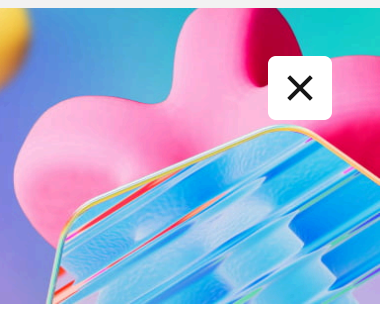

Remove-Item

Reference

Module: Microsoft.PowerShell.Management

In this article

Deletes the specified items.

Syntax

PowerShell Copy

```
Remove-Item
      [-Path] <String[]>
      [-Filter <String>]
      [-Include <String[]>]
      [-Exclude <String[]>]
      [-Recurse]
      [-Force]
      [-Credential <PSCredential>]
      [-WhatIf]
      [-Confirm]
      [-UseTransaction]
      [-Stream <String[]>]
      [<CommonParameters>]
```

PowerShell Copy

```
Remove-Item
      -LiteralPath <String[]>
      [-Filter <String>]
      [-Include <String[]>]
      [-Exclude <String[]>]
      [-Recurse]
      [-Force]
      [-Credential <PSCredential>]
      [-WhatIf]
      [-Confirm]
      [-UseTransaction]
```

```
      [-Stream <String[]>]
      [<CommonParameters>]
```

PowerShell Copy

```
Remove-Item
      [-Path] <String[]>
      [-Filter <String>]
      [-Include <String[]>]
      [-Exclude <String[]>]
      [-Recurse]
      [-Force]
      [-Credential <PSCredential>]
      [-WhatIf]
      [-Confirm]
      [-UseTransaction]
      [-DeleteKey] <CommonParameters>]
```

PowerShell Copy

```
Remove-Item
      -LiteralPath <String[]>
      [-Filter <String>]
      [-Include <String[]>]
      [-Exclude <String[]>]
      [-Recurse]
      [-Force]
      [-Credential <PSCredential>]
      [-WhatIf]
      [-Confirm]
      [-UseTransaction]
      [-DeleteKey]
      [<CommonParameters>]
```

PowerShell Copy

```
Remove-Item
      [-Path] <string[]>
      [-Filter <string>]
      [-Include <string[]>]
      [-Exclude <string[]>]
      [-Recurse]
      [-Force]
      [-Credential <pscredential>]
      [-WhatIf]
      [-Confirm]
      [-UseTransaction]
      [<CommonParameters>]
```

PowerShell Copy

```
Remove-Item
      -LiteralPath <string[]>
      [-Filter <string>]
      [-Include <string[]>]
      [-Exclude <string[]>]
      [-Recurse]
      [-Force]
      [-Credential <pscredential>]
      [-WhatIf]
      [-Confirm]
      [-UseTransaction]
      [<CommonParameters>]
```

Description

The `Remove-Item` cmdlet deletes one or more items. Because it's supported by many providers, it can delete many different types of items, including files, folders, registry keys, variables, aliases, and functions.

Examples

Example 1: Delete files that have any file extension

This example deletes all files with names that include a dot (`.`) from the `C:\Test` folder. Because the command specifies a dot, the command doesn't delete folders or files that have no file extension.

PowerShell Copy

```
Remove-Item C:\Test\*.*
```

Example 2: Delete document files in a folder

This example deletes from the current folder all files that have a `.doc` file extension and a name that doesn't include `*1*`.

PowerShell Copy

```
Remove-Item * -Include *.doc -Exclude *1*
```

It uses the wildcard character (`*`) to specify the contents of the current folder. It uses the **Include** and **Exclude** parameters to specify the files to delete.

Example 3: Delete hidden, read-only files

This command deletes a file that's both *hidden* and *read-only*.

PowerShell Copy

```
Remove-Item -Path C:\Test\hidden-RO-file.txt -Force
```

It uses the **Path** parameter to specify the file. It uses the **Force** parameter to delete it. Without **Force**, you can't delete *read-only* or *hidden* files.

Example 4: Delete files in subfolders recursively

This command deletes all the CSV files in the current folder and all subfolders recursively.

Because the **Recurse** parameter in `Remove-Item` has a known issue, the command in this example uses `Get-ChildItem` to get the desired files, and then uses the pipeline operator to pass them to `Remove-Item`.

PowerShell Copy

```
Get-ChildItem * -Include *.csv -Recurse | Remove-Item
```

In the `Get-ChildItem` command, **Path** has a value of (`*`), which represents the contents of the current folder. It uses **Include** to specify the CSV file type, and it uses **Recurse** to make the retrieval recursive. If you try to specify the file type in the path, such as `-Path *.csv`, the cmdlet interprets the subject of the search to be a file that has no child items, and **Recurse** fails.

> **Note**
>
> This behavior was fixed in Windows versions 1909 and up.

Example 5: Delete subkeys recursively

This command deletes the "OldApp" registry key and all its subkeys and values. It uses `Remove-Item` to remove the key. The path is specified, but the optional parameter name (**Path**) is omitted.

The **Recurse** parameter deletes all of the contents of the "OldApp" key recursively. If the key contains subkeys and you omit the **Recurse** parameter, you are prompted to confirm that you want to delete the contents of the key.

PowerShell Copy

```
Remove-Item HKLM:\Software\MyCompany\OldApp -Recurse
```

Example 6: Deleting files with special characters

The following example shows how to delete files that contain special characters like brackets or parentheses.

PowerShell Copy

```
Get-ChildItem

Directory: C:\temp\Downloads

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          6/1/2018  12:19 PM           1362 myFile.txt
-a---          6/1/2018  12:30 PM           1132 myFile[1].txt
-a---          6/1/2018  12:19 PM           1283 myFile[2].txt
-a---          6/1/2018  12:19 PM           1432 myFile[3].txt

Get-ChildItem | Where-Object Name -Like '*`[*'

Directory: C:\temp\Downloads

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          6/1/2018  12:30 PM           1132 myFile[1].txt
-a---          6/1/2018  12:19 PM           1283 myFile[2].txt
-a---          6/1/2018  12:19 PM           1432 myFile[3].txt

Get-ChildItem | Where-Object Name -Like '*`[*' | ForEach-Object { Remove-Item -L
Get-ChildItem

Directory: C:\temp\Downloads

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          6/1/2018  12:19 PM           1362 myFile.txt
```

Example 7: Remove an alternate data stream

This example shows how to use the **Stream** dynamic parameter of the `Remove-Item` cmdlet to delete an alternate data stream. The stream parameter is introduced in Windows PowerShell 3.0.

PowerShell Copy

```
Get-Item C:\Test\Copy-Script.ps1 -Stream Zone.Identifier

FileName: \\C:\Test\Copy-Script.ps1

Stream                   Length
------                   ------
Zone.Identifier              26
```

```
Remove-Item C:\Test\Copy-Script.ps1 -Stream Zone.Identifier
Get-Item C:\Test\Copy-Script.ps1 -Stream Zone.Identifier

Get-Item : Could not open alternate data stream 'Zone.Identifier' of file 'C:\Te
At line:1 char:1
+ Get-Item 'C:\Test\Copy-Script.ps1' -Stream Zone.Identifier
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\Test\Copy-Script.ps1:String) [
    + FullyQualifiedErrorId : AlternateDataStreamNotFound,Microsoft.PowerShell.C
```

The **Stream** parameter `Get-Item` gets the `Zone.Identifier` stream of the `Copy-Script.ps1` file. `Remove-Item` uses the **Stream** parameter to remove the `Zone.Identifier` stream of the file. Finally, the `Get-Item` cmdlet shows that the `Zone.Identifier` stream was deleted.

Parameters

-Confirm

Prompts you for confirmation before running the cmdlet. For more information, see the following articles:

- about_Preference_Variables
- about_Functions_CmdletBindingAttribute

Expand table

Type:	SwitchParameter
Aliases:	cf
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Credential

> **Note**
>
> This parameter isn't supported by any providers installed with PowerShell. To impersonate another user, or elevate your credentials when running this cmdlet, use Invoke-Command.

Expand table

Type:	PSCredential
Position:	Named
Default value:	Current user
Required:	False
Accept pipeline input:	True
Accept wildcard characters:	False

-DeleteKey

This is a dynamic parameter made available by the **Certificate** provider. The **Certificate** provider and this parameter are only available on Windows platforms.

When provided, the cmdlet deletes the private key when the certificate is deleted.

For more information, see about_Certificate_Provider.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Exclude

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.

When using **Recurse** with **Exclude**, **Exclude** only filters results of the current directory. If there are files that match the **Exclude** pattern in subfolders, those files are removed along with its parent directory.

Expand table

Type:	String[]
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	True

-Filter

Specifies a filter to qualify the **Path** parameter. The FileSystem 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. 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're retrieved.

Expand table

Type:	String
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False

Accept wildcard characters:	True

-Force

Forces the cmdlet to remove items that can't otherwise be changed, such as hidden or read-only files or read-only aliases or variables. The cmdlet can't remove constant aliases or variables. Implementation varies from provider to provider. For more information, see about_Providers. Even using the **Force** parameter, the cmdlet can't override security restrictions.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Include

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.

Expand table

Type:	String[]
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	True

-LiteralPath

Specifies a path to one or more locations. The value of **LiteralPath** is used exactly as it's 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.

Expand table

Type:	String[]
Aliases:	PSPath
Position:	Named
Default value:	None
Required:	True

Accept pipeline input:	True
Accept wildcard characters:	False

-Path

Specifies a path of the items being removed. Wildcard characters are permitted.

Expand table

Type:	String[]
Position:	0
Default value:	None
Required:	True
Accept pipeline input:	True
Accept wildcard characters:	True

-Recurse

Indicates that this cmdlet deletes the items in the specified locations and in all child items of the locations.

The **Recurse** parameter might not delete all subfolders or all child items. This is a known issue.

> **Note**
>
> This behavior was fixed in Windows versions 1909 and newer.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Stream

This is a dynamic parameter made available by the **FileSystem** provider. This parameter is only available on Windows. This parameter can't be used in combination with the **Recurse** parameter.

You can use `Remove-Item` to delete an alternative data stream, such as `Zone.Identifier`. However, it isn't the recommended way to eliminate security checks that block files that are downloaded from the Internet. If you verify that a downloaded file is safe, use the `Unblock-File` cmdlet.

This parameter was introduced in Windows PowerShell 3.0.

For more information, see about_FileSystem_Provider.

Expand table

Type:	String[]
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	True

-UseTransaction

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

Expand table

Type:	SwitchParameter
Aliases:	usetx
Position:	Named
Default value:	False
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-WhatIf

Shows what would happen if the cmdlet runs. The cmdlet isn't run.

Expand table

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

Inputs

String

You can pipe a string that contains a path, but not a literal path, to this cmdlet.

Outputs

None

This cmdlet returns no output.

Notes

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

- `del`
- `erase`
- `rd`
- `ri`
- `rm`
- `rmdir`

The `Remove-Item` 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.

When you try to delete a folder that contains items without using the **Recurse** parameter, the cmdlet prompts for confirmation. Using `-Confirm:$false` doesn't suppress the prompt. This is by design.

Related Links

- Clear-Item
- Copy-Item
- Get-Item
- Invoke-Item
- Move-Item
- New-Item
- Remove-ItemProperty
- Rename-Item
- Set-Item
- about_Providers
- about_Preference_Variables
- about_Functions_CmdletBindingAttribute

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see our contributor guide.

PowerShell feedback

PowerShell is an open source project. Select a link to provide feedback:

Open a documentation issue

Provide product feedback