

# Set-ExecutionPolicy

Referenz

 [Feedback](#)

Modul: [Microsoft.PowerShell.Security](#)

## In diesem Artikel

- [Syntax](#)
- [Beschreibung](#)
- [Beispiele](#)
- [Parameter](#)
- [4 weitere anzeigen](#)

Legt die PowerShell-Ausführungsrichtlinien für Windows-Computer fest.

## Syntax

```
Set-ExecutionPolicy
  [-ExecutionPolicy] <ExecutionPolicy>
  [[-Scope] <ExecutionPolicyScope>]
  [-Force]
  [-WhatIf]
  [-Confirm]
  [<CommonParameters>]
```

## Beschreibung

Das `Set-ExecutionPolicy` Cmdlet ändert PowerShell-Ausführungsrichtlinien für Windows-Computer. Weitere Informationen finden Sie unter [Informationen zu Ausführungsrichtlinien](#).

Ab PowerShell 6.0 für Nicht-Windows-Computer ist `Unrestricted` die Standardausführungsrichtlinie und kann nicht geändert werden. Das `Set-ExecutionPolicy` Cmdlet ist verfügbar, aber PowerShell zeigt eine Konsolenmeldung an, dass es nicht unterstützt wird.

Eine Ausführungsrichtlinie ist Teil der PowerShell-Sicherheitsstrategie. Ausführungsrichtlinien bestimmen, ob Sie Konfigurationsdateien wie Ihr PowerShell-Profil laden oder Skripts ausführen können. Und ob Skripts digital signiert werden müssen, bevor sie ausgeführt werden.

Der `Set-ExecutionPolicy` Standardbereich des Cmdlets ist `LocalMachine`, was sich auf alle Benutzer des Computers auswirkt. Um die Ausführungsrichtlinie für `LocalMachine` zu ändern, starten Sie PowerShell mit "**Als Administrator** ausführen".

Um die Ausführungsrichtlinien für jeden Bereich anzuzeigen, verwenden Sie `Get-ExecutionPolicy -List`. Um die effektive Ausführungsrichtlinie für Ihre PowerShell-Sitzung `Get-ExecutionPolicy` ohne Parameter anzuzeigen.

# Beispiele

## Beispiel 1: Festlegen einer Ausführungsrichtlinie

In diesem Beispiel wird gezeigt, wie die Ausführungsrichtlinie für den lokalen Computer festgelegt wird.

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine
Get-ExecutionPolicy -List

Scope ExecutionPolicy
-----
MachinePolicy          Undefined
    UserPolicy          Undefined
        Process          Undefined
            CurrentUser    RemoteSigned
                LocalMachine RemoteSigned
```

Das `Set-ExecutionPolicy` Cmdlet verwendet den **ExecutionPolicy-Parameter** , um die `RemoteSigned` Richtlinie anzugeben. Der **Parameter Scope** gibt den Standardbereichswert an. `LocalMachine` Verwenden Sie das `Get-ExecutionPolicy` Cmdlet mit dem **Parameter List** , um die Ausführungsrichtlinieneinstellungen anzuzeigen.

## Beispiel 2: Festlegen einer Ausführungsrichtlinie, die mit einer Gruppenrichtlinie in Konflikt ist

Dieser Befehl versucht, die Ausführungsrichtlinie des `LocalMachine` Bereichs auf `Restricted` `LocalMachine` ist restriktiver, aber nicht die effektive Richtlinie, da sie mit einer Gruppenrichtlinie in Konflikt steht. Die `Restricted` Richtlinie wird in die Registrierungsstruktur `HKEY_LOCAL_MACHINE` geschrieben.

```
PS> Set-ExecutionPolicy -ExecutionPolicy Restricted -Scope LocalMachine

Set-ExecutionPolicy : PowerShell updated your local preference successfully, but overridden by the Group Policy applied to your system. Due to the override, your its current effective execution policy of "AllSigned". Contact your Group Policy more information. At line:1 char:20 + Set-ExecutionPolicy <<<< restricted

PS> Get-ChildItem -Path HKLM:\SOFTWARE\Microsoft\PowerShell\1\ShellIds

Hive: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PowerShell\1\ShellIds

Name                Property
----                -
Microsoft.PowerShell Path           : C:\Windows\System32\WindowsPowerShell\
ExecutionPolicy      : Restricted
ScriptedDiagnostics ExecutionPolicy : Unrestricted
```

Das `Set-ExecutionPolicy` Cmdlet verwendet den **ExecutionPolicy-Parameter** , um die `Restricted` Richtlinie anzugeben. Der **Parameter Scope** gibt den Standardbereichswert an. `LocalMachine` Das `Get-ChildItem` Cmdlet verwendet den **Parameter "Path "** mit dem `HKLM:` Laufwerk, um den Registrierungsspeicherort anzugeben.

## Beispiel 3: Anwenden der Ausführungsrichtlinie von einem Remotecomputer auf einen lokalen Computer

Dieser Befehl ruft das Ausführungsrichtlinienobjekt von einem Remotecomputer ab und legt die Richtlinie auf dem lokalen Computer fest. `Get-ExecutionPolicy` sendet ein **Microsoft.PowerShell.ExecutionPolicy** -Objekt nach unten in der Pipeline. `Set-ExecutionPolicy` akzeptiert Pipelineeingaben und erfordert nicht den **ExecutionPolicy-Parameter** .

```
Invoke-Command -ComputerName Server01 -ScriptBlock { Get-ExecutionPolicy } | Set-ExecutionPolicy
```

Das `Invoke-Command` Cmdlet wird auf dem lokalen Computer ausgeführt und sendet `scriptBlock` an den Remotecomputer. Der **Parameter ComputerName** gibt den Remotecomputer Server01 an. Der **ScriptBlock-Parameter** wird auf dem Remotecomputer ausgeführt `Get-ExecutionPolicy` . Das `Get-ExecutionPolicy` Objekt wird an die `Set-ExecutionPolicy` Pipeline gesendet. `Set-ExecutionPolicy` wendet die Ausführungsrichtlinie auf den Standardbereich des lokalen Computers an. `LocalMachine`

## Beispiel 4: Festlegen des Bereichs für eine Ausführungsrichtlinie

In diesem Beispiel wird gezeigt, wie Eine Ausführungsrichtlinie für einen angegebenen Bereich festgelegt wird. `CurrentUser` Der `CurrentUser` Bereich betrifft nur den Benutzer, der diesen Bereich festlegt.

```
Set-ExecutionPolicy -ExecutionPolicy AllSigned -Scope CurrentUser
Get-ExecutionPolicy -List

Scope ExecutionPolicy
-----
MachinePolicy          Undefined
    UserPolicy          Undefined
        Process          Undefined
            CurrentUser    AllSigned
                LocalMachine RemoteSigned
```

`Set-ExecutionPolicy` verwendet den **ExecutionPolicy-Parameter** , um die `AllSigned` Richtlinie anzugeben. Der **Parameter Scope** gibt die `CurrentUser` . Verwenden Sie das `Get-ExecutionPolicy` Cmdlet mit dem **Parameter List** , um die Ausführungsrichtlinieneinstellungen anzuzeigen.

Die effektive Ausführungsrichtlinie für den Benutzer wird `AllSigned` .

## Beispiel 5: Entfernen der Ausführungsrichtlinie für den aktuellen Benutzer

In diesem Beispiel wird gezeigt, wie die `Undefined` Ausführungsrichtlinie zum Entfernen einer Ausführungsrichtlinie für einen bestimmten Bereich verwendet wird.

```
Set-ExecutionPolicy -ExecutionPolicy Undefined -Scope CurrentUser
Get-ExecutionPolicy -List

Scope ExecutionPolicy
-----
MachinePolicy          Undefined
    UserPolicy          Undefined
        Process          Undefined
```

CurrentUser	Undefined
LocalMachine	RemoteSigned

`Set-ExecutionPolicy` verwendet den **ExecutionPolicy-Parameter** , um die `Undefined` Richtlinie anzugeben. Der **Parameter Scope** gibt die `CurrentUser` . Verwenden Sie das `Get-ExecutionPolicy` Cmdlet mit dem **Parameter List** , um die Ausführungsrichtlinieneinstellungen anzuzeigen.

## Beispiel 6: Festlegen der Ausführungsrichtlinie für die aktuelle PowerShell-Sitzung

Der `Process` Bereich wirkt sich nur auf die aktuelle PowerShell-Sitzung aus. Die Ausführungsrichtlinie wird in der Umgebungsvariablen `$env:PSExecutionPolicyPreference` gespeichert und beim Schließen der Sitzung gelöscht.

```
Set-ExecutionPolicy -ExecutionPolicy AllSigned -Scope Process

Scope ExecutionPolicy
-----
MachinePolicy          Undefined
    UserPolicy          Undefined
        Process          AllSigned
    CurrentUser          RemoteSigned
    LocalMachine          RemoteSigned
```

Der `Set-ExecutionPolicy` Parameter `ExecutionPolicy` wird verwendet, um die `AllSigned` Richtlinie anzugeben. Der **Parameter Scope** gibt den Wert `Process` an. Verwenden Sie das `Get-ExecutionPolicy` Cmdlet mit dem **Parameter List** , um die Ausführungsrichtlinieneinstellungen anzuzeigen.

## Beispiel 7: Aufheben der Blockierung eines Skripts, um es auszuführen, ohne die Ausführungsrichtlinie zu ändern

Dieses Beispiel zeigt, wie die `RemoteSigned` Ausführungsrichtlinie verhindert, dass Sie nicht signierte Skripts ausführen.

Es empfiehlt sich, den Code des Skripts zu lesen und zu überprüfen, ob es vor der Verwendung des `Unblock-File` Cmdlets sicher ist. Das `Unblock-File` Cmdlet hebt die Blockierung von Skripts auf, sodass sie ausgeführt werden können, ändert jedoch nicht die Ausführungsrichtlinie.

```
PS> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine

PS> Get-ExecutionPolicy

RemoteSigned

PS> .\Start-ActivityTracker.ps1

.\Start-ActivityTracker.ps1 : File .\Start-ActivityTracker.ps1 cannot be loaded.
The file .\Start-ActivityTracker.ps1 is not digitally signed.
The script will not execute on the system.
For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=294628
At line:1 char:1
+ .\Start-ActivityTracker.ps1
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

```
PS> Unblock-File -Path .\Start-ActivityTracker.ps1

PS> Get-ExecutionPolicy

RemoteSigned

PS> .\Start-ActivityTracker.ps1

Task 1:
```

Der `Set-ExecutionPolicy` Parameter `ExecutionPolicy` **wird** verwendet, um die `RemoteSigned` Richtlinie anzugeben. Die Richtlinie wird für den Standardbereich festgelegt. `LocalMachine`

Das `Get-ExecutionPolicy` Cmdlet zeigt, dass `RemoteSigned` es sich um die effektive Ausführungsrichtlinie für die aktuelle PowerShell-Sitzung handelt.

`RemoteSigned` `Start-ActivityTracker.ps1` script is executed from the current directory. The script is blocked by , weil das Skript nicht digital signiert ist.

In diesem Beispiel wurde der Code des Skripts überprüft und als sicher ausgeführt. Das `Unblock-File` Cmdlet verwendet den **Path-Parameter** , um die Blockierung des Skripts aufzuheben.


Um zu überprüfen, ob `Unblock-File` die Ausführungsrichtlinie nicht geändert wurde, `Get-ExecutionPolicy` wird die effektive Ausführungsrichtlinie angezeigt. `RemoteSigned`

Das Skript `Start-ActivityTracker.ps1` wird aus dem aktuellen Verzeichnis ausgeführt. Das Skript beginnt mit der Ausführung, da es vom `Unblock-File` Cmdlet entsperrt wurde.

## Parameter

### -Confirm

Hiermit werden Sie vor der Ausführung des Cmdlets zur Bestätigung aufgefordert.

 **Tabelle erweitern**

Type:	<a href="#">SwitchParameter</a>
Aliases:	cf
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

### -ExecutionPolicy

Gibt die Ausführungsrichtlinie an. Wenn keine Gruppenrichtlinien vorhanden sind und die Ausführungsrichtlinie jedes Bereichs auf `Undefined` festgelegt ist, wird die `Restricted` effektive Richtlinie für alle Benutzer.

Die zulässigen Werte für die Ausführungsrichtlinie sind wie folgt:

- `AllSigned`. Erfordert, dass alle Skripts und Konfigurationsdateien von einem vertrauenswürdigen Herausgeber signiert werden, einschließlich Skripts, die auf dem lokalen Computer geschrieben wurden.

- `Bypass` . Es findet keine Blockierung statt und es werden keine Warnungen oder Eingabeaufforderungen ausgegeben.
- `Default` . Legt die Standardausführungsrichtlinie fest. `Restricted` für Windows-Clients oder `RemoteSigned` für Windows-Server.
- `RemoteSigned` . Erfordert, dass alle skripts und Konfigurationsdateien, die aus dem Internet heruntergeladen wurden, von einem vertrauenswürdigen Herausgeber signiert sind. Die Standardausführungsrichtlinie für Windows-Servercomputer.
- `Restricted` . Lädt keine Konfigurationsdateien oder führt Skripts aus. Die Standardausführungsrichtlinie für Windows-Clientcomputer.
- `Undefined` . Für den Bereich ist keine Ausführungsrichtlinie festgelegt. Entfernt eine zugewiesene Ausführungsrichtlinie aus einem Bereich, der nicht durch eine Gruppenrichtlinie festgelegt ist. Wenn die Ausführungsrichtlinie in allen Bereichen lautet, lautet `Undefined Restricted` die effektive Ausführungsrichtlinie .
- `Unrestricted` . Ab PowerShell 6.0 ist dies die Standardausführungsrichtlinie für Nicht-Windows-Computer und kann nicht geändert werden. Lädt alle Konfigurationsdateien und führt alle Skripts aus. Wenn Sie ein nicht signiertes Skript ausführen, das aus dem Internet heruntergeladen wurde, werden Sie vor der Ausführung zur Berechtigung aufgefordert.

[🔗](#) Tabelle erweitern

Type:	<code>ExecutionPolicy</code>
Accepted values:	AllSigned, Bypass, Default, RemoteSigned, Restricted, Undefined, Unrestricted
Position:	0
Default value:	None
Required:	True
Accept pipeline input:	True
Accept wildcard characters:	False

**-Force**

Unterdrückt alle Bestätigungsaufforderungen. Verwenden Sie mit diesem Parameter Vorsicht, um unerwartete Ergebnisse zu vermeiden.

[🔗](#) Tabelle erweitern

Type:	<code>SwitchParameter</code>
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

**-Scope**

Gibt den Bereich an, der von einer Ausführungsrichtlinie betroffen ist. Der Standardbereich ist `LocalMachine` .

Die effektive Ausführungsrichtlinie wird wie folgt durch die Reihenfolge der Rangfolge bestimmt:

- `MachinePolicy` – Durch eine Gruppenrichtlinie für alle Benutzer des Computers festgelegt
- `UserPolicy` – Durch eine Gruppenrichtlinie für den aktuellen Benutzer des Computers festgelegt
- `Process` – Betrifft nur die aktuelle PowerShell-Sitzung
- `LocalMachine` - Standardbereich, der sich auf alle Benutzer des Computers auswirkt
- `CurrentUser` – Betrifft nur den aktuellen Benutzer

Der `Process` Bereich wirkt sich nur auf die aktuelle PowerShell-Sitzung aus. Die Ausführungsrichtlinie wird in der Umgebungsvariable `$env:PSExecutionPolicyPreference` anstelle der Registrierung gespeichert. Wenn die PowerShell-Sitzung geschlossen wird, werden die Variable und der Wert gelöscht.

Ausführungsrichtlinien für den `CurrentUser` Bereich werden in die Registrierungsstruktur `HKEY_LOCAL_USER` geschrieben.

Ausführungsrichtlinien für den `LocalMachine` Bereich werden in die Registrierungsstruktur `HKEY_LOCAL_MACHINE` geschrieben.

 [Tabelle erweitern](#)

Type:	<code>ExecutionPolicyScope</code>
Accepted values:	CurrentUser, LocalMachine, MachinePolicy, Process, UserPolicy
Position:	1
Default value:	LocalMachine
Required:	False
Accept pipeline input:	True
Accept wildcard characters:	False

**-WhatIf**

Zeigt, was geschieht, wenn das Cmdlet ausgeführt wird. Das Cmdlet wird nicht ausgeführt.

 [Tabelle erweitern](#)

Type:	<code>SwitchParameter</code>
Aliases:	<code>wi</code>
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

## Eingaben

**`ExecutionPolicy`**

Sie können ein Ausführungsrichtlinienobjekt an dieses Cmdlet weiterleiten.

**`String`**

Sie können eine Zeichenfolge mit dem Namen einer Ausführungsrichtlinie an dieses Cmdlet weiterleiten.

# Ausgaben

None

Dieses Cmdlet gibt keine Ausgabe zurück.

## Hinweise

`Set-ExecutionPolicy` ändert die `MachinePolicy` Bereiche `UserPolicy` und Bereiche nicht, da sie von Gruppenrichtlinien festgelegt werden.

`Set-ExecutionPolicy` überschreibt keine Gruppenrichtlinie, auch wenn die Benutzereinstellung restriktiver als die Richtlinie ist.

Wenn die Gruppenrichtlinie "**Skriptausführung** aktivieren" für den Computer oder Benutzer aktiviert ist, wird die Benutzereinstellung gespeichert, ist aber nicht wirksam. PowerShell zeigt eine Meldung an, die den Konflikt erläutert.

## Ähnliche Themen

- [about\\_Execution\\_Policies](#)
- [about\\_Group\\_Policy\\_Settings](#)
- [about\\_Providers](#)
- [Get-AuthenticodeSignature](#)
- [Get-ChildItem](#)
- [Get-ExecutionPolicy](#)
- [Invoke-Command](#)
- [Set-AuthenticodeSignature](#)
- [Unblock-File](#)



### Zusammenarbeit auf GitHub

Die Quelle für diesen Inhalt finden Sie auf GitHub, wo Sie auch Issues und Pull Requests erstellen und überprüfen können. Weitere Informationen finden Sie in unserem [Leitfaden für Mitwirkende](#).



### Feedback zu PowerShell

PowerShell ist ein Open Source-Projekt. Wählen Sie einen Link aus, um Feedback zu geben:

[Problem in der Dokumentation öffnen](#)

[Abgeben von Produktfeedback](#)