



Powershell Remote Taskplaner

Mit diesem Powershell Skript installieren wie über die Aufgabenplanung ein Windows Feature. Das Ganze setzt die eingerichtete Remoteverwaltung voraus.

```
Invoke-Command -ComputerName srv01 -ScriptBlock {Register-ScheduledJob -Name WSBBackupInstall -ScriptBlock {Install-WindowsFeature -ComputerName srv01 -Name Windows-Server-Backup} -Trigger (New-JobTrigger -Once -At "08/04/2017 18:00")}
```

Server DC01

```
PS C:\Windows\system32> Invoke-Command -ComputerName srv01 -ScriptBlock {Register-ScheduledJob -Name WSBBackupInstall -ScriptBlock {Install-WindowsFeature -ComputerName srv01 -Name Windows-Server-Backup} -Trigger Strigger}

PSComputerName      : srv01
RunspaceId          : 7f25297e-5ca1-47f2-84bd-6a5acc4792f3
InvocationInfo      : Microsoft.PowerShell.ScheduledJob.ScheduledJobInvocationInfo
Definition           : System.Management.Automation.JobDefinition
Options             : Microsoft.PowerShell.ScheduledJob.ScheduledJobOptions
Credential           :
JobTriggers         : [Microsoft.PowerShell.ScheduledJob.ScheduledJobTrigger]
Id                  : 1
GlobalId            : 4698be3f-fd34-4b76-9b58-047b077ac18c
Name                : WSBBackupInstall
Command             : Install-WindowsFeature -ComputerName srv01 -Name Windows-Server-Backup
ExecutionHistoryLength : 32
Enabled             : True
PSExecutionPath    : powershell.exe
PSExecutionArgs    : -NoLogo -NonInteractive -WindowStyle Hidden -Command "Import-Module PSScheduledJob; $jobDef = [Microsoft.PowerShell.ScheduledJob.ScheduledJobDefinition]::LoadFromStore('WSBackupInstall', 'C:\Users\NDO\AppData\Local\Microsoft\Windows\PowerShell\ScheduledJobs\'); $jobDef.Run()"

PS C:\Windows\system32>
```

Task remote prüfen:

```
Invoke-Command -ComputerName srv01 {Get-Job | Select-Object PSComputerName,PSBeginTime,PSEndtime; Get-WindowsFeature Windows-Server-Backup | Select-Object Name,PSComputerName,Installstate}
```

Server DC01

```
PS C:\Windows\system32> Invoke-Command -ComputerName srv01 {Get-Job | Select-Object PSComputerName,PSBeginTime,PSEndtime; Get-WindowsFeature Windows-Server-Backup | Select-Object Name,PSComputerName,Installstate}

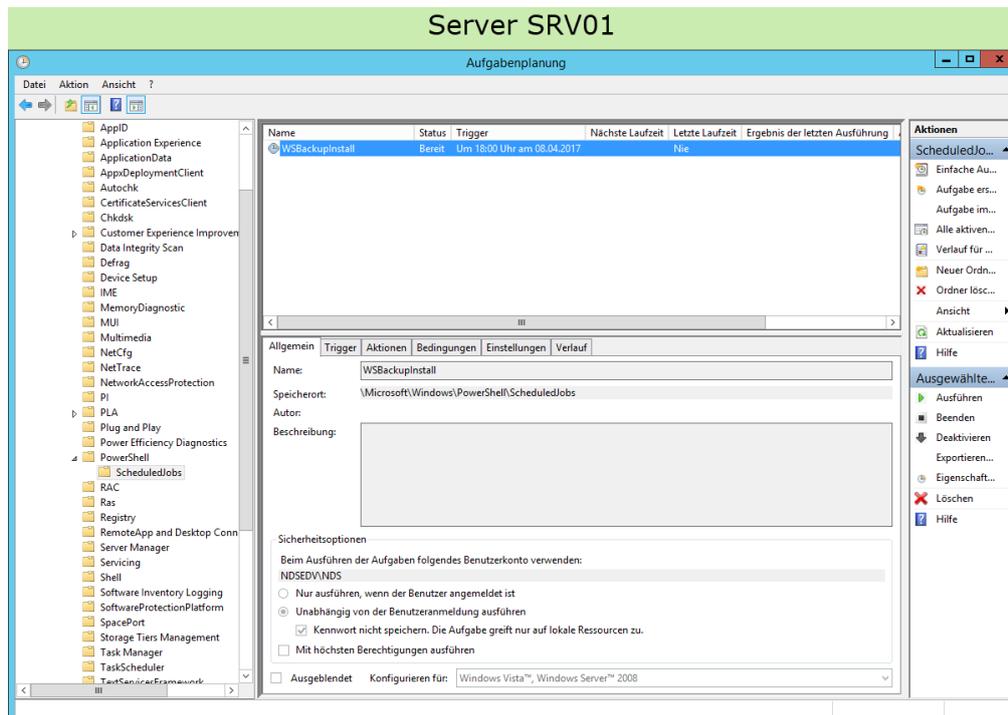
Name                : Windows-Server-Backup
InstallState        : Available
PSComputerName      : srv01
RunspaceId          : f0bec8fa-6adc-4f56-8c92-1e96ec39fa2d

PS C:\Windows\system32>
```

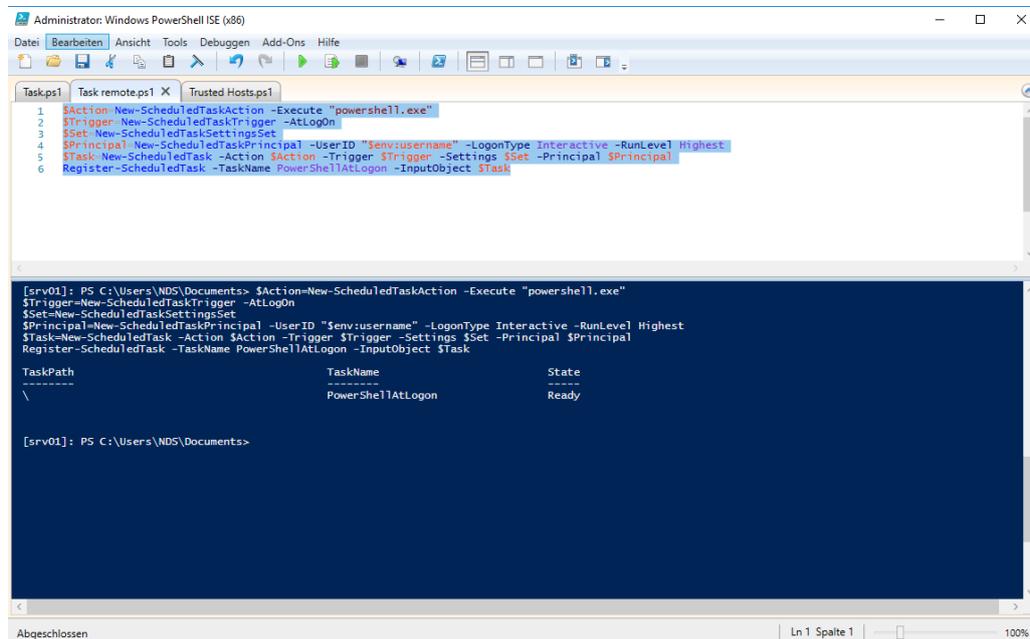


Powershell Remote Taskplaner

Auf dem Zielsystem finden wir die geplante Aufgabe unter Powershell wieder:



Mit diesem Powershell Skript erstellen wir einen Task, der die Powershell Konsole nach jedem LogOn automatisch startet.



```
$Action=New-ScheduledTaskAction -Execute "powershell.exe"
```

```
$Trigger=New-ScheduledTaskTrigger -AtLogOn
```

```
$Set=New-ScheduledTaskSettingsSet
```

```
$Principal=New-ScheduledTaskPrincipal -UserID "$env:username" -LogonType  
Interactive -RunLevel Highest
```

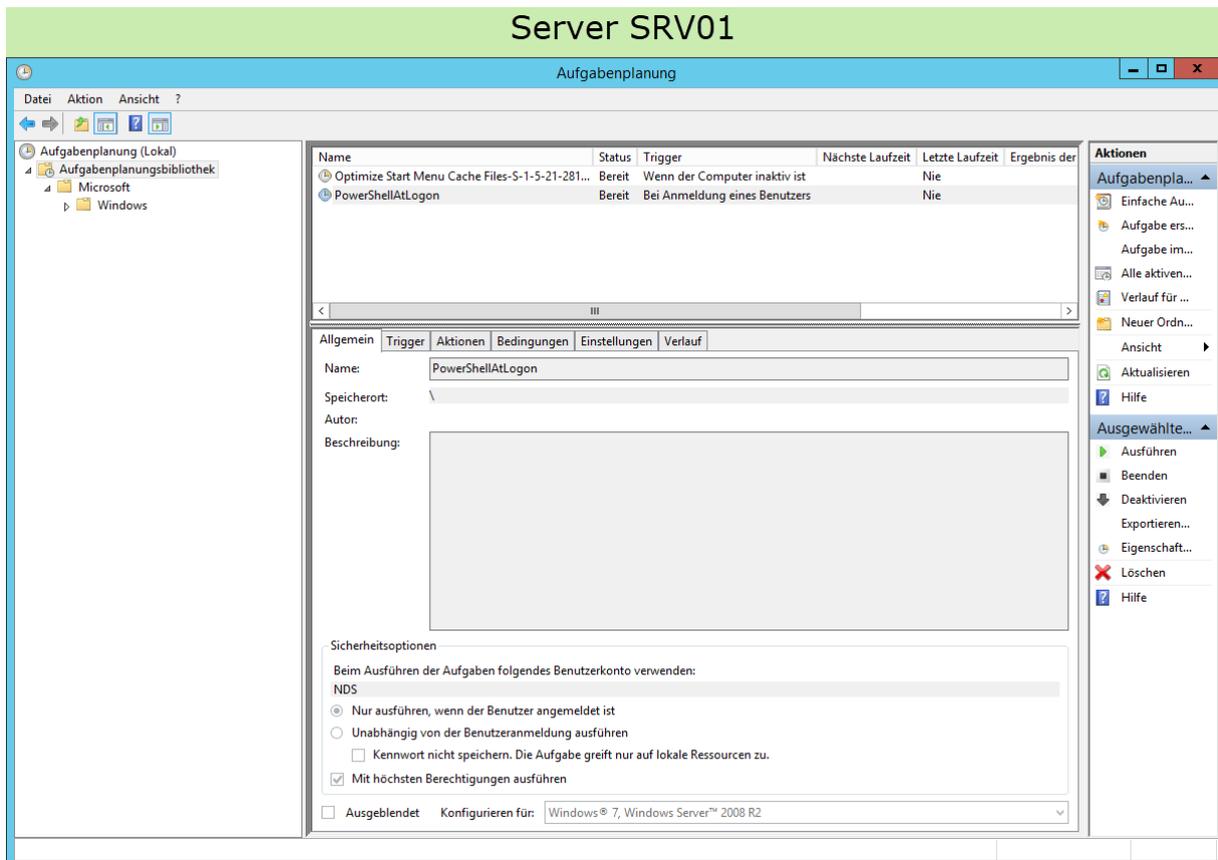
```
$Task=New-ScheduledTask -Action $Action -Trigger $Trigger -Settings $Set -Principal  
$Principal
```

```
Register-ScheduledTask -TaskName PowerShellAtLogon -InputObject $Task
```

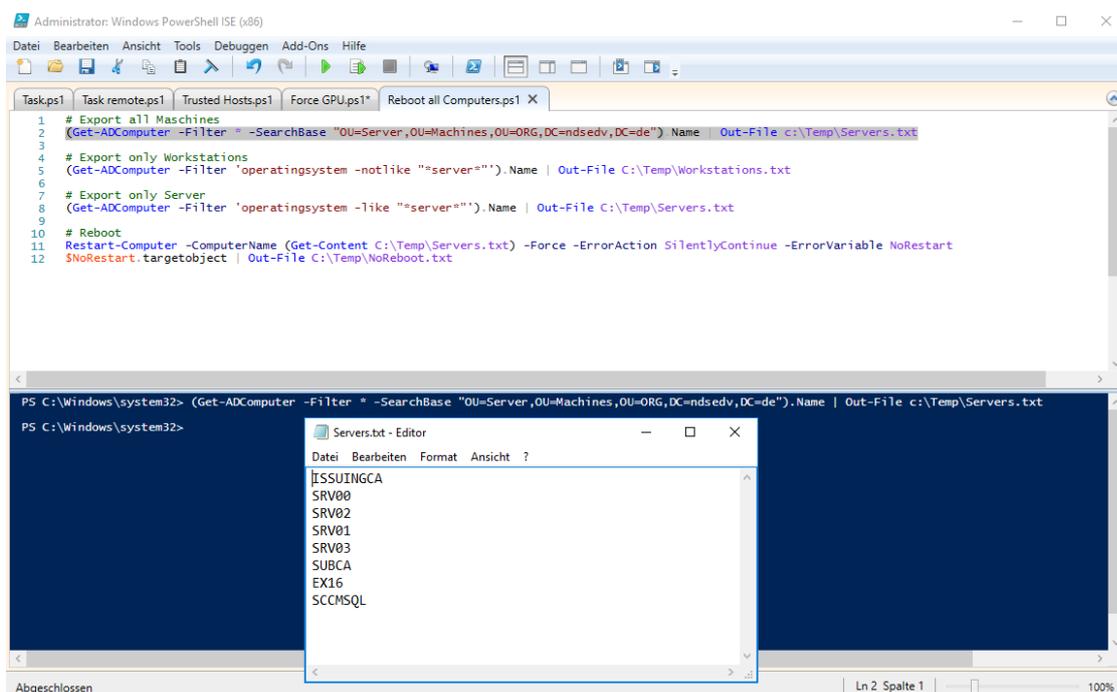


Powershell Remote Taskplaner

Den Task finden wir auf dem Zielsystem an dieser Stelle wieder:



Mit diesem Powershell Skript fahren wir entfernte Maschinen herunter. Als erstes erstellen wir eine Liste auf Basis einer OU oder eines Maschinentyps wie z.B. Server oder Workstations. Das Ganze setzt natürlich voraus das die Remoteverwaltung eingerichtet ist.





Powershell Remote Taskplaner

Export all Maschines

```
(Get-ADComputer -Filter * -SearchBase "OU=Server,OU=Machines,OU=ORG,DC=ndsedv,DC=de").Name | Out-File c:\Temp\Servers.txt
```

Export only Workstations

```
(Get-ADComputer -Filter 'operatingsystem -notlike "*server*").Name | Out-File C:\Temp\Workstations.txt
```

Export only Server

```
(Get-ADComputer -Filter 'operatingsystem -like "*server*").Name | Out-File C:\Temp\Servers.txt
```

Reboot

```
Restart-Computer -ComputerName (Get-Content C:\Temp\Servers.txt) -Force -ErrorAction SilentlyContinue -ErrorVariable NoRestart
```

```
$NoRestart.targetobject | Out-File C:\Temp\NoReboot.txt
```

```
Server DC01
Administrator: Windows PowerShell ISE (x86)
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe
Taskps1 Task remote.ps1 Trusted Hosts.ps1 Force GPU.ps1 Reboot all Computers.ps1 X
1 # Export all Maschines
2 (Get-ADComputer -Filter * -SearchBase "OU=Server,OU=Machines,OU=ORG,DC=ndsedv,DC=de").Name | Out-File c:\Temp\Servers.txt
3
4 # Export only Workstations
5 (Get-ADComputer -Filter 'operatingsystem -notlike "*server*").Name | Out-File C:\Temp\Workstations.txt
6
7 # Export only Server
8 (Get-ADComputer -Filter 'operatingsystem -like "*server*").Name | Out-File C:\Temp\Servers.txt
9
10 # Reboot
11 Restart-Computer -ComputerName (Get-Content C:\Temp\Servers.txt) -Force -ErrorAction SilentlyContinue -ErrorVariable NoRestart
12 $NoRestart.targetobject | Out-File C:\Temp\NoReboot.txt
PS C:\Windows\system32> (Get-ADComputer -Filter * -SearchBase "OU=Server,OU=Machines,OU=ORG,DC=ndsedv,DC=de").Name | Out-File c:\Temp\Servers.txt
PS C:\Windows\system32> Restart-Computer -ComputerName (Get-Content C:\Temp\Servers.txt) -Force -ErrorAction SilentlyContinue -ErrorVariable NoRestart
$NoRestart.targetobject | Out-File C:\Temp\NoReboot.txt
PS C:\Windows\system32>
```



Powershell Remote Taskplaner



SRV01 wird neu gestartet und steht auch nicht in der Liste NoReboots.txt

Server DC01

```
Administrator: Windows PowerShell ISE (x86)
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe
Task.ps1 Task remote.ps1 Trusted Hosts.ps1 Force GPU.ps1* Reboot all Computers.ps1 X
1 # Export all Maschines
2 (Get-ADComputer -Filter * -SearchBase "OU=Server,OU=Machines,OU=ORG,DC=ndsedv,DC=de") Name | Out-File c:\Temp\Servers.txt
3
4 # Export only Workstations
5 (Get-ADComputer -Filter 'operatingsystem -notlike "*server*") Name | Out-File C:\Temp\Workstations.txt
6
7 # Export only Server
8 (Get-ADComputer -Filter 'operatingsystem -like "*server*") Name | Out-File C:\Temp\Servers.txt
9
10 # Reboot
11 Restart-Computer -ComputerName (Get-Content C:\Temp\Servers.txt) -Force -ErrorAction SilentlyContinue -ErrorVariable NoRestart
12 $NoRestart.targetobject | Out-File C:\Temp\NoReboot.txt

PS C:\Windows\system32> (Get-ADComputer -Filter * -SearchBase "OU=Server,OU=Machines,OU=ORG,DC=ndsedv,DC=de").Name | Out-File c:\Temp\Servers.txt
PS C:\Windows\system32> Restart-Computer -ComputerName (Get-Content C:\Temp\Servers.txt) -Force -ErrorAction SilentlyContinue -ErrorVariable NoRestart
$NoRestart.targetobject | Out-File C:\Temp\NoReboot.txt
PS C:\Windows\system32>

NoReboot.txt - Editor
Datei Bearbeiten Format Ansicht ?
ISSUINGCA
SRV00
SRV02
SRV03
SUBCA
EX16
SCCMSQL
```