

Join-Path

Das Cmdlet `Join-Path` kombiniert mehrere Pfadsegmente zu einem gültigen Dateisystempfad.

Es wird verwendet, um:

- Pfade plattformunabhängig zusammensetzen
- manuelles Hantieren mit Trennzeichen (`\` oder `/`) zu vermeiden
- Fehler durch doppelte oder fehlende Separatoren zu verhindern

Syntax

```
Join-Path [-Path] <String> [-ChildPath] <String> [[-AdditionalChildPath] <String[]>]  
[<CommonParameters>]
```

`-Path`

- **Typ:** `String`
- **Pflicht:** Ja
- Basispfad (z. B. Verzeichnis)

`-ChildPath`

- **Typ:** `String`
- **Pflicht:** Ja
- Pfadsegment, das an `-Path` angehängt wird

```
Join-Path -Path "C:\Temp" -ChildPath "Datei.txt"
```

`-AdditionalChildPath`

Verfügbar ab: **PowerShell 6+**

- **Typ:** `String[]`

- **Pflicht:** Nein
- Weitere Pfadsegmente, die nacheinander angehängt werden

```
Join-Path -Path "C:\Temp" -ChildPath "Logs" -AdditionalChildPath "2026","April"
```

Hinweise zur Verwendung

- Trennzeichen werden automatisch korrekt gesetzt (kein manuelles `\` nötig)
- Funktioniert providerübergreifend (z. B. Registry, Zertifikate)
- Mehrere Segmente werden sauber zusammengeführt
- Bestehende Trennzeichen im Input werden berücksichtigt (keine doppelten `\\`)

Verhalten

Eigenschaft	Beschreibung
Plattformabhängigkeit	Berücksichtigt das jeweilige Dateisystem
Rückgabewert	<code>String</code> (zusammengesetzter Pfad)
Validierung	Keine Existenzprüfung des Pfades
Separator-Handling	Automatisch korrekt

Beispiele

Einfaches Zusammenfügen

```
Join-Path -Path "C:\Temp" -ChildPath "Datei.txt"
```

Ergebnis: `C:\Temp\Datei.txt`

Mehrere Segmente

```
Join-Path -Path "C:\Temp" -ChildPath "Logs" -AdditionalChildPath "2026","April"
```

Ergebnis: `C:\Temp\Logs\2026\April`

Mit Variablen

```
$base = "C:\Temp"  
$file = "report.txt"  
  
Join-Path -Path $base -ChildPath $file
```

Ergebnis: `C:\Temp\report.txt`

Provider-unabhängig (z. B. Registry)

```
Join-Path -Path "HKCU:\Software" -ChildPath "Microsoft"
```

Typische Anwendungsfälle

- Dynamische Dateipfade erstellen
 - Arbeiten mit temporären Verzeichnissen
 - Plattformunabhängige Skripte schreiben
 - Zusammenbau von Registry-Pfaden
-

Alternativen / Ergänzungen

`String`-Konkatenation

```
# ❌ Fehleranfällig  
"C:\Temp\" + "Datei.txt"
```

- Fehleranfällig bei fehlenden oder doppelten Trennzeichen
- Nicht plattformunabhängig

[System.IO.Path]::Combine()

```
[System.IO.Path]::Combine("C:\Temp", "Datei.txt")
```

Unterschiede:

- .NET-Methode, nicht PowerShell-spezifisch
- Keine Provider-Unterstützung (nur Dateisystem)
- Kein Support für `CommonParameters`

📌 Best Practices

- Immer `Join-Path` statt String-Konkatenation verwenden
- Ab PowerShell 6 kann für mehrere Segmente `-AdditionalChildPath` verwendet werden.
- Kombination mit `Test-Path` für Existenzprüfung
- Variablen statt Hardcoding verwenden

Wenn du Pfade immer noch per String zusammenklebst, dann sabotierst du dich halt selbst mit Ansage. Funktioniert kurz, bricht später, und dann suchst du den Fehler wie ein Detektiv ohne Kaffee. Nimm einfach `Join-Path` und erspar dir das Drama.

Revision #5

Created 2026-04-24 11:54:49 UTC by John-Andreas Borinas

Updated 2026-05-15 04:43:06 UTC by John-Andreas Borinas