



# PowerShell für Einsteiger

Grundlagen und Lernsituationen

© Julius Angres 2023

# Inhaltsübersicht

- ▶ Begrüßung/Vorstellung
- ▶ Impulsvortrag PowerShell
- ▶ PowerShell Grundlagen (Cmdlets, Hilfe zur Selbsthilfe)
- ▶ Pipeline, Prozesse und Dienste verwalten
- ▶ **Benutzer und Gruppen verwalten, Benutzerprofile**
- ▶ Dateisystem und NTFS-Rechte, Freigaben, Netzlaufwerke
- ▶ Netzwerkkonfiguration
- ▶ Serveraufgaben, Loganalyse, Webzugriff, Jobs
- ▶ Programmierung mit PowerShell (ps1-Skripte, Zugriff auf .NET-Objekte)
- ▶ Klassenarbeiten und Lernsituationen mit PowerShell erstellen und vorstellen


# Benutzer- und Gruppenobjekte

- ▶ Forderung: CRUD-Pattern für Benutzer- und Gruppenobjekte
  - Create           Anlegen von Benutzern
  - Read            Auslesen der Benutzereigenschaften
  - Update          Verändern von Benutzern
  - Delete          Löschen von Benutzern
- ▶ Hierfür müssen passende Cmdlets bzw. Parameter bereitgestellt werden!

# Benutzerverwaltung

- ▶ Lokale Benutzer auf Workstation
  - Cmdlets der Familie LocalUser
- ▶ Benutzer in Active Directory (AD) Umgebungen
  - Cmdlets der Familie ADUser

# Anlegen eines neuen Benutzers

- ▶ Cmdlet `New-LocalUser`
- ▶  Das Verb `Create` gibt es für Cmdlets (bis heute) nicht
- ▶ Beispiel:
  - Benutzer soll mindestens Namen und Passwort haben.
  - Rückblick in cmd-Welt für Benutzer *otto* mit Passwort *o*

```
C:\Users\anr>net user otto o /add
```



Einfache Handhabung, gut in Batch-Skripten einsetzbar



Passwort im Klartext angeben

# Anlegen eines neuen Benutzers

- ▶ **Cmdlet** `New-LocalUser`
- ▶ **Parameter** `-Name` und `-Password`
- ▶ Das Password ist ein Objekt vom Typ `System.SecurityString`
- ▶ Drei Eingabemöglichkeiten für Parameter `-Password`:
  - Wert weglassen  
→ Passwort maskiert am Prompt eingeben
  - Cmdlet `Get-Credential` als Wert verwenden  
→ Passwort in Pop-up-Fenster eingeben

# Anlegen eines neuen Benutzers

- ▶ Cmdlet `New-LocalUser`
- ▶ Parameter `-Name` und `-Password`
- ▶ Das Password ist ein Objekt vom Typ `System.SecurityString`
- ▶ Drei Eingabemöglichkeiten für Parameter `-Password`:
  - Wert direkt in SecureString konvertieren
    - `(ConvertTo-SecureString -AsPlainText -Force -String "o")`

# Anlegen eines neuen Benutzers

- ▶ Wer das Password ad hoc in SecureString verwandelt, hat keinen Vorteil ggü. dem `net user` Befehl (Passwort im Klartext)

## ☞ Achtung:

- ▶ Bei `net user /add` ist der Neue sofort Mitglied der Gruppe Benutzer
- ▶ Der `New-LocalUser` ist standardmäßig nicht Mitglied der Gruppe Benutzer
  - d.h. er kann sich nicht anmelden
  - muss von Hand der Gruppe Benutzer hinzugefügt werden
  - erhöht Kontrolle



# RUD für Benutzer

- ▶ Informationen auslesen mit `Get-LocalUser`
- ▶ Standardmäßig wird nur Name und Enabled-Status angezeigt
- ▶ Tipp:
  - Für mehr Informationen Formatieren mit `Format-List -Property *`
  - Funktioniert auch für alle anderen Cmdlets

# RUD für Benutzer

## Informationsanzeige erweitern

```
PS C:\Users\anr> Get-LocalUser anr
```

```
Name Enabled Description
----
anr   True
```

```
PS C:\Users\anr> Get-LocalUser anr | Format-List -Property *
```

```
AccountExpires      :
Description          :
Enabled              : True
FullName             :
PasswordChangeableDate : 27.12.2022 09:35:06
PasswordExpires      : 24.06.2023 10:35:06
UserMayChangePassword : True
PasswordRequired      : False
PasswordLastSet       : 26.12.2022 09:35:06
LastLogon            : 09.01.2023 14:34:31
Name                  : anr
SID                   : S-1-5-21-2609673462-2318655437-1353779694-1002
PrincipalSource       : Local
ObjectClass           : Benutzer
```

# RUD für Benutzer

- ▶ Benutzereigenschaften verändern
- ▶ Cmdlet `Set-LocalUser`
- ▶ Benutzer löschen
- ▶ Cmdlet `Remove-LocalUser`
- ▶ Für Verwendung und Hilfe die bekannten Hilfs-Cmdlets nutzen

# Gruppenverwaltung

- ▶ Es gibt verschiedene Typen von Gruppen in der Windows-Welt:
- ▶ Lokale Gruppen (hier behandelt)
- ▶ Domänenlokale Gruppen
- ▶ Universelle Gruppen
- ▶ Weitere Arten und Unterarten mit z.T. eigenen Cmdlet-Familien

# Gruppenverwaltung CRUD-Pattern

## ▶ Cmdlets der Familie `LocalGroup` für das

- Erstellen,
- Auslesen,
- Verändern und
- Löschen von Gruppen

# Gruppenmitglieder verwalten

- ▶ Mitglieder der Gruppe *Administratoren* anzeigen

```
PS C:\Users\anr> Get-LocalGroupMember -Group Administratoren
```

ObjectClass	Name	PrincipalSource
Benutzer	HP-8B66VS859PI8\Administrator	Local
Benutzer	HP-8B66VS859PI8\anr	Local

- ▶ Benutzer *anr* in Gruppe *Benutzer* aufnehmen (als Mitglied hinzufügen)

```
PS C:\Users\anr> Add-LocalGroupMember -Group Benutzer -Member anr
```

# Gruppenmitgliedschaft

- ▶ Aufgabe: Gruppenmitgliedschaften von Benutzern anzeigen
- ▶ **Besonderheit:**
- ▶ Es gibt keine (einfache) Möglichkeit, mit Cmdlets alle Gruppenmitgliedschaften eines Benutzers anzuzeigen
- ▶ Informationsanzeige erweitern bringt keinen Erfolg

# Gruppenmitgliedschaft

- ▶ Eingabeaufforderung: `net user anr`
- ▶ **Achtung:** `net user` gibt String zurück und kein Objekt!

```
PS C:\Users\anr> net user anr
Benutzername                anr
```

```
Lokale Gruppenmitgliedschaften  *Administratoren
Globale Gruppenmitgliedschaften *Kein
```

- ▶ Best Practice:
  - `net user` Befehl weiterhin verwenden
  - Alternativ: über alle Gruppen iterieren, Mitglieder anzeigen und filtern



# Benutzerprofile

- ▶ Profilinformationen sind in PS-Variable `$Profile` gespeichert
- ▶ Es gibt insgesamt vier Profile:
  - `CurrentUserCurrentHost`
  - `CurrentUserAllHosts`
  - `AllUsersCurrentHost`
  - `AllUsersAllHosts`
- ▶ Standardprofil ist `CurrentUserCurrentHost`
- ▶ Einstellungen werden in Profildatei gespeichert

# PowerShell Aliase

- ▶ PowerShell besitzt bereits zahlreiche Aliase
- ▶ Wesentliche Cmdlets:
  - `Get-Alias`
  - `Set-Alias`
- ▶ Zweck von Aliasen:
  - Abkürzung häufiger Befehle
  - Komfort für Umsteiger von cmd und GNU/Linux

# PowerShell Aliase

- ▶ Beispiel:
  - Löschen des PowerShell Bildschirms (bzw. des Puffers)

Interface	Cmdlet/Befehl
PowerShell	Clear-Host
Windows cmd	cls
GNU/Linux bash	clear

- ▶ PowerShell implementiert `cls` und `clear` als Alias von Clear-Host
- ▶ Dadurch erleichterter Umstieg von cmd und bash

# Übung PS41, PS42, PS44

## Benutzer, Gruppen und Benutzerprofile

- ▶ Cmdlets zum Anlegen und Verwalten von Benutzern
- ▶ Cmdlets zum Anlegen und Verwalten von Gruppen
- ▶ Optional zusätzlich Übung PS43
- ▶ PowerShell Benutzerprofile anlegen
- ▶ PowerShell Aliase verwenden