

# Windows Batch Skript mit automatischen Admin-Rechten (Elevation)

Viele Batch-Skript müssen mit erhöhten Rechten ausgeführt werden, z. B. um Prozesse zu beenden oder Dienste zu starten.

Da es etwas umständlich das Skript "Als Administrator ausführen" zu starten bzw. den Umweg über eine Verknüpfung zu dem Skript gehen. wird hier eine Lösung beschrieben, die das im Skript selbst tun kann.

```
:checkPrivileges
NET FILE 1>NUL 2>NUL
if '%errorlevel%' == '0' ( goto gotPrivileges ) else ( goto getPrivileges )

:getPrivileges
if '%1'=='ELEV' (echo ELEV & shift /1 & goto gotPrivileges)

setlocal DisableDelayedExpansion
set "batchPath=%~0"
setlocal EnableDelayedExpansion
ECHO Set UAC = CreateObject^("Shell.Application"^) > "%temp%\OEgetPrivileges.vbs"
ECHO args = "ELEV " >> "%temp%\OEgetPrivileges.vbs"
ECHO For Each strArg in WScript.Arguments >> "%temp%\OEgetPrivileges.vbs"
ECHO args = args ^& strArg ^& " " >> "%temp%\OEgetPrivileges.vbs"
ECHO Next >> "%temp%\OEgetPrivileges.vbs"
ECHO UAC.ShellExecute "!batchPath!", args, "", "runas", 1 >> "%temp%\OEgetPrivileges.vbs"
"%SystemRoot%\System32\WScript.exe" "%temp%\OEgetPrivileges.vbs" %*
exit /B

:gotPrivileges
```

Das Snippet kann einfach vor einem bestehenden Skript eingefügt werden. Es prüft, ob schon erhöhte Rechte vorhanden sind. Ist das nicht der Fall wird ein VB-Skript erzeugt, welches diese abrufen und dann das eigentliche Skript startet.

## Verwandte Artikel

- [Disable Windows Bitlocker Boot Shutdown](#)
- [Windows Batch Skript mit automatischen Admin-Rechten \(Elevation\)](#)