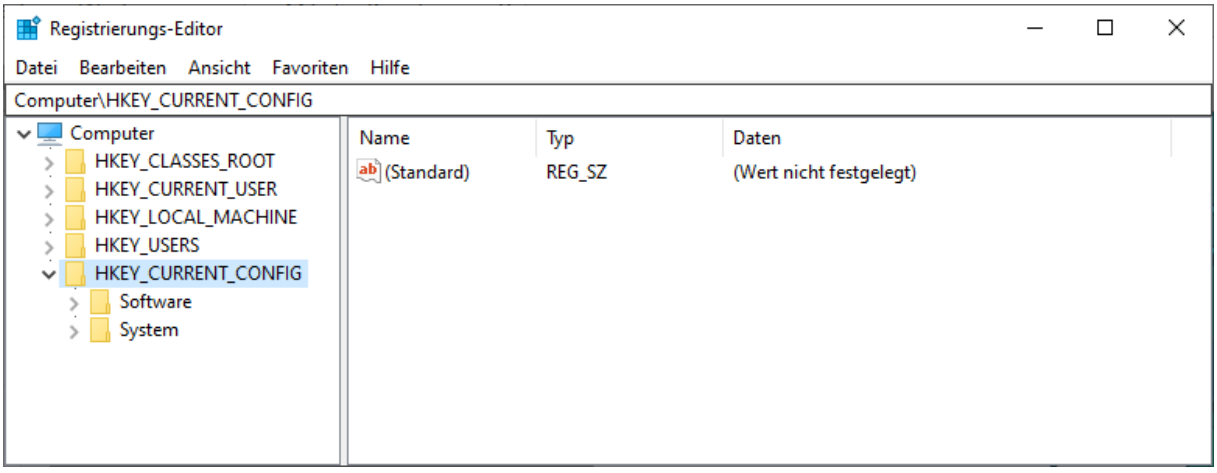
Umgang mit der Registry

Mit Hilfe der Powershell ist das Einsehen und Bearbeiten der Registry recht einfach. In dieser Doku zeige ich einige Beispiele, wie wir zu definierten Hives gelangen, und die gesetzten Werte auslesen bzw. verändern.

Der Aufbau der Registry ähnelt dem des Dateisystems. Statt Ordner und Unterordner gibt es in der Registry Schlüssel und Unterschlüssel, statt Dateien sind es hier Werte.



HKEY_CLASSES_ROOT enthält zumeist Dateinamenserweiterungen und die dazugehörigen Programmverknüpfungen. Die meisten Schlüssel verweisen auf eine Class-ID (CLSID), zu dem ein gleichnamiger Schlüssel unter "HKEY_CLASSES_ROOT \Clsid\" existiert. Dabei handelt es sich um Funktionen meist in DLL-Dateien, die Windows oder eine Anwendung für die interne Nutzung registriert haben.

Unter HKEY_CURRENT_USER sind alle Einstellungen des aktuell angemeldeten Benutzers gespeichert. Unterhalb von „HKEY_CURRENT_USER\SOFTWARE“ legen Anwendungen bei der Installation und Konfiguration ihre Daten ab. Die Windows-Konfiguration ist unter "HKEY_CURRENT_USER\SOFTWARE\Microsoft" zu finden.

Umgang mit der Registry

HKEY_LOCAL_MACHINE speichert systemspezifische Einstellungen für Windows und Anwendungen. Diese können in der Regel durch einen Administrator geändert werden und gelten für alle Benutzer. Im Zweig unter "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet" finden wir die Einstellungen für Dienste und Treiber.

HKEY_USERS enthält Schlüssel mit Benutzer-IDs.

- S-1-5-18 ist der Benutzer "System"
- S-1-5-19 gehört zu "NT Authority" und
- S-1-5-20 zu "Network Service".

Die beiden anderen IDs sind die des aktuellen Benutzers und entsprechen dem Inhalt von "HKEY_CURRENT_USER" beziehungsweise "HKEY_CLASSES_ROOT ".

HKEY_CURRENT_CONFIG ist eine Verlinkung zu den Schlüsseln unterhalb von "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles\Current\". Dadurch erreicht man für jegliche Anwendungen, einen kürzeren Pfad, um auf die Werte zu greifen zu können.

Fangen wir damit an, dass wir uns zuerst die beiden wichtigsten Pfade ansehen. Navigiert wird in der Registry ähnlich wie im Windows Explorer. Das sind die Pfade HKEY_LOCAL_MACHINE (HKLM) und HKEY_CURRENT_USER (HKCU).

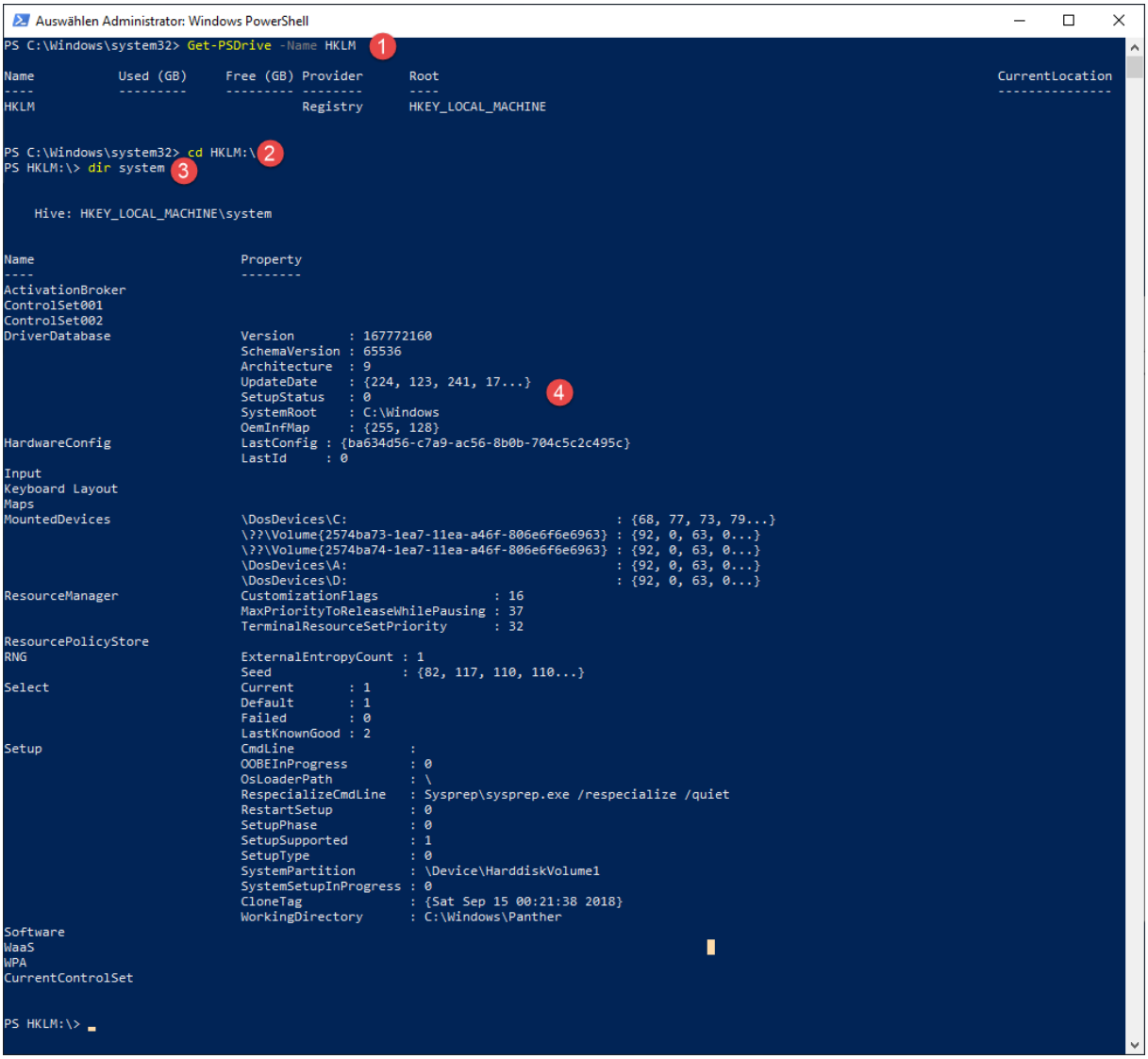
Wir öffnen die Powershell mit administrativen Rechten und setzen folgende Befehle ab:

```
Get-PSDrive -Name HKLM
Get-PSDrive -Name HKCU
```

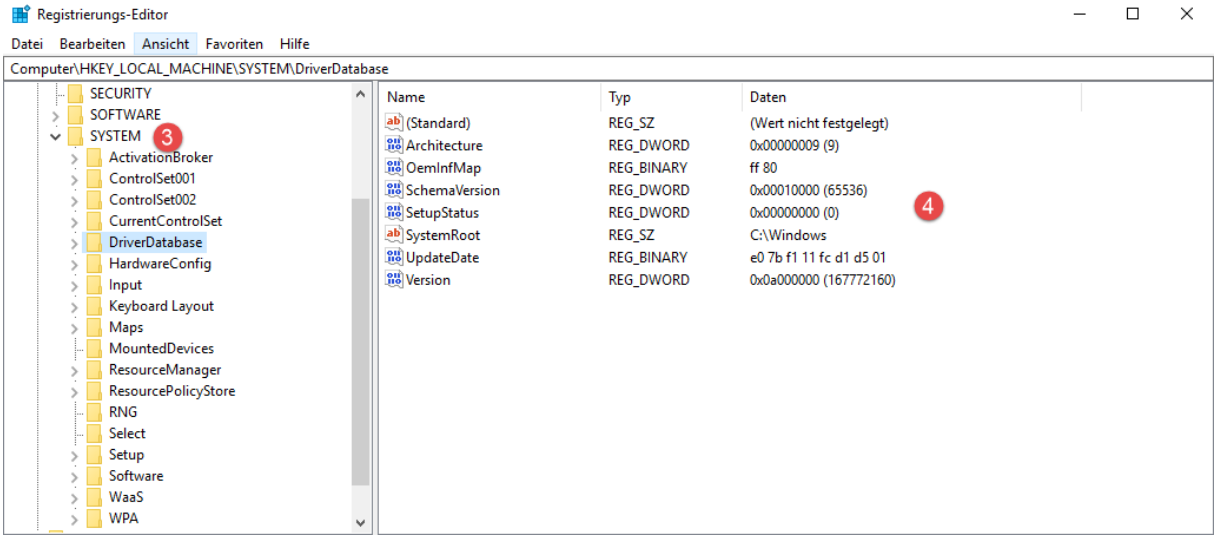
Umgang mit der Registry

Mit diesen 3 Befehlen navigieren wir schon etwas tiefer in die Untiefen der Verzweigungen.

Get-PSDrive -Name HKLM
cd HKLM:\
dir system

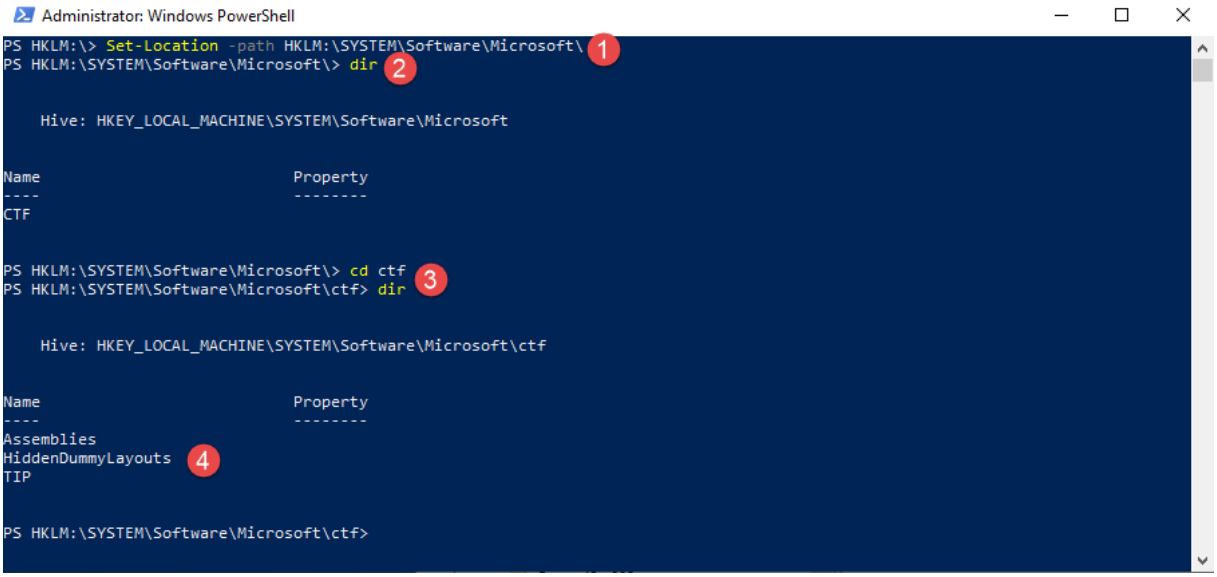


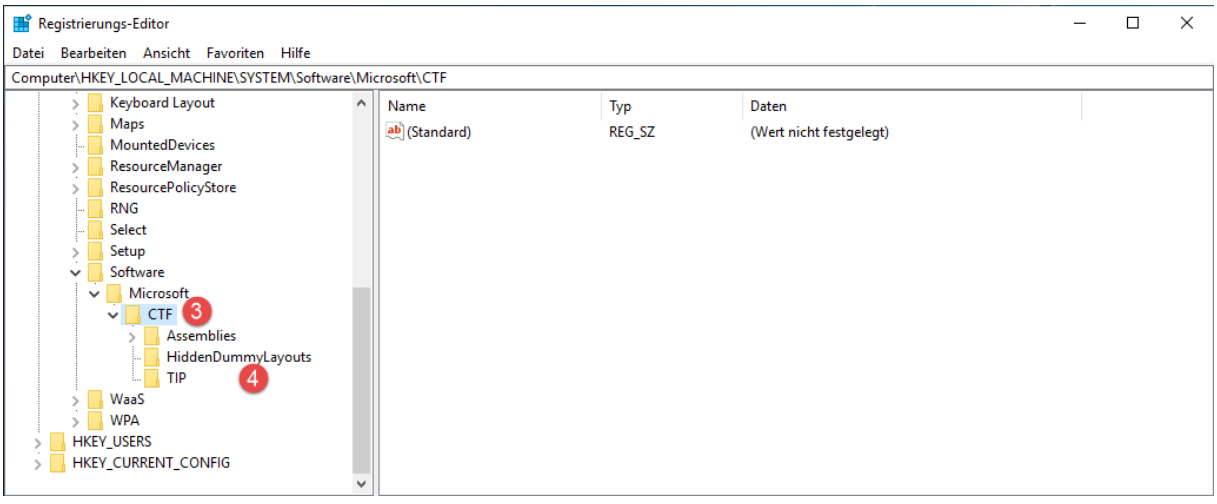
Umgang mit der Registry



Es ist auch möglich direkt zu einem Pfad zu springen und zwar über…

```
Set-Location -path HKLM:\SYSTEM\Software\Microsoft\
cd ctf
dir
```





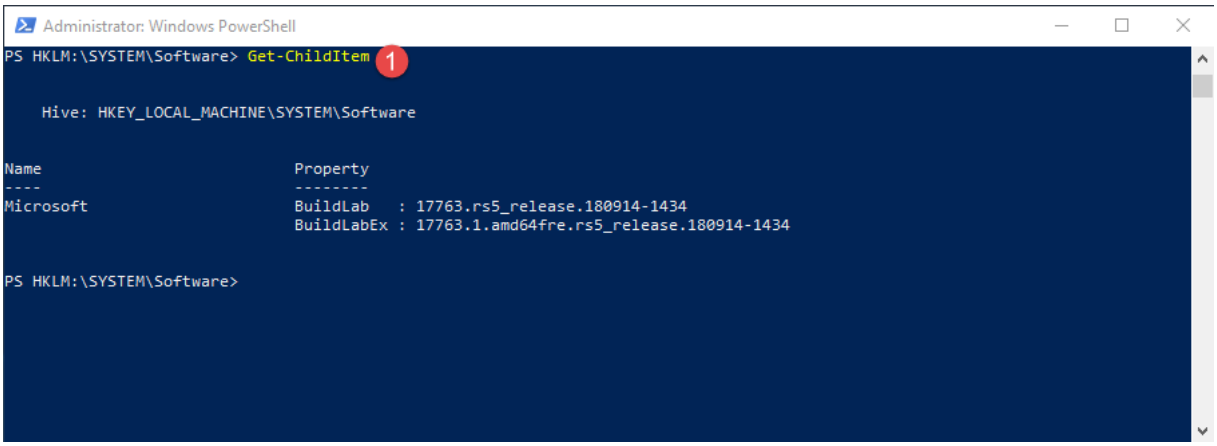
Erstellt von Jörn Walter **28.01.2020**

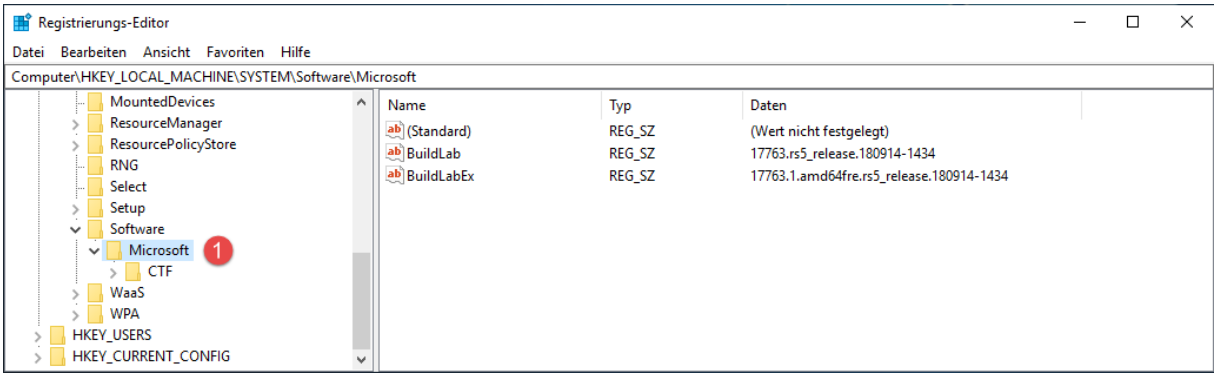
Umgang mit der Registry

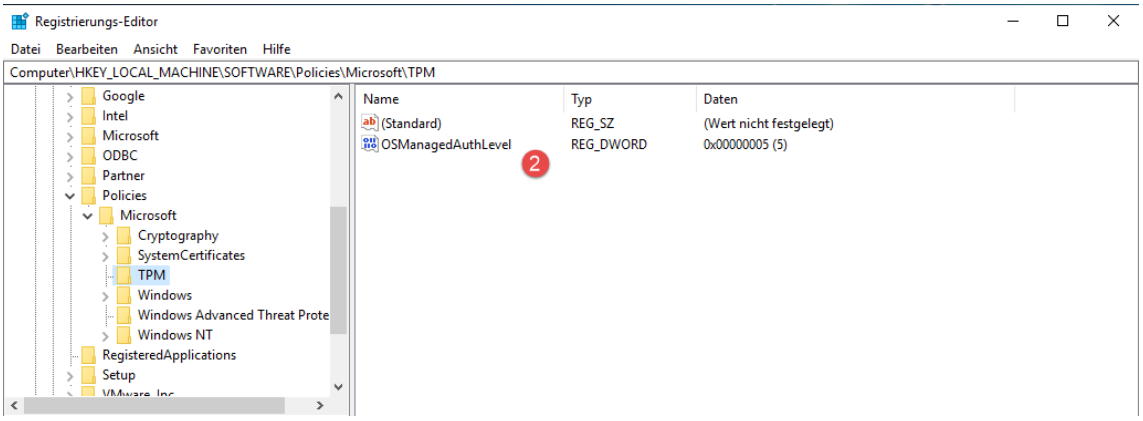
Mit cd navigieren wir vorwärts und mit cd.. rückwärts mit cd / springen wir zurück zum Stammverzeichnic

Mit diesem Befehl lassen wir uns sofern vorhanden alle Properties unter dem Schlüssel Software anzeigen.

Get-ChildItem

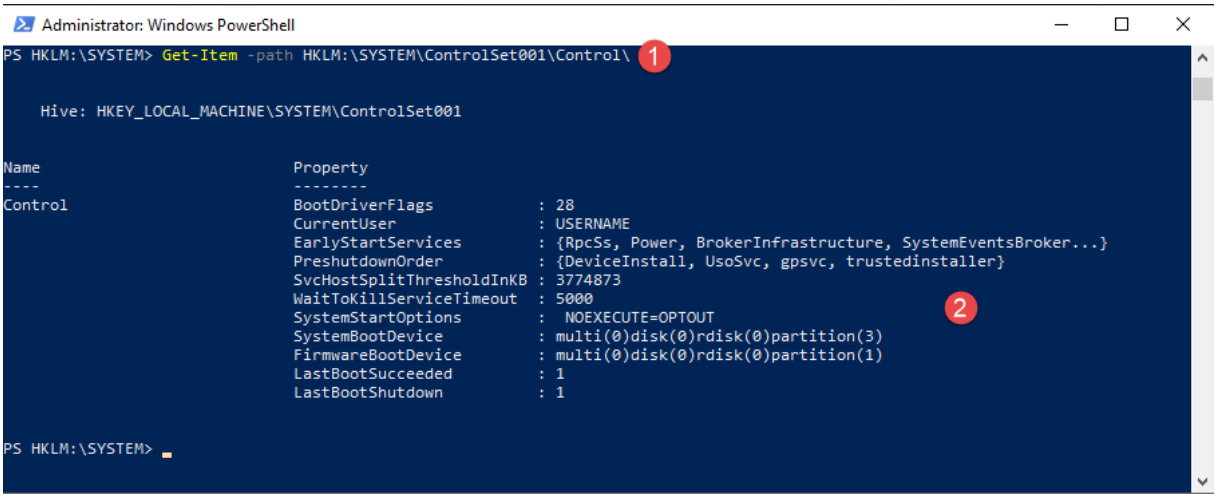


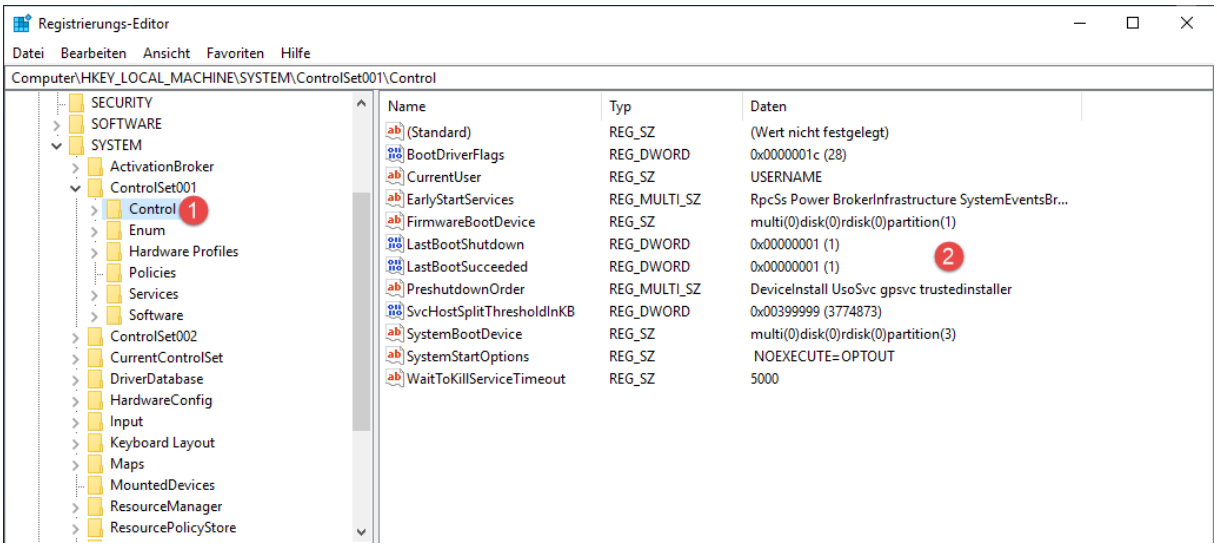




Mit diesem Befehl lassen wir uns die Properties gezielt ab unter dem Schlüssel Control anzeigen.

Get-Item -path HKLM:\SYSTEM\ControlSet001\Control\

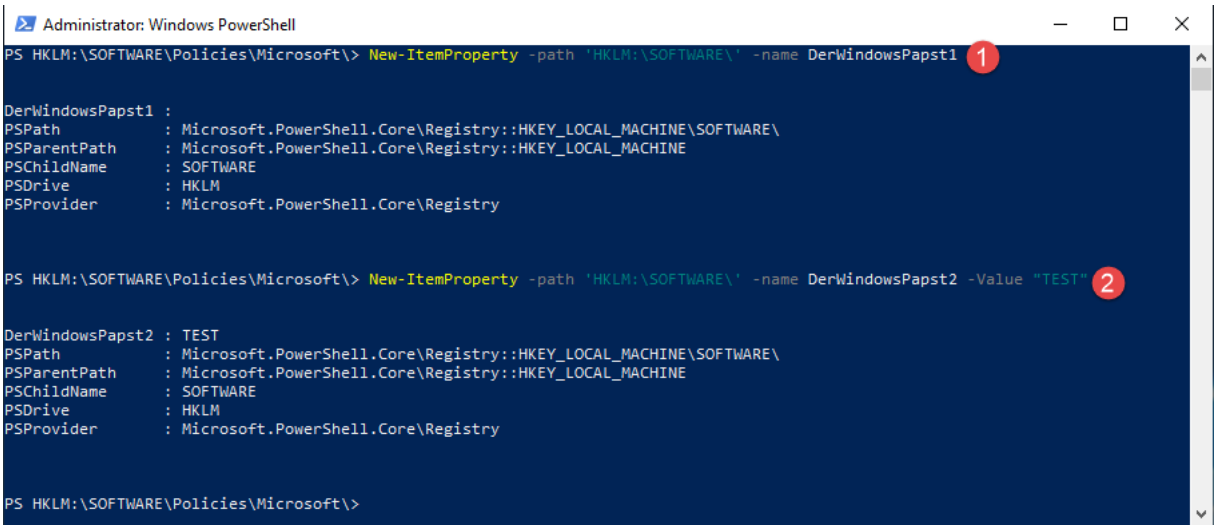


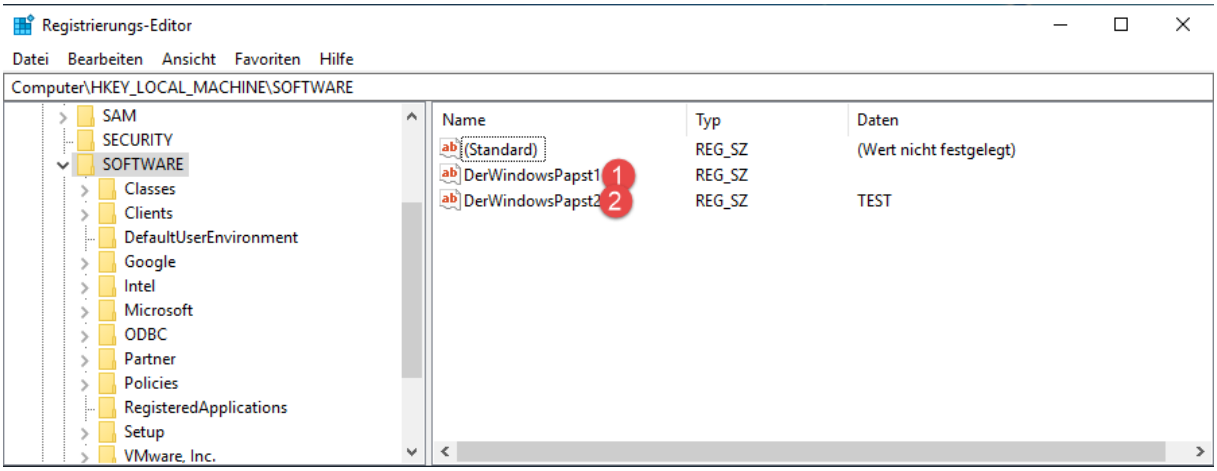


Umgang mit der Registry

Schauen wir uns gleich an, wie wir die Parameter von Registry-Einträgen ändern können. Aber zuerst legen wir welche an. Und zwar vom TYP REG_SZ unterhalb von SOFTWARE.

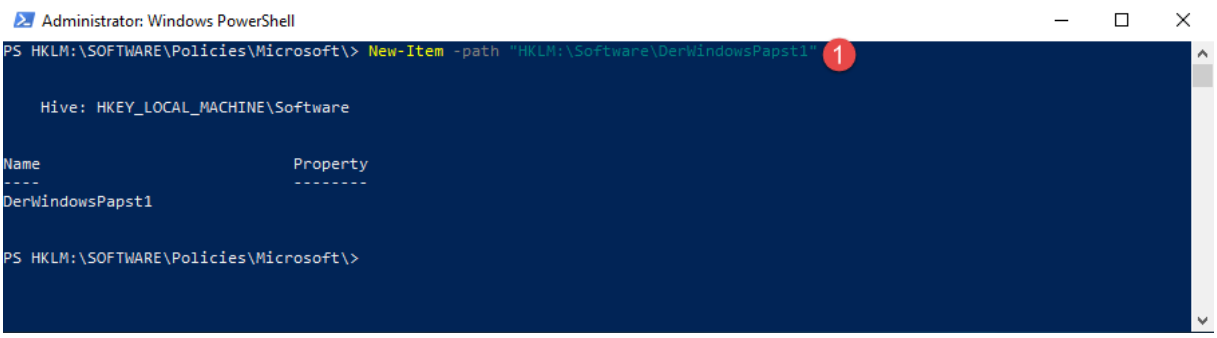
New-ItemProperty -path 'HKLM:\SOFTWARE\' -name DerWindowsPapst1
New-ItemProperty -path 'HKLM:\SOFTWARE\' -name DerWindowsPapst2 -Value "TEST"



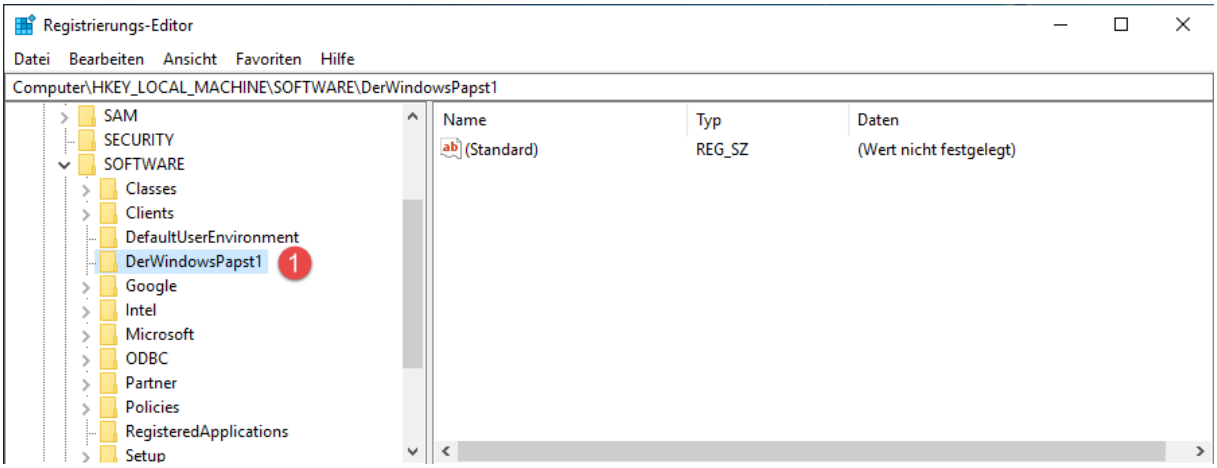


Als nächstes legen wir einen neuen Schlüssel an und darunter einen neuen Eintrag vom TYP DWORD.

New-Item -path "HKLM:\Software\DerWindowsPapst1"

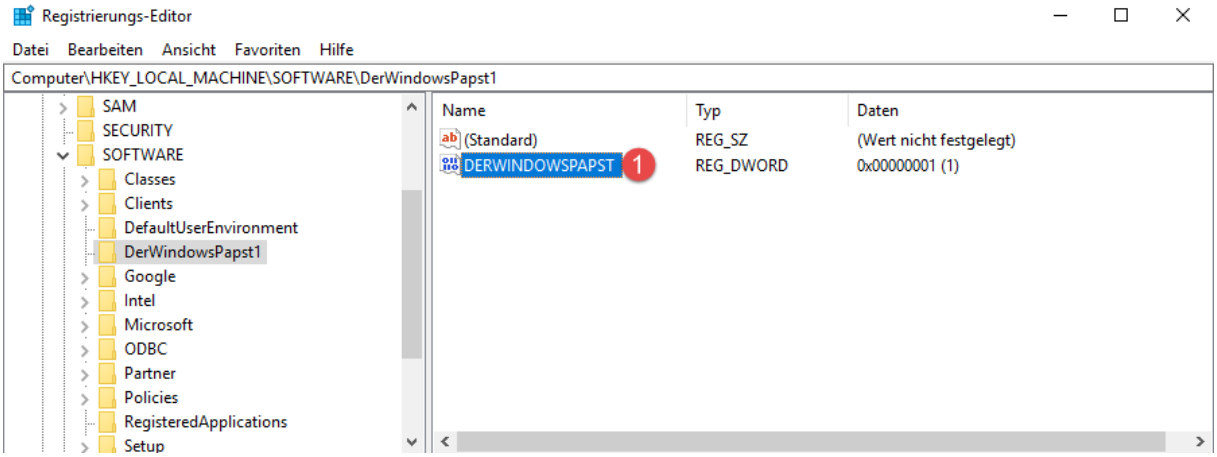


Umgang mit der Registry



Legen wir nun ein ItemProperty unterhalb des neuen Schlüssels „DerWindowsPapst1" an.

```
New-ItemProperty -path "HKLM:\Software\DerWindowsPapst1" -name
DERWINDOWSPAPST -PropertyType DWORD -value 1
```



Weitere Beispiele zu den Typen (String, DWORD, MultiString)

```
New-ItemProperty -path "HKLM:\Software\DerWindowsPapst1" -name
DERWINDOWSPAPST -PropertyType DWORD -value 1
```

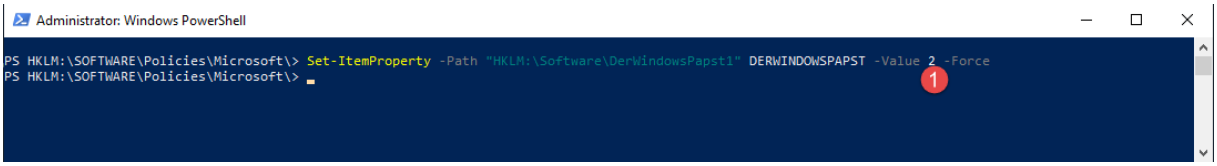
```
New-ItemProperty -path "HKLM:\Software\DerWindowsPapst1"  -name
DERWINDOWSPAPST1 -PropertyType String -value "1"
```

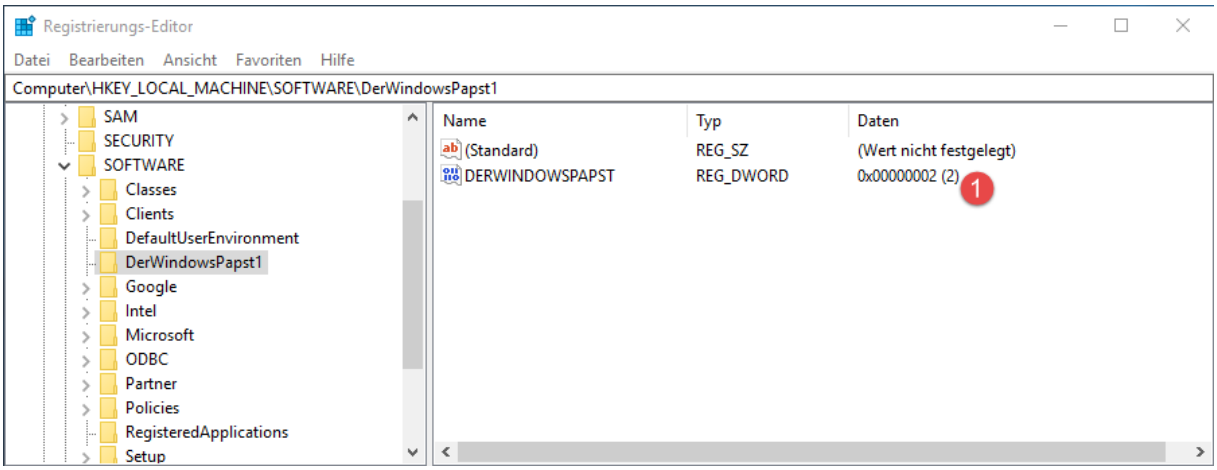
```
New-ItemProperty -path "HKLM:\Software\DerWindowsPapst1" -name
DERWINDOWSPAPST2 -PropertyType MultiString -value "Der-Windows-Papst","1","2"
```

PowerShell Type	Registry Type
Binary	REG_BINARY
DWord	REG_DWORD
ExpandString	REG_EXPAND_SZ
MultiString	REG_MULTI_SZ
None	–
QWord	REG_QWORD
String	REG_SZ
Unknown	–

Gehen wir nun über und ändern den Wert des Eintrags von 1 auf 2.

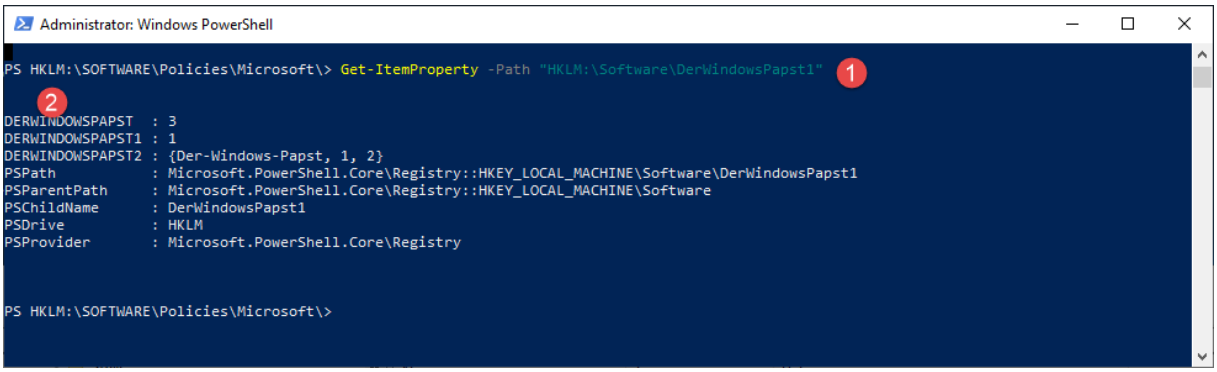
Set-ItemProperty -Path "HKLM:\Software\DerWindowsPapst1" DERWINDOWSPAPST -Value 2 –Force

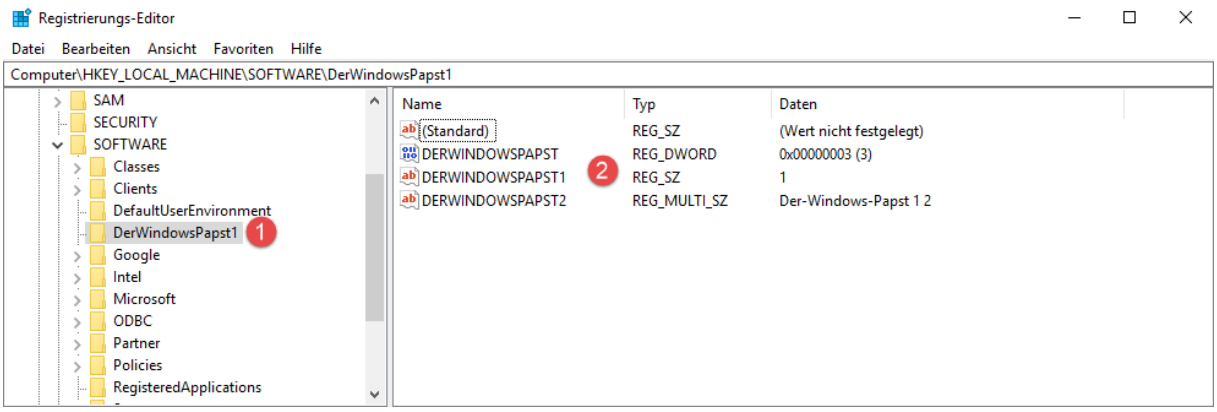




Prüfen wir doch mal eben, welche Einträge wir bereits angelegt haben.

Get-ItemProperty -Path "HKLM:\Software\DerWindowsPapst1"

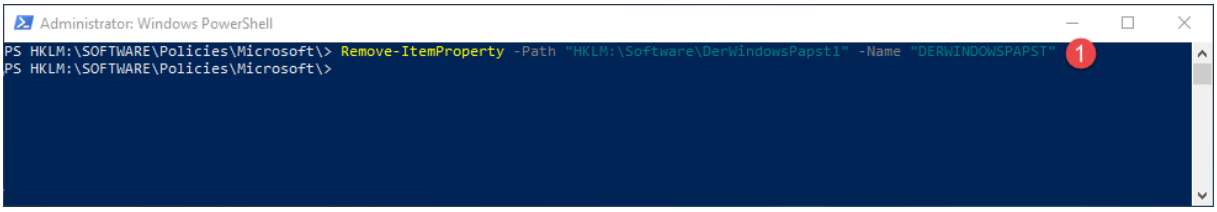


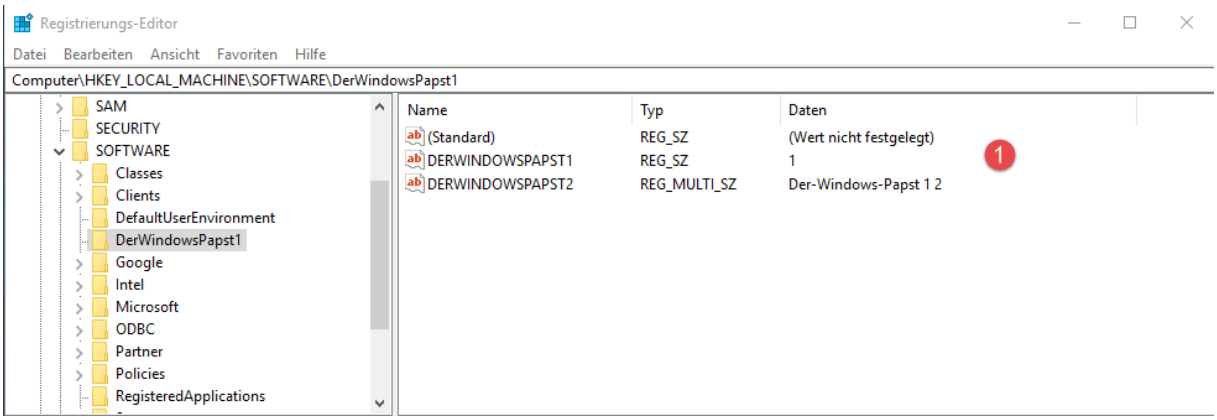


Umgang mit der Registry

Fangen wir an Einträge zu löschen.

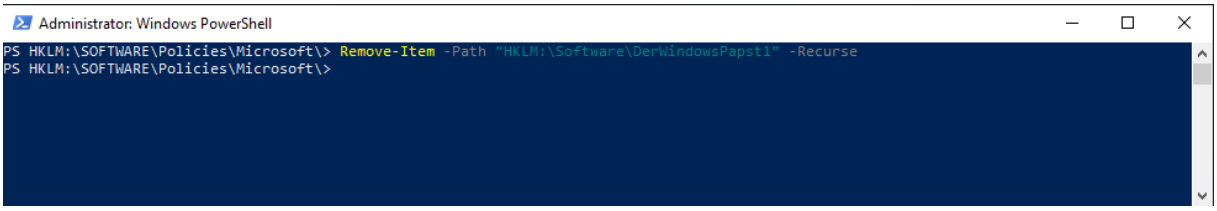
Remove-ItemProperty -Path "HKLM:\Software\DerWindowsPapst1" -Name "DERWINDOWSPAPST"





Jetzt löschen wir den ganzen Schlüssel „DerWindowsPapst1"

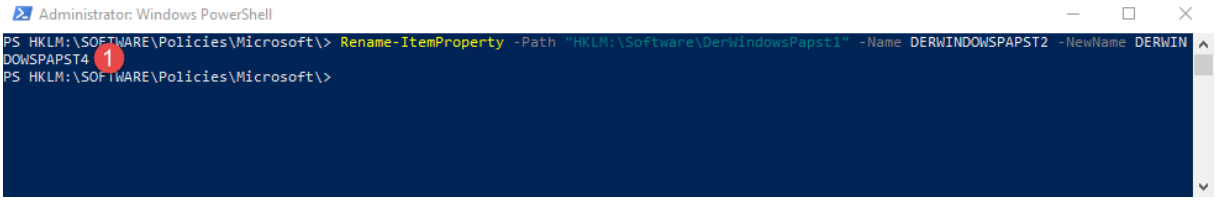
Remove-Item -Path "HKLM:\Software\DerWindowsPapst1" -Recurse



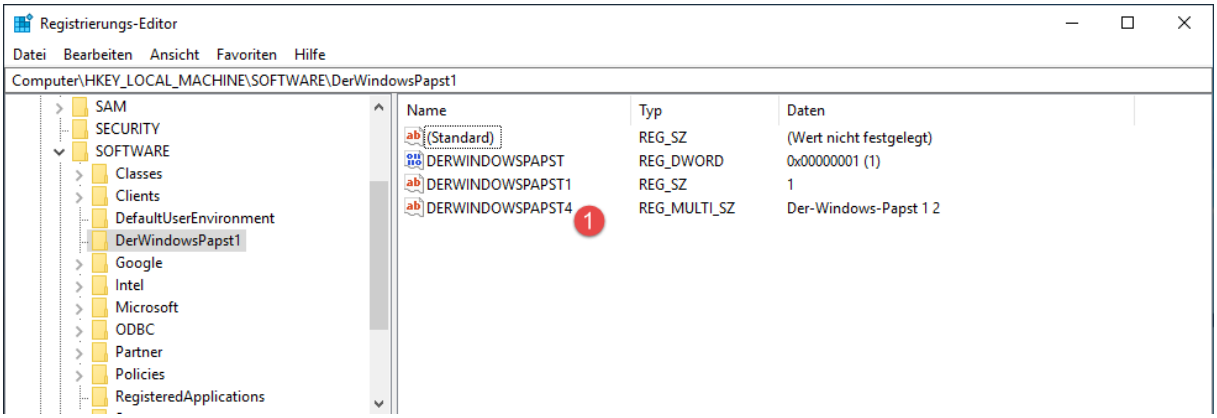
Damit es weitergehen kann, lege ich den Schlüssel wieder an, wir sind ja noch nicht fertig.

Jetzt benennen wir einen Eintrag um. Und zwar DERWINDOWSPAPST2 in DERWINDOWSPAPST4

Rename-ItemProperty -Path "HKLM:\Software\DerWindowsPapst1" -Name DERWINDOWSPAPST2 -NewName DERWINDOWSPAPST4



Umgang mit der Registry



Zum Abschluss werden wir einen ganzen Schlüssel verschieben.

Move-Item "HKLM:\Software\DerWindowsPapst1*" "HKLM:\Software\Google"