



Windows PowerShell Get-Help on Cmdlet 'Move-Item'

PS:\>Get-HELP Move-Item -Full

NAME

Move-Item

SYNOPSIS

Moves an item from one location to another.

SYNTAX

```
Move-Item [[-Destination] <System.String>] [-Credential <System.Management.Automation.PSCredential>] [-Exclude
<System.String[]>] [-Filter <System.String>] [-Force]
```

```
[-Include <System.String[]>] -LiteralPath <System.String[]> [-PassThru] [-UseTransaction] [-Confirm] [-WhatIf]
[<CommonParameters>]
```

```
Move-Item [-Path] <System.String[]> [[-Destination] <System.String>] [-Credential
<System.Management.Automation.PSCredential>] [-Exclude <System.String[]>] [-Filter
<System.String>] [-Force] [-Include <System.String[]>] [-PassThru] [-UseTransaction] [-Confirm] [-WhatIf]
[<CommonParameters>]
```

DESCRIPTION

The `Move-Item` cmdlet moves an item, including its properties, contents, and child items, from one location to another location. The locations must be supported by the same provider.`

For example, it can move a file or subdirectory from one directory to another or move a registry subkey from one key to another. When you move an item, it is added to the new location and deleted from its original location.

PARAMETERS

`-Credential <System.Management.Automation.PSCredential>`

> [!NOTE] > This parameter is not supported by any providers installed with PowerShell. > To impersonate another user, or elevate your credentials when running this cmdlet, > use `Invoke-Command (./Microsoft.PowerShell.Core/Invoke-Command.md)`.

Required?	false
Position?	named
Default value	Current user
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	true

`-Destination <System.String>`

Specifies the path to the location where the items are being moved. The default is the current directory. Wildcards are permitted, but the result must resolve to a single location.

To rename the item being moved, specify a new name in the value of the Destination parameter.

Required?	false
Position?	1
Default value	Current directory
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	true

-Exclude <System.String[]>

Specifies, as a string array, an item or items that this cmdlet excludes in the operation. The value of this parameter qualifies the Path parameter. Enter a path

element or pattern, such as ``.txt``. Wildcard characters are permitted. The Exclude `*` parameter is effective only when the command includes the contents of an

item, such as ``C:\Windows*``, where the wildcard character specifies the contents of the ``C:\Windows`` directory.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? true

-Filter <System.String>

Specifies a filter to qualify the Path parameter. The FileSystem
(../Microsoft.PowerShell.Core/About/about_FileSystem_Provider.md)provider is the only installed

PowerShell provider that supports the use of filters. You can find the syntax for the FileSystem filter language in
about_Wildcards

(../Microsoft.PowerShell.Core/About/about_Wildcards.md). Filters are more efficient than other parameters, because
the provider applies them when the cmdlet gets

the objects rather than having PowerShell filter the objects after they are retrieved.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? true

-Force <System.Management.Automation.SwitchParameter>

Forces the command to run without asking for user confirmation. Implementation varies from provider to provider. For
more information, see about_Providers

(../Microsoft.PowerShell.Core/About/about_Providers.md).

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

`-Include <System.String[]>`

Specifies, as a string array, an item or items that this cmdlet includes in the operation. The value of this parameter qualifies the Path parameter. Enter a path

element or pattern, such as ``.txt``. Wildcard characters are permitted. The Include `*` parameter is effective only when the command includes the contents of an

item, such as ``C:\Windows*``, where the wildcard character specifies the contents of the ``C:\Windows`` directory.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	true

`-LiteralPath <System.String[]>`

Specifies the path to the current location of the items. The value of LiteralPath is used exactly as it is typed. No characters are interpreted as wildcards. If

the path includes escape characters, enclose it in single quotation marks (```). Single quotation marks tell PowerShell not to interpret any characters as escape sequences.

For more information, see [about_Quoting_Rules](#) (`../Microsoft.Powershell.Core/About/about_Quoting_Rules.md`).

Required?	true
Position?	named
Default value	None
Accept pipeline input?	True (ByPropertyName)

Accept wildcard characters? false

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object representing the moved item. By default, this cmdlet does not generate any output.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Path <System.String[]>

Specifies the path to the current location of the items. The default is the current directory. Wildcard characters are permitted.

Required? true

Position? 0

Default value Current directory

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? true

-UseTransaction <System.Management.Automation.SwitchParameter>

Includes the command in the active transaction. This parameter is valid only when a transaction is in progress. For more information, see [about_Transactions](#)

(../Microsoft.PowerShell.Core/About/about_Transactions.md).

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see [about_CommonParameters \(https://go.microsoft.com/fwlink/?LinkID=113216\)](https://go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.String

You can pipe a string that contains a path to this cmdlet.

OUTPUTS

None

By default, this cmdlet returns no output.

When you use the PassThru parameter, this cmdlet returns an object representing the moved item.

NOTES

Windows PowerShell includes the following aliases for ``Move-Item``:

- ``mi``

- ``move``

- ``mv``

- This cmdlet will move files between drives that are supported by the same provider, but it will move directories only within the same drive. - Because a

- ``Move-Item`` command moves the properties, contents, and child items of an item, all moves are recursive by default.

- This cmdlet is designed to work with the

- data exposed by any provider. To list the providers available in your session, type ``Get-PSProvider``. For more information, see `about_Providers`

- (../Microsoft.PowerShell.Core/About/about_Providers.md).

-- Example 1: Move a file to another directory and rename it --

```
Move-Item -Path C:\test.txt -Destination E:\Temp\tst.txt
```

Example 2: Move a directory and its contents to another directory

```
Move-Item -Path C:\Temp -Destination C:\Logs
```

Example 3: Move all files of a specified extension from the current directory to another directory

```
Move-Item -Path *.*.txt -Destination C:\Logs
```

Example 4: Recursively move all files of a specified extension from the current directory to another directory

```
Get-ChildItem -Path ".\*.txt" -Recurse | Move-Item -Destination "C:\TextFiles"
```

The command uses the `Get-ChildItem` cmdlet to get all of the child items in the current directory (represented by the dot (`.`)) and its subdirectories that have a

`*.txt` file name extension. It uses the `Recurse` parameter to make the retrieval recursive and the `Include` parameter to limit the retrieval to `*.txt` files.

The pipeline operator (`|`) sends the results of this command to `Move-Item`, which moves the text files to the `TextFiles` directory.

If files that are to be moved to `C:\Textfiles` have the same name, `Move-Item` displays an error and continues, but it moves only one file with each name to

`C:\Textfiles`. The other files remain in their original directories.

If the `Textfiles` directory (or any other element of the destination path) does not exist, the command fails. The missing directory is not created for you, even if

you use the `Force` parameter. `Move-Item` moves the first item to a file called `Textfiles` and then displays an error explaining that the file already exists.

Also, by default, `Get-ChildItem` does not move hidden files. To move hidden files, use the `Force` parameter with `Get-ChildItem`.

> [!NOTE] > In Windows PowerShell 2.0, when using the `Recurse` parameter of the `Get-ChildItem` cmdlet, the `>` value of the `Path` parameter must be a container. Use the

Include parameter to specify the `> *.txt` file name extension filter > (`Get-ChildItem -Path .\ -Include *.txt -Recurse |`

Move-Item -Destination C:\TextFiles`).

--- Example 5: Move registry keys and values to another key ---

```
Move-Item "HKLM:\software\mycompany\*" "HKLM:\software\mynewcompany"
```

Example 6: Move a directory and its contents to a subdirectory of the specified directory

```
Move-Item -LiteralPath 'Logs[Sept`06]' -Destination 'Logs[2006]'
```

The LiteralPath parameter is used instead of Path , because the original directory name includes left bracket and right bracket characters (`[` and `] `). The path is

also enclosed in single quotation marks (` `), so that the backtick symbol (` ` ` `) is not misinterpreted.

The Destination parameter must also be enclosed in single quotation marks because it includes brackets that can be misinterpreted.

RELATED LINKS

Online

Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.management/move-item?view=powershell-5.1&WT.mc_id=ps-gethelp

Clear-Item

Copy-Item

Get-Item

Invoke-Item

New-Item

Remove-Item

Rename-Item

Set-Item

about_Providers