

## PowerShell Expert Class

### **Dr. Tobias Weltner**

Haben Sie ein gutes Grundwissen von PowerShell und möchten ihr Wissen noch vertiefen.

Dr. Tobias Weltner ist einer der anerkanntesten deutschsprachigen PowerShell Experten und wurde zum x-ten Mal in Folge mit dem Microsoft MVP Award ausgezeichnet. Dr. Weltner ist Autor des deutschsprachigen PowerShell Standardwerks "PowerShell 5 - Windows Automation für Einsteiger und Profis" und hat auch über 130 weitere Fachbücher zu PowerShell publiziert.

Der Kurs ist in Deutsch

## PowerShell Expert Class

### Arbeitsmittel:

Jeder Teilnehmer kann die Übungen auf den Schulungsrechnern durchführen und braucht kein Arbeitsmittel mitzunehmen.

### Voraussetzung:

Gute Kenntnisse von PowerShell

### Dauer:

3 Tage

### Mitnehmen:

Bringen Sie Ihre Fragen und Scripts mit

### Kursthemen:

Der Kurs basiert auf Windows PowerShell 5.0 und umfasst die unten aufgeführten Themenkreise. Mit Ihren Fragen und mitgebrachten Scripts können Sie den Kursinhalt aktiv beeinflussen.

#### **Einführung**

- PowerShell Architektur und Roadmap
- PowerShell Core /7 im Vergleich zu Windows PowerShell

#### **PowerShell Pipeline**

- Wie funktioniert die Pipeline genau und was passiert hinter den Kulissen
- Pipeline als Flaschenhals
- Performance und Optimierung beim Verwenden der Pipeline
- Sequenzieller- / Streaming Mode

#### **Functions**

- Einfache und Advanced Funktion
- CmdletBinding Attribut
- Function Parameters für Fortgeschrittene
- Parameter Attributte (welche sind wichtig, wo finden wir die Definition, System.Management.Automation.ParameterAttribute)
- Splatting
- Dynamische Parameter
- Referenz Parameter, Parameter bei Reference übergeben
- PowerShell Funktionen welche die Pipeline unterstützen
- ValueFromPipeline, ValueFromPipelineByPropertyName
- Gültigkeitsbereiche (lexical und dynamic scoping) / Scope modifiers

- Rückgabewerte / das Return Statement
- Objekte als Rückgabewerte / Rückgabotyp

### Exception Handling, Debugging und Logging

- Debuggen eines bereits laufenden Scripts / Debuggen eines Scheduled Task
- Remote Debugging, Runspace Debugging
- Best Practices beim Exception Handling
- Try-Catch-Finally / Throw / eigenen Exception Klasse
- Error Record und Exception
- Trace-Command (die wichtigsten trace Möglichkeiten z.B. Trace-Command -Name ParameterBinding)
- Logging in PowerShell (Logtools, Eventlog)
- Umgang mit den Streams von PowerShell

### Datentypen und Arbeiten mit Objekten

- Inspect Objects (Welcher Typ haben wir, was für Methoden und Properties existieren)
- Extended Type System (ETS)
- PowerShell 5 Classes (Attribute, Methoden, Konstruktoren, statische Mitglieder)
- Gültigkeitsbereiche von PowerShell 5 Classes
- Rückgabewerte von PowerShell 5 Methoden
- Aufzählungstypen (Enums)
- Type casting (Implizite / Explizite)
- Type Accelerators (wo finden wir diese, welche gibt es)
- PowerShell und die .Net Class Library (Wichtige Typen für SystemAdmins z.B. System.IO.Path)
- Typen aus DLLs nachladen / .Net-Code kompilieren / Systemassemblies nachladen
- Collections
- PObject
- JSON

### Entwicklungswerkzeuge

- Visual Studio Code
- ISE / ISESteroids

### Advanced Scripting

- Gültigkeitsbereiche / Dotsourcing
- #requires
- Strict Mode
- Help und dokumentieren von Scripts (manuell, automatisch, tags)
- Signieren von Scripts / wie kann sichergestellt werden, dass das Script aus einer vertrauenswürdigen Quelle stammt
- PowerShell Scripte kompilieren (Wie geht das, wo sind die Grenzen, wann sinnvoll, warum überhaupt)

### PowerShell Lifecycle

- Vom Script zum Modul
- Gültigkeitsbereiche / Imports / Exports / Modul Gültigkeitsbereich
- Wo sind Module gespeichert, wie werden diese installiert

- Wichtige Regeln beim Erstellen von Modulen (Namensgebung, Namenskonflikte)
- Repositories
- PowerShell Get
- Zusammenarbeit und Versionierung
- Wie kann man einem fremden Modul vertrauen

### **Remoting und Background Jobs**

- Universelles PowerShell Remoting mit Enter-PSSession und Invoke-Command
- Gültigkeitsbereiche / Variablen / Argumente / Rückgabewerte / Serialisierung
- Session Verhalten und Session lifetime
- WSMAN
- Double Hop Problematik
- PowerShell Runspaces

### **Benutzeroberflächen (GUI)**

- WPF /XAML
- Effizientes Erstellen von GUIs
- Wo sind die Grenzen

### **Win32 / Low Level Zugang**

- WMI
- Registry
- COM Komponenten
- plnvoke / PowerShell und die API von Windows
- Events (z.B. FileSystem Watcher / WMI Events)

### **Daten aus dem Internet**

- Webservices mit PowerShell ansprechen
- Invoke-WebRequest
- Feeds / Umgehen mit einem Proxy

### **Windows PowerShell Reference**

- Einführung in die wichtigen Class Libraries z.B. System.Management.Automation, Microsoft.PowerShell / Wie findet man sich in der Reference in MSDN zurecht
- Das PS SDK - Was müssen wir wissen darüber
- Abstract Syntax Tree / Der PowerShell Parser

### **Security**

- Credentials / Wie können diese sicher abgelegt werden
- Transcript
- Sichere Evaluation
- Restricted Endpoints