



# The Beauty Of Powershell

Matthias Gessenay, Raphael Föh

Partner:  Microsoft ***Computerworld***

**DIGICOMP**

# Agenda

- Begrüssung
- Vorstellung Referent
- Content
- F&A
- Weiterführende Kurse

## Vorstellung Referent

Name: Matthias Gessenay  
Firma: Corporate Software AG  
Funktion: Consultant & Trainer

[matthias.gessenay@corporatesoftware.ch](mailto:matthias.gessenay@corporatesoftware.ch)  
[www.corporatesoftware.ch](http://www.corporatesoftware.ch)

Follow: @CorpSoftCH on Twitter  
Session Hashtag: #TBofPS



## Vorstellung Referent

Name: Raphael Fäh  
Firma: Corporate Software AG  
Funktion: Consultant & Trainer

[raphael.faeh@corporatesoftware.ch](mailto:raphael.faeh@corporatesoftware.ch)  
[www.corporatesoftware.ch](http://www.corporatesoftware.ch)

Follow: @CorpSoftCH on Twitter  
Session Hashtag: #TBofPS



## Nested Pipes

```
Get-Childitem -Path "C:\Scripts" | Foreach-Object {  
    (Get-Content -Path $_.FullName) -replace "TestValue","TestValueRenamed!" | Set-Content  
    -Path $_.Fullname  
}
```

# Demo Nested Pipes

## Conclusion Nested Pipes

- Wenn man `$_` in einem Scriptblock verwendet, der in einem Cmdlet in der Pipeline ausgeführt wird, wird das am weitesten “genestete” Objekt verwendet
- Wenn man `$_` als Parameter für ein Cmdlet in der Pipeline verwendet, wird das Objekt der äusseren Pipeline verwendet

## Adapted Type System (ATS) & Extended Type System (ETS)

- Begriffe:
  - Objekt (Enthält Daten und Möglichkeiten, mit dem Objekt und den Daten zu interagieren)
  - Member
    - Properties
    - Methoden
    - NoteProperties
    - Scriptmethoden
- Objekt = Sammlung von Members



# PSObject

- Datenquellen:
  - WMI
  - COM
  - .NET
  - Etc.
- Powershell verwendet nicht direkt die Objekte der Datenquellen, sondern wrapped sie in eigene Objekte (PSObject)

## Adapted Type System (ATS)

- Wenn ein WMI Objekt durch Powershell aufgerufen wird, adaptiert Powershell die Properties und Methoden des Objekts auf ein neuerstelltes Powershell Objekt. Durch diese Adaption werden alle Member des darunterliegenden Objekts direkte primäre Member des Powershell Objekts.
- Ausgehend vom Basis WMI Objekt kann ein Mapping vorgenommen werden. Für einen Vergleich kann .NET herangezogen werden.
- `Get-WMIObject Win32_Process | Get-Member -View Base #Member`, wie Sie im .NET Objekt auftauchen (37 Member)
- `Get-WMIObject Win32_Process | Get-Member -View Adapted #Member`, wie Sie im Powershell Objekt auftauchen (61 Member)

## Extended Type System (ETS)

- Das ATS erlaubt es Powershell, eine direkte Repräsentation eines Basisobjekt in einem Powershellobjekt zu erstellen
- Das ETS erlaubt es Powershell, jederzeit und zu jedem Objekt dynamisch weitere Member hinzuzufügen
- `Get-WmiObject Win32_Process | Get-Member -View Extended #Dynamic ETS Members (8 Member)`

## Demo ATS & ETS

# Dynamic Parameters

- Mittels Dynamic Parameter kann eine Bedingung definiert werden, unter der ein Parameter einer Funktion angezeigt werden soll. Es kann also verhindert werden dass ein Benutzer einen Parameter angeben kann, obwohl das nicht erforderlich oder sogar nicht gut ist.
- Vorgehen:
  - Die Parameterattribute müssen definiert werden
  - Die Attribute müssen in eine Collection gespeichert werden
  - Der RuntimeDefined Parameter muss erstellt werden
  - Für den RuntimeDefined Parameter muss ein Dictionary Objekt erstellt werden und der Parameter dem Dictionary hinzugefügt werden

# Demo

# Proxy Functions & Implizites Remoting

- Proxy Functions existieren seit Powershell 2.0
- Sie erlauben, für existierende Cmdlets und Funktionen einen weiteren Wrapper zu schreiben
- Anwendungsmöglichkeiten
  - Export-PSSession
  - Erweitern existierender Core-Cmdlets

## Erweiterung existierender Core Cmdlets

- Vorgehen:
- Cmdletdefinition finden
  - `$metadata = New-Object System.Management.Automation.CommandMetaData (Get-Command Export-CSV)`
  - `[System.Management.Automation.ProxyCommand]::Create($metadata) | Clip`
- Mit einer Funktion umgeben
- Die eigenen Parameter hinzufügen
- Den Cmdletinhalt erweitern um die Verarbeitung der eigenen Parameter zu ermöglichen



# Demo Erweiterung Core Cmdlet

## Export-PSSession & Implizites Remoting

- Aus einer Powershellsession können mit Export-PSSession einzelne Cmdlets oder ganze Module exportiert und lokal gespeichert werden
- Die lokal gespeicherten Cmdlets werden als Modul abgelegt und können als solches auch wieder importiert werden

---

# Export-PSSession & Implizites Remoting Demo

# F&A

---

## Weiterführende Kurse

## Links

- Source Files for this Session:
  - <https://github.com/CorpSoftCH/EvolutionDay2015>

