

Move-Item, przeniesienie-elementów z poświadczeniami w PowerShell

Poznasz PowerShell Move-Item. Polecenie cmdlet Move-Item w Powershell działa podobnie do opcji Wytnij/Wklej w systemie Windows.

Przenosi pliki, foldery, rejestr oraz ich właściwości, zawartość i elementy podrzędne z jednej lokalizacji do drugiej.

Lokalizacja musi być obsługiwana przez tego samego dostawcę. Kiedy przenosisz elementy z jednej lokalizacji do drugiej, to w zasadzie usuwa elementy z oryginalnej lokalizacji i przenosi się do innej lokalizacji.

Składnia

Move-Item

[-Path] <String[]>

[-Destination] <String> [-Force] [-Filter <String>] [-Include <String[]>] [-Exclude <String[]>] [-PassThru] [-Credential <PSCredential>] [-WhatIf] [-Confirm] [<CommonParameters>]

Poniżej znajduje się parametr używany w PowerShell Move-Item:

- **Path:** Określa lokalizację źródłową pliku. Katalogiem domyślnym jest katalog bieżący. Możesz również określić (.) jako bieżącą lokalizację. Parametry blankietowe są dozwolone. Możesz użyć symbolu wieloznacznego (*), aby odnieść się do wszystkich elementów w katalogu.
- **LiternalPath:** Prawie to samo co parametr -path, z tym wyjątkiem, że musisz wpisać dokładną wartość ścieżki, ponieważ nie są dozwolone żadne parametry wieloznaczne. Żadne znaki nie są interpretowane jako symbol wieloznaczny. Jeśli chcesz użyć znaków ucieczki, użyj ich pod pojedynczym cudzysłowem. Gdy znajdują się w jednym cudzysłowie, program PowerShell traktuje znak ucieczki jako ciąg. Aby dowiedzieć się więcej o postaciach ucieczki (specjalnych). <https://docs.microsoft.com/>
- **Destination:** Ścieżka elementów, które mają trafić do żądanej lokalizacji. Jeśli ścieżka docelowa nie jest dostępna, zgłosi wyjątek, ale możesz zmienić nazwę pliku, aby podać nową nazwę pliku w samym parametrze docelowym. Domyślnym miejscem docelowym jest bieżąca lokalizacja i znaki wieloznaczne są dozwolone, ale wynik musi określać jedną lokalizację.
- **Force:** Przenieś elementy do folderu docelowego bez potwierdzenia użytkownika. Ponadto, jeśli jakiegokolwiek zduplikowane elementy zostaną znalezione w lokalizacji docelowej, nadpisuje je, jeśli uprawnienia bezpieczeństwa są spełnione. Może również przenieść plik systemowy tylko do odczytu.

- **Include:** Dołącza określone nazwy plików (C:\temp\test*) lub rozszerzenia (*.pdf) ze ścieżki źródłowej, aby przenieść je do lokalizacji docelowej. Dozwolone są symbole wieloznaczne. Parametr Include jest pomocny, gdy określona zawartość ścieżki źródłowej. Na przykład w ścieżce źródłowej c:\temp* uwzględni wszystkie pliki z folderu c:\temp, a następnie niezależnie od filtrów ustawionych w parametrze Include Move-Item cmdlet będzie działać odpowiednio.
- **Exclude:** wyklucza określone nazwy plików (c:\temp\test*) lub rozszerzenia (*.txt) ze ścieżki źródłowej, dzięki czemu nie zostaną przeniesione do lokalizacji docelowej. Parametry dzikiej karty są dozwolone. Ponownie musisz użyć całego folderu lub zawartości rejestru, aby odfiltrować określone dane.
- **Filter:** parametr Filtr jest bardziej wydajny niż parametr Uwzględnij i Wyklucz, ponieważ stosuje filtr podczas pobierania obiektów, a nie filtruje obiekty po pobraniu wartości. Dokładna składnia filtra, na przykład obsługa symboli wieloznacznych, zależy od dostawcy.
- **PassThru:** Gdy używany jest parametr PassThru, dane wyjściowe polecenia wracają do konsoli PowerShell.
- **Credential:** Jeśli katalogi źródłowe i docelowe znajdują się w różnych domenach lub stacjach roboczych i używają różnych poświadczeń, należy określić poświadczenia hosta docelowego. Możesz przechowywać poświadczenia w zmiennej za pomocą Get-Credential i podać je do parametru poświadczeń. Na przykład \$cred = Get-Credential. Jeśli podasz nazwę użytkownika po parametrze -Credential, zostanie wyświetlony monit o podanie hasła.
- **Whatif:** Pokazuje, co się stanie, gdy polecenie zostanie uruchomione bez wykonywania polecenia.
- **Confirm:** potwierdza przed wykonaniem jakiejkolwiek czynności. Kiedy użyjesz tego parametru otrzymasz poniższe opcje. [Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
- **CommonParameters:** Poniżej znajdują się wspólne parametry używane z poleceniem cmdlet Move-Item, które są również nazywane parametrami funkcji zaawansowanych. Verbose, -Debug, -ErrorAction, -WarningAction, - InformationAction, -ErrorVariable, - WarningVariable, -InformationVariable, -OutVariable, -OutBuffer, -Pipelinevariable,-UseTransacation

Przykłady implementacji PowerShell Move-Item

Wykonaj poniższe polecenia według przykładów:

0. Przygotowanie

```
New-Item -Path 'C:\temp\New Folder' -ItemType Directory
```

```
New-Item -Path 'C:\temp\New Folder\file.txt' -ItemType File -f -v "nnn"
```

```
New-Item -Path 'C:\temp\cars.xml' -ItemType File -f -v "xml_nnn"
```

```
New-Item -Path 'C:\temp\kealtcheck.html' -ItemType File -f -v "html_nnn"
```

```
New-Item -Path 'C:\temp\servicereport.html' -ItemType File -f -v "service_html_nnn"
```

```
New-Item -Path 'C:\temp\style.css' -ItemType File -f -v "css_nnn"
```

```
New-Item -Path 'C:\temp\test.xlsx' -ItemType File -f -v "xlsx_nnn"
```

```
New-Item -Path 'C:\temp\testcsv.csv' -ItemType File -f -v "csv_nnn"
```

```
New-Item -Path 'C:\Temp0' -ItemType Directory
```

```
New-Item -Path 'C:\Temp1' -ItemType Directory
```

```
New-Item -Path 'C:\Temp2' -ItemType Directory
```

1. Przenieś element ze źródłem i miejscem docelowym

```
Move-Item -Path C:\temp\* -Destination C:\Temp0
```

```
Get-ChildItem -Path C:\Temp0
```

Directory: C:\Temp0				
Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
d-----	30.07.2024	10:44		New Folder
-a-----	30.07.2024	10:44	7	cars.xml
-a-----	30.07.2024	10:44	8	kealtcheck.html
-a-----	30.07.2024	10:44	16	servicereport.html
-a-----	30.07.2024	10:44	7	style.css
-a-----	30.07.2024	10:44	8	test.xlsx
-a-----	30.07.2024	10:44	7	testcsv.csv

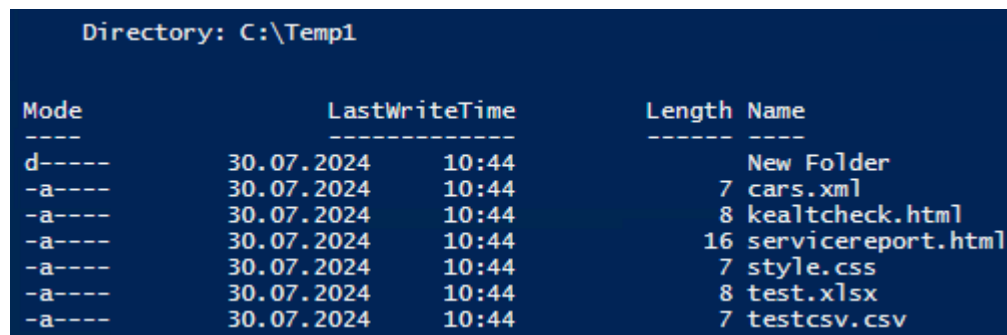
Jeśli użyjesz symbolu wieloznacznego (*), który jest wyświetlany w powyższym przykładzie, przeniesie on zawartość folderu, w przeciwnym razie przeniesie cały folder. W powyższym przykładzie nie możesz zobaczyć, które pliki zostały przeniesione ze źródła do miejsca docelowego, ponieważ nie określono jawnych parametrów, ale gdy używasz PassThru (który zwraca dane wyjściowe w samej konsoli) lub parametru - Verbose, dane wyjściowe/procesy będą wyświetlane tylko w konsoli PowerShell .

2. Przenieś element z - PassThru

Kod:

```
Move-Item -Path C:\Temp0\* -Destination C:\Temp1 -PassThru
```

Efekt:



Directory: C:\Temp1

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
d-----	30.07.2024	10:44		New Folder
-a-----	30.07.2024	10:44	7	cars.xml
-a-----	30.07.2024	10:44	8	kealtcheck.html
-a-----	30.07.2024	10:44	16	servicereport.html
-a-----	30.07.2024	10:44	7	style.css
-a-----	30.07.2024	10:44	8	test.xlsx
-a-----	30.07.2024	10:44	7	testcsv.csv

3. Przenieś element z - Verbose

Kod:

```
Move-Item -Path C:\temp1\* -Destination C:\Temp2 -Verbose
```

Efekt:

```

VERBOSE: Performing the operation "Move Directory" on target "Item: C:\temp1\New Folder Destination: C:\Temp2\New Folder".
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp1\cars.xml Destination: C:\Temp2\cars.xml".
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp1\kealtcheck.html Destination: C:\Temp2\kealtcheck.html".
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp1\servicereport.html Destination: C:\Temp2\servicereport.html".
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp1\style.css Destination: C:\Temp2\style.css".
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp1\test.xlsx Destination: C:\Temp2\test.xlsx".
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp1\testcsv.csv Destination: C:\Temp2\testcsv.csv".

```

4. Przenieś element z - Include

Kod:

```
Move-Item -Path C:\temp2\* -Destination C:\Temp1 -Include *.html -Verbose
```

Wyjaśnienie do powyższego kodu: W powyższym przykładzie skrypt przeniesie wszystkie pliki, które mają rozszerzenie .html. Gdy użyjemy tylko parametru Include, skrypt zwróci błąd dla plików lub folderów, które nie mają rozszerzenia HTML.

Efekt:

```

Move-Item : Cannot move item because the item at 'C:\temp2\New Folder' does not exist.
At line:1 char:1
+ Move-Item -Path C:\temp2\* -Destination C:\Temp1 -Include *.html -Ver ...
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (:) [Move-Item], PSInvalidOperationException
+ FullyQualifiedErrorId : InvalidOperation,Microsoft.PowerShell.Commands.MoveItemCommand

Move-Item : Cannot move item because the item at 'C:\temp2\cars.xml' does not exist.
At line:1 char:1
+ Move-Item -Path C:\temp2\* -Destination C:\Temp1 -Include *.html -Ver ...
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (:) [Move-Item], PSInvalidOperationException
+ FullyQualifiedErrorId : InvalidOperation,Microsoft.PowerShell.Commands.MoveItemCommand

VERBOSE: Performing the operation "Move File" on target "Item: C:\temp2\kealtcheck.html Destination: C:\Temp1\kealtcheck.html".
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp2\servicereport.html Destination: C:\Temp1\servicereport.html".
Move-Item : Cannot move item because the item at 'C:\temp2\style.css' does not exist.
At line:1 char:1

```

Powyższy wyjątek możemy po prostu wyeliminować za pomocą parametru -ErrorAction.

Kod:

```
Move-Item -Path C:\temp2\* -Destination C:\Temp1 -Include *.html -ErrorAction Ignore -Verbose
```

Efekt:

```
PS C:\Windows\system32>
```

5. Przenieś-przedmiot z - Exclude

Kod:

```
Move-Item -Path C:\temp2\* -Destination C:\Temp1 -Exclude *.html -ErrorAction Ignore -Verbose
```

Efekt:

```
VERBOSE: Performing the operation "Move Directory" on target "Item: C:\temp2\New Folder Destination: C:\Temp1\New Folder".  
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp2\cars.xml Destination: C:\Temp1\cars.xml".  
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp2\style.css Destination: C:\Temp1\style.css".  
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp2\test.xlsx Destination: C:\Temp1\test.xlsx".  
VERBOSE: Performing the operation "Move File" on target "Item: C:\temp2\testcsv.csv Destination: C:\Temp1\testcsv.csv".
```

W powyższym przykładzie pliki zostaną przeniesione z wyjątkiem plików .html do folderu docelowego.

6. Przenieś element z elementami podrzędnymi

Kiedy używamy znaku wieloznacznego, obejmuje on tylko elementy w tym folderze, ale nie podfoldery. Na przykład folder C:\temp* nie może filtrować plików w folderze C:\temp\LGPO lub w innych podfolderach. Aby to zrobić, musimy najpierw rozpakować pliki za pomocą parametru Get-ChildItem i -Recurse, a następnie przenieść je w inne miejsce.

Kod:

```
Get-ChildItem -Path C:\Temp1 -Recurse -Include *.html | Move-Item -Destination C:\Temp1 -Verbose
```

```
Get-ChildItem -Path C:\Temp1
```

Efekt:

Directory: C:\Temp1

Mode	LastWriteTime		Length	Name
----	-----		-----	----
d-----	30.07.2024	11:02		New Folder
-a----	30.07.2024	11:02	7	cars.xml
-a----	30.07.2024	11:02	8	kealtcheck.html
-a----	30.07.2024	11:02	16	servicereport.html
-a----	30.07.2024	11:02	7	style.css
-a----	30.07.2024	11:02	8	test.xlsx
-a----	30.07.2024	11:02	7	testcsv.csv

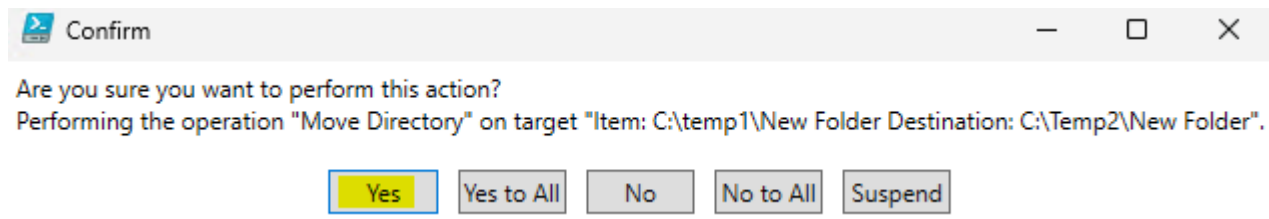
7. Przenieś-element z - Confirm

Kod:

```
Move-Item -Path C:\temp1\* -Destination C:\Temp2 -Confirm -Verbose
```

Poczekaj na potwierdzenie od użytkownika przed przeniesieniem jakiegokolwiek pozycji.

Efekt:



8. Przenieś element z -WhatIf

Kod:

```
Move-Item -Path C:\temp2\* -Destination C:\Temp1 -WhatIf
```

Pokazuje, co się stanie, gdy polecenie zostanie wykonane bez faktycznego uruchomienia go.

Efekt:

```
What if: Performing the operation "Move Directory" on target "Item: C:\temp2\New Folder Destination: C:\Temp1\New Folder".
What if: Performing the operation "Move File" on target "Item: C:\temp2\cars.xml Destination: C:\Temp1\cars.xml".
What if: Performing the operation "Move File" on target "Item: C:\temp2\kealtcheck.html Destination: C:\Temp1\kealtcheck.html".
What if: Performing the operation "Move File" on target "Item: C:\temp2\servicereport.html Destination: C:\Temp1\servicereport.html".
What if: Performing the operation "Move File" on target "Item: C:\temp2\style.css Destination: C:\Temp1\style.css".
What if: Performing the operation "Move File" on target "Item: C:\temp2\test.xlsx Destination: C:\Temp1\test.xlsx".
What if: Performing the operation "Move File" on target "Item: C:\temp2\testcsv.csv Destination: C:\Temp1\testcsv.csv".
```

9. Przenieś-element z poświadczeniami

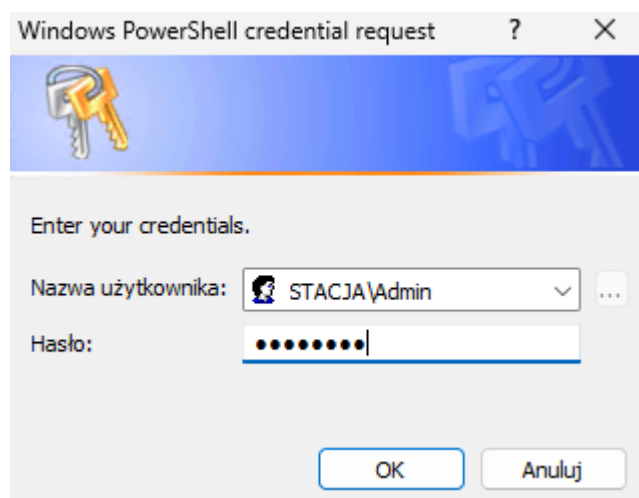
Kod:

```
$Source = "C:\temp2\*"
```

```
New-PSDrive -Name K -PSProvider FileSystem -Root "\\127.0.0.1\c$\Temp1" -Credential "$($env:COMPUTERNAME)\Admin" -Persist
```

```
Copy-Item -Path $Source -Destination "K:\"
```

Powyższe polecenie otworzy pole danych logowania z nazwą użytkownika ad\xyz (domena\nazwa użytkownika).



Efekt:

Name	Used (GB)	Free (GB)	Provider	Root	CurrentLocation
----	-----	-----	-----	----	-----
K	30,83	992,29	FileSystem	\\127.0.0.1\c\$\Temp1	

Get-ChildItem -Path C:\Temp1

Directory: C:\Temp1					
Mode	LastWriteTime		Length	Name	
----	-----	-----	-----	----	----
d-----	30.07.2024	11:13		New Folder	
-a----	30.07.2024	11:02	7	cars.xml	
-a----	30.07.2024	11:02	8	kealtcheck.html	
-a----	30.07.2024	11:02	16	servicereport.html	
-a----	30.07.2024	11:02	7	style.css	
-a----	30.07.2024	11:02	8	test.xlsx	
-a----	30.07.2024	11:02	7	testcsv.csv	

Zgłoszenie

Posprzątaj

Usuń zbędne foldery i pliki

Remove-Item -Path "C:\temp","C:\Temp2","C:\Temp1","C:\Temp0" -Recurse

Usuń dysk z systemem plików

Remove-PSDrive -Name K

Wyświetl dyski w bieżącej sesji

Get-PSDrive

Koniec