

# Windows PowerShell

## Windows Management Instrumentation (WMI)

### 1 Theorie

#### 1.1 Grundlagen

WMI ist eine Schnittstelle, über welche lokal oder über das Netzwerk, auf nahezu alle Einstellungen eines Windows-Computers zugegriffen werden kann.

WMI ist ein objektorientiertes Konzept, bei dem alle Informationen in Form von strukturierten Objekten bereitgestellt werden.

WMI-Klassen beginnen meistens mit dem Präfix Win32\_ (konkrete Windows Ressourcen) oder CIM\_ (allgemeine, betriebssystemunabhängige Ressourcen).

Eine Liste aller WMI-Objekte erhält man mit:

```
Get-WmiObject -List
```

WMI erlaubt zusätzlich Suchanfragen in ANSI-SQL. Allgemein:

```
SELECT attributlist FROM class WHERE condition
```

Die nachfolgenden Abfragen sind identisch:

```
Get-WmiObject Win32_Service | Where-Object {$_.State -eq  
"Running" -and $_.StartMode -eq "auto"}
```

```
Get-WmiObject -query "select * from Win32_Service where  
state='Running' and StartMode = 'Auto'"
```

## 1.2 WMI-Beispiele

Probieren Sie die nachfolgenden Beispiele aus!

<code>Get-WmiObject -List</code>	Liste aller WMI-Klassen
<code>Get-WmiObject Win32_BIOS   Select-Object *</code>	Alle Informationen zum BIOS ausgeben
<code>Get-WmiObject Win32_Processor   Get-Member</code>	Alle Eigenschaften und Methoden der Klasse Win32_Processor anzeigen
<code>Get-WmiObject Win32_ComputerSystem   Select-Object *</code>	Die Werte aller Eigenschaften der Klasse Win32_ComputerSystem anzeigen
<code>Get-WmiObject Win32_Networkadapter</code>	Alle Netzwerkadapter anzeigen
<code>Get-WmiObject Win32_Networkadapter   Where-Object {\$_.Macaddress}</code>	Alle Netzwerkadapter auflisten, welche über eine MAC-Adresse verfügen
<code>Get-WmiObject -List   Where-Object {\$_.Properties.Name -like "*memory*"}</code>	Auflisten aller Klassen, die über eine Eigenschaft verfügen, die den Begriff "memory" beinhalten
<code>Get-WmiObject -query "select * from Win32_Product"</code>	Alle installierten Programme ausgeben
<code>Get-WmiObject -query "select * from Win32_Product where Vendor like 'Microsoft%'"</code>	Alle installierten Programme von Microsoft ausgeben

## 2 Übungen

### Aufgabe 1

Schreiben Sie eine Funktion, welche die freie Diskkapazität pro Disk in GB in einer Hashtable zurückgibt (**Win32\_LogicalDisk**).

### Aufgabe 2

Schreiben Sie eine Funktion, welche die totale RAM-Kapazität in GB zurückgibt (**Win32\_PhysicalMemory**).

### Aufgabe 3

Schreiben Sie eine Funktion, welche die freie RAM-Kapazität zurückgibt (**Win32\_OperatingSystem**).

### Aufgabe 4

Schreiben Sie eine Funktion, welche alle lokalen Benutzer (Benutzernamen) in einem Array zurückgibt (**Win32\_UserAccount**).

### Aufgabe 5

Schreiben Sie eine Funktion, welche Name und Pfad aller Freigaben zurückgibt.