ein solches Hilfsmenü gerade nicht angezeigt, können Sie es durch die Tastenkombination Strg Leertaste einblenden.

Tabulator-Vervollständigung in der PowerShell-Konsole

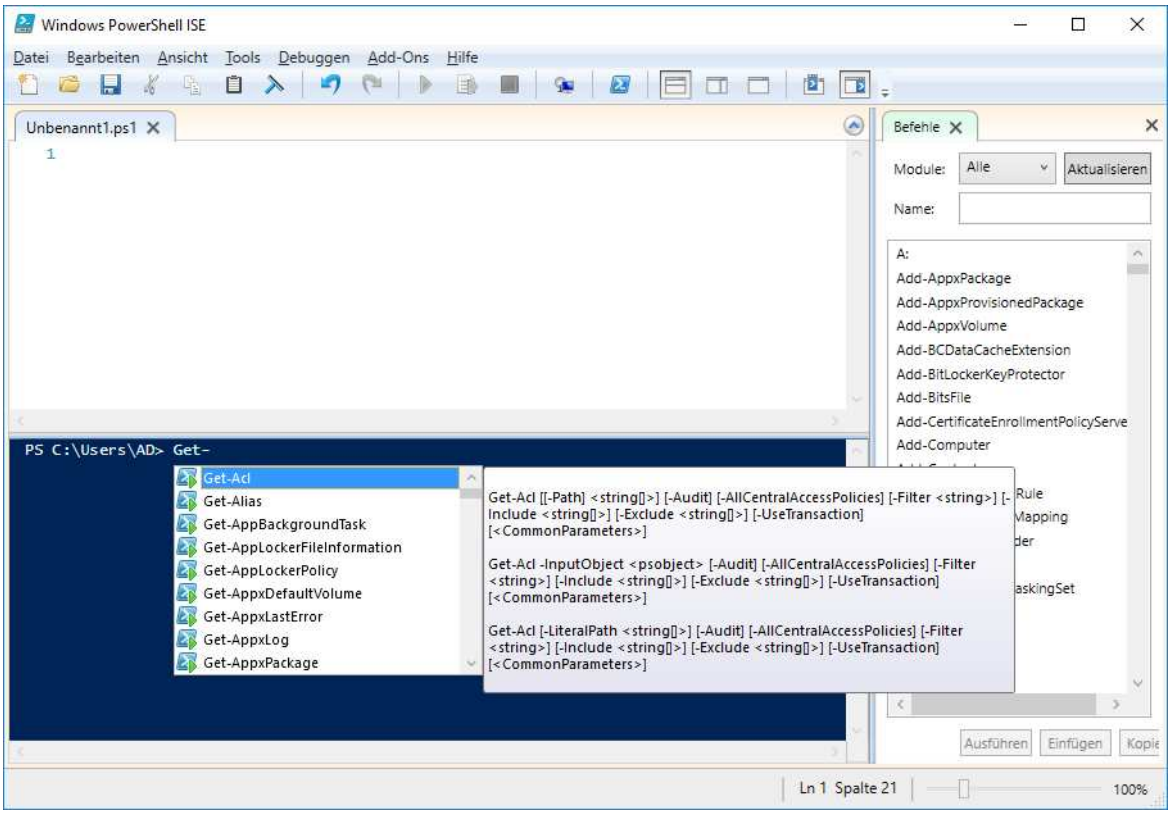
Wenn Sie die automatische Vervollständigung in der PowerShell-Konsole nutzen wollen, betätigen Sie mitten in einer unvollständigen Befehls- oder Pfadeingabe die Tabulatortaste ⇥.

- In einer Pfadangabe tippen Sie nur einen Teilpfad, z. B. *C:*\, und betätigen die Tabulatortaste ⇥. Mit jeder Betätigung von ⇥ schlägt die PowerShell ein anderes Verzeichnis oder eine andere Datei im Stammverzeichnis des Laufwerks C: vor. Die Anzeige erfolgt alphabetisch aufsteigend. Je mehr Buchstaben Sie eintippen, desto genauer wird die Auswahl der PowerShell, da Ihre Eingabe als Vorgabe verwendet wird.
- Zur Vervollständigung von Befehlen tippen Sie einen Teil des Befehls und drücken die ⇥-Taste. Die ersten Befehle lernen Sie bereits in diesem Kapitel kennen. Probieren Sie diese Funktion aus und sparen sich Tipparbeit.

IntelliSense in der PowerShell ISE

IntelliSense steht in der PowerShell ISE zur Verfügung, und zwar sowohl im dortigen Skripteditor als auch im Konsolenbereich.

Hier erscheint automatisch bei einer Teileingabe von Befehlen und Pfaden ein Hilfsmenü, wie Sie in der folgenden Abbildung sehen können. Die dort präsentierten Angaben können viel Arbeit ersparen, da die Hilfen weitgehend sind: Pfade, Befehle, Parameter und Werte werden passend zur vorgenommenen Eingabe vorgeschlagen.



IntelliSense: Grafisch orientierte automatische Vervollständigung in der PowerShell ISE

1.5 PowerShell starten

Verwendete Betriebssysteme

Im ersten Teil des Buches arbeiten Sie mit Windows 10 (oder alternativ mit Windows Server 2016 bzw. 2019). In den Kapiteln des zweiten Teils verwenden Sie Windows Server 2016 (oder 2019) für Arbeiten mit Active Directory sowie der Netzwerkverwaltung.

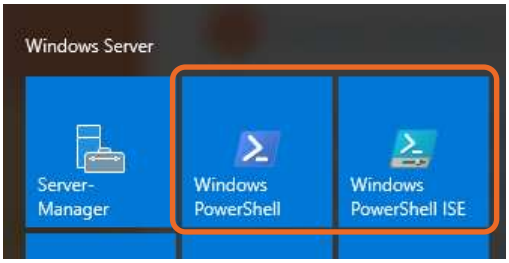
Falls Sie die genannten Betriebssysteme nicht zur Verfügung haben, keine Administratorrechte besitzen oder eine Testumgebung anstelle der Produktivumgebung verwenden möchten, richten Sie sich eine Testumgebung mit Windows 10 und/oder Windows Server 2016 (oder 2019) ein.

Plus **Ergänzender Lerninhalt**: *Testumgebung_einrichten.pdf*

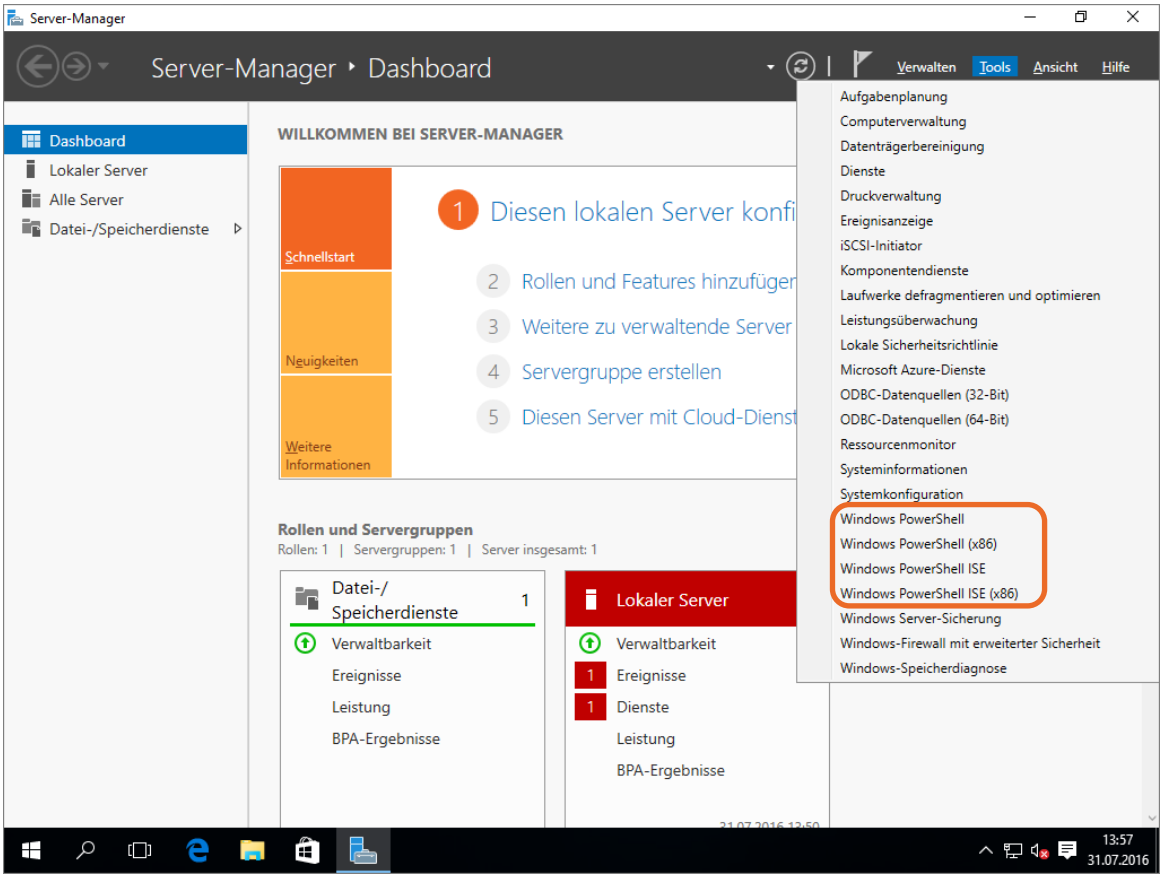
PowerShell unter Windows Server 2016/2019

Die PowerShell zählt zu den wichtigsten Werkzeugen für den Administrator. Aus diesem Grund müssen Sie nicht lange nach der PowerShell suchen. PowerShell-Kacheln für Konsole und ISE sind bereits im Startmenü des Servers vorhanden.

Darüber hinaus ist die PowerShell in das zentrale Administrationsinstrument, den Server Manager, unter dem Menüpunkt *Tools* integriert.



Windows Server 2016: PowerShell-Kacheln im Startmenü



Server Manager in Windows Server 2016: PowerShell-Einträge im Menü „Tools“

In Windows Server 2016 (oder Windows Server 2019) ist die PowerShell standardmäßig nicht mehr in der Taskleiste zu finden. Wenn Sie dies ändern möchten, führen Sie die folgenden Schritte aus:

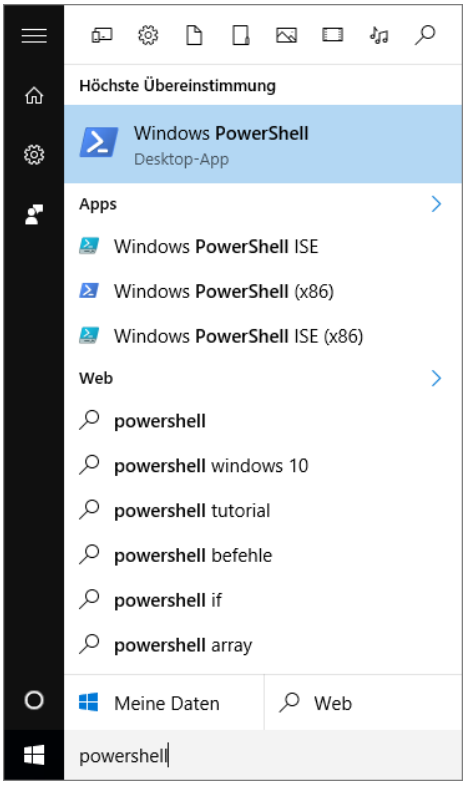
- Öffnen Sie das Startmenü (über ⊞ in der Taskleiste).
- Klicken Sie mit der rechten Maustaste auf die Kachel *Windows PowerShell*, um das Kontextmenü zu öffnen.

- Rufen Sie den Eintrag *Mehr* auf und wählen Sie *An Taskleiste anheften*.

Windows Server 2016 (oder 2019): Die PowerShell an die Taskleiste anheften

PowerShell unter Windows 10

In Windows 10 ist die PowerShell standardmäßig weder im Startmenü noch auf dem Desktop zu finden. Sie können die PowerShell jedoch auf verschiedenen Wegen aufrufen und sich die Umgebung so einrichten, dass keine weitere Suche nötig ist:



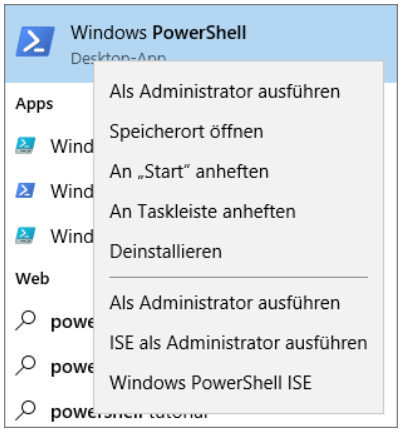
- Drücken Sie ⊞, um das Startmenü zu öffnen.
- Geben Sie die Zeichenfolge „powershell" in das Suchfeld ein, um im System nach Anwendungen mit dieser Zeichenfolge suchen zu lassen.

  Es reicht bereits aus, einen Teil der Zeichenkette einzugeben.

  Die Position des Mauszeigers spielt beim Start der Eingabe keine Rolle. Groß- und Kleinschreibung werden nicht unterschieden. Sie müssen auch keinSuchfenster öffnen, sondern können einfach lostippen.
- Klicken Sie mit der linken Maustaste auf das Suchergebnis *Windows PowerShell Desktop-App*.

  Die PowerShell wird in einem Konsolenfenster auf dem Desktop geöffnet.

Wenn Sie mit der rechten Maustaste auf den Eintrag im oben gezeigten Suchergebnis klicken, startet die Anwendung nicht, sondern es wird ein Kontextmenü geöffnet.



Sie können z. B. folgende Aktionen ausführen:

Aktion	Erläuterung	
Als Administrator ausführen	Nach einer Abfrage der Benutzerkontensteuerung, ob Sie die Aktion zulassen wollen, öffnet Windows ein neues PowerShell-Konsolenfenster mit erhöhten Rechten im Startverzeichnis *C:\Windows\System32*. Die Titelleiste der Anwendung weist Sie darauf hin, dass Sie die Anwendung als Administrator geöffnet haben. Der Titel des Fensters lautet *Administrator: Windows PowerShell* – und nicht, wie üblich, *Windows PowerShell*.	
Speicherort öffnen	Durch die Wahl dieser Option öffnet Windows ein Explorerfenster in einem Unterordner des Startmenüordners des angemeldeten Benutzers (*C:\Users\<NAME>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Windows PowerShell*; wobei <NAME> als Platzhalter für den angemeldeten Benutzer steht). Dort finden Sie die Verknüpfungen von PowerShell und PowerShell ISE (jeweils 32 Bit und 64 Bit). Die Verknüpfungen können Sie bei Bedarf über den Kontextmenüpunkt *Eigenschaften* anpassen.	
An „Start“ anheften	Die Anwendung wird als Kachel im Startmenü angezeigt. Somit können Sie PowerShell leicht finden und durch einen einfachen Linksklick auf die Kachel öffnen. Die Position der Kachel können Sie per Drag & Drop ändern. Durch einen Klick auf die Kachel mit der rechten Maustaste wird ein Kontextmenü angezeigt, das z. B. das Löschen der Kachel ermöglicht (Eintrag *Von „Start“ lösen*).	 *PowerShell-Kachel im Startmenü*
An Taskleiste anheften	Durch diese Aktion wird ein PowerShell-Icon an die Taskleiste des Windows-Desktops angeheftet. Die Anwendung können Sie mit einem einfachen Linksklick auf das Icon öffnen.	 *Icon in der Taskleiste*

Ist die PowerShell als Icon in der Taskleiste verfügbar, können Sie mit der rechten Maustaste die benötigte Aufgabe festlegen. Sie können die PowerShell oder die PowerShell ISE standardmäßig oder mit erhöhten Rechten starten oder PowerShell wieder aus der Taskleiste entfernen.

PowerShell in der Taskleiste: rechte Maustaste

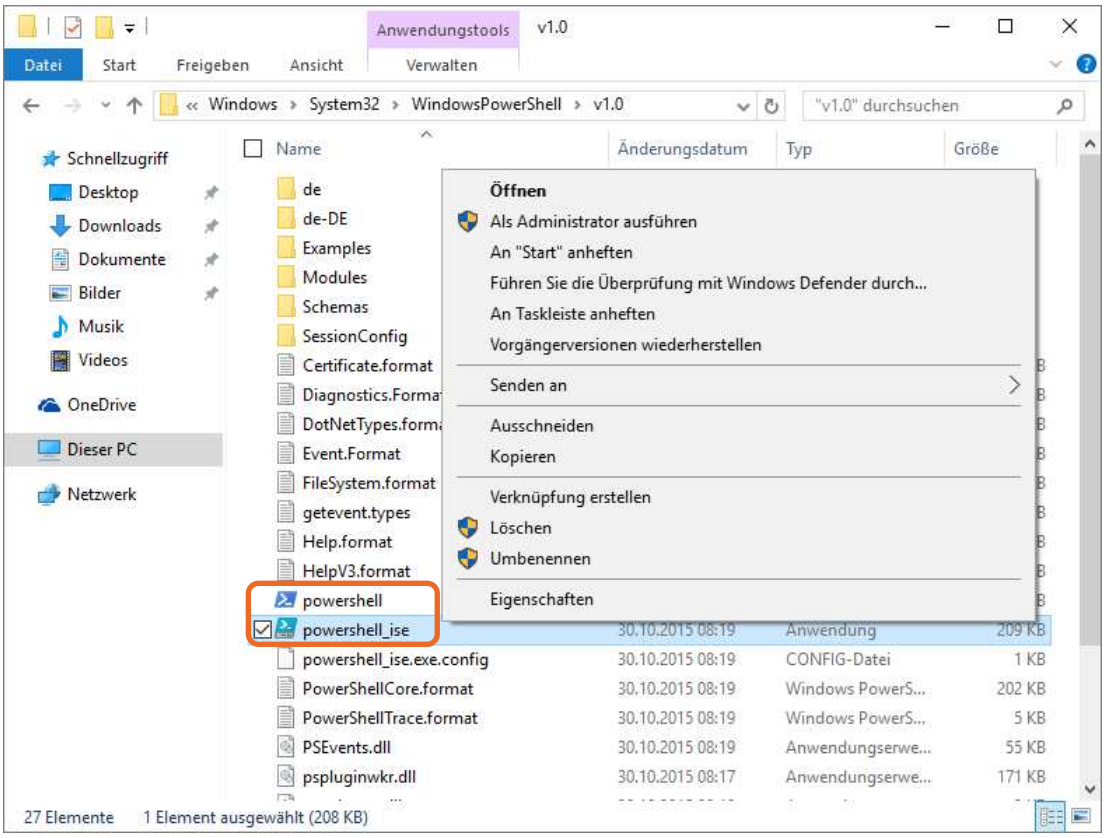
- Alternativ öffnen Sie den Windows-Explorer und navigieren Sie zum Installationsverzeichnis von PowerShell.

  Das Verzeichnis heißt *%WINDIR%\System32\WindowsPowerShell\ v1.0*, wobei die Systemvariable *%WINDIR%* das Windows-Installationsverzeichnis bezeichnet. Standardmäßig ist dies das Verzeichnis *C:\Windows*.

Lassen Sie sich von der Versionsnummer im Pfadnamen nicht verwirren. Jede PowerShell-Version ist im Verzeichnis *v1.0* zu finden. Der Pfad konnte aus Gründen der Abwärtskompatibilität ab PowerShell-Version 2.0 nicht mehr verändert werden.

- Im Installationsverzeichnis von PowerShell öffnen Sie die gewünschte PowerShell durch einen Doppelklick.

  Im Verzeichnis stehen Ihnen zwei Anwendungen zur Verfügung: *powershell.exe* für die PowerShell-Konsole und *powershell_ise.exe* für PowerShell ISE. Durch einen Klick mit der rechten Maustaste können Sie weitere Aktionen ausführen, z. B. die Anwendung als Kachel im Startmenü anzeigen, an die Taskleiste im Desktop anheften und das Ausführen der Anwendung mit erhöhten Rechten.



Installationsverzeichnis von PowerShell: „powershell.exe“ und „powershell_ise.exe“ (rechter Mausklick)