



Informationen

In der PowerShell können vier verschiedene Profile angelegt werden. Die Pfade zu den zugehörigen Profildateien werden in der PowerShell-Variable `$Profile` abgelegt. Die Profildatei bietet die Möglichkeit zur Individualisierung und Konfiguration von Profilen für verschiedenen Benutzern und Computern. Sie entspricht für den einzelnen Benutzer in der Funktionsweise grob der `bashrc` aus GNU/Linux Systemen. Die Profildatei wird bei jedem Start einer PowerShell-Session ausgeführt.



Aufgabe 1

- Lassen Sie sich durch Eingabe von `$Profile` anzeigen, unter welchem Pfad Sie Ihre Profildatei finden bzw. anlegen können. Existiert die Datei an diesem Pfad bereits? Prüfen Sie mit `Test-Path` nach.
- Verwenden Sie die Variable `$Profile` in Kombination mit der Formatierung durch `Format-List` mit dem Parameter `-Force`, um sich die Pfade zu allen vier Profildateien anzeigen zu lassen. Notieren Sie die Namen der vier Profile.


Aufgabe 2



- Legen Sie die in `$Profile` gespeicherte `ps1`-Datei mit `New-Item` an. Nutzen Sie ggf. den Parameter `-Force`, um fehlende Elemente im Pfad zur Datei mit anzulegen.
- Öffnen Sie die Datei mit dem Cmdlet `Invoke-Item`. Mit welcher App sind `ps1`-Dateien auf Ihrem System verknüpft?



- Verändern Sie die Profildatei derart, dass beim Start der PowerShell angezeigt wird, als welcher Benutzer auf welchem Computer¹ Sie gerade eine PowerShell-Session geöffnet haben.

 PowerShell 7 (x64)

```
PowerShell 7.3.1
Logged on as anr@HP-8B66VS859PI8
PS C:\Users\anr>
```



- Speichern Sie die Profildatei und starten Sie eine neue PowerShell. Ihnen sollte eine Nachricht vergleichbar mit der in der Abbildung angezeigt werden.

¹ Unter GNU/Linux gibt es keine Variable in `Env`: für den Computernamen. Verwenden Sie `uname` mit einem geeigneten Parameter.



Aufgabe 3 (Bonus)

Überlegen Sie, in welchen Anwendungsfällen Sie die anderen Profildateien verwenden würden.



Aufgabe 4

- Verwenden Sie das Cmdlet *Get-Alias* um sich Alternativnamen des Cmdlets *Get-ChildItem* anzeigen zu lassen. Für Hilfe zur Verwendung von *Get-Alias* benutzen Sie *Get-Help* o.ä. Welche Alias(e) gibt es für *Get-ChildItem* und woher kennen Sie diese ggf. bereits?
- Finden Sie die Aliase für das Cmdlet zum Entfernen (Löschen) von Objekten im Dateisystem.
- Finden Sie heraus, welches Cmdlet zum Alias *ixc* gehört.

Aufgabe 5

- Listen Sie alle Aliase auf, die mit dem Buchstaben ‚H‘ beginnen.
- Listen Sie alle Aliase für das Cmdlet *Remove-Item* auf.
- Definieren Sie die folgenden eigenen Aliase:
 - serv* für *Get-Service*
 - rand* für *Get-Random*



Aufgabe 6

- Legen Sie im Datenbereich zwei (leere) Dateien *loeschbar1.txt* und *loeschbar2.txt* an. Erstellen Sie den Alias *loesche* für das Cmdlet *Remove-Item*.
- Löschen Sie die angelegte Datei *loeschbar1.txt* mit dem Befehl *loesche*.
- Schließen Sie PowerShell, starten Sie eine neue Session und versuchen Sie, die zweite Datei *loeschbar2.txt* mit dem Befehl *loesche* zu löschen. Was folgern Sie daraus für die Persistenz von Aliasen?



Aufgabe 7 (Bonus)

Starten Sie eine PowerShell.

- a. Öffnen Sie die Profildatei Ihres aktuellen Benutzerprofils und fügen Sie ihr die folgenden dauerhaften Aliase hinzu:
 - *loesche* für *Remove-Item*
 - *ll* für *Get-ChildItem*
- b. Im Betriebssystem Ubuntu (eine GNU/Linux Distribution) werden die Befehle *poweroff* und *reboot* verwendet, um einen Computer herunterzufahren bzw. neu zu starten. Legen Sie in der Profildatei Ihres aktuellen Benutzerprofils geeignete Aliase mit diesen Namen an.
- c. Testen Sie alle angelegten Aliase in einer neuen PowerShell Session mit Testdateien.