



python



PowerShell

FPDF Library
PDF generator

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Rename-Item'

PS C:\Users\wahid> Get-Help Rename-Item

NAME

Rename-Item

SYNOPSIS

Renames an item in a PowerShell provider namespace.

SYNTAX

```
Rename-Item [-NewName] <System.String> [-Credential  
<System.Management.Automation.PSCredential>] [-Force] -LiteralPath  
<System.String> [-PassThru] [-UseTransaction] [-Confirm] [-WhatIf]  
[<CommonParameters>]
```

```
Rename-Item [-Path] <System.String> [-NewName] <System.String> [-Credential  
<System.Management.Automation.PSCredential>] [-Force] [-PassThru]  
[-UseTransaction] [-Confirm] [-WhatIf] [<CommonParameters>]
```

DESCRIPTION

The `Rename-Item` cmdlet changes the name of a specified item. This cmdlet does not affect the content of the item being renamed.

You can't use `Rename-Item`` to move an item, such as by specifying a path together with the new name. To move and rename an item, use the `Move-Item`` cmdlet.

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`).

`-Force <System.Management.Automation.SwitchParameter>`

Forces the cmdlet to rename items that can't otherwise be changed, such as hidden or read-only files or read-only aliases or variables. The cmdlet can't change constant aliases or variables. Implementation varies from provider to provider. For more information, see `about_Providers` (`../Microsoft.PowerShell.Core/About/about_Providers.md`).

Even using the Force parameter, the cmdlet can't override security restrictions.

`-LiteralPath <System.String>`

Specifies a path to one or more locations. 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`).

-NewName <System.String>

Specifies the new name of the item. Enter only a name, not a path and name. If you enter a path that differs from the path that is specified in the Path parameter, `Rename-Item` generates an error. To rename and move an item, use `Move-Item`.

You can't use wildcard characters in the value of the NewName parameter.

To specify a name for multiple files, use the Replace operator in a regular expression. For more information about the Replace operator, see [about_Comparison_Operators](#)

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

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object that represents the item to the pipeline. By default, this cmdlet does not generate any output.

-Path <System.String>

Specifies the path of the item to rename.

-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).

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

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

<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>).

----- Example 1: Rename a file -----

```
Rename-Item -Path "c:\logfiles\daily_file.txt" -NewName "monday_file.txt"
```

----- Example 2: Rename and move an item -----

```
Rename-Item -Path "project.txt" -NewName "d:\archive\old-project.txt"
```

Rename-Item : can't rename because the target specified represents a path or device name.

At line:1 char:12

```
+ Rename-Item <<<< -path project.txt -NewName d:\archive\old-project.txt
```

```
+ CategoryInfo      : InvalidArgument: (:) [Rename-Item], PS> Move-Item
```

```
-Path "project.txt" -De
```

```
stination "d:\archive\old-project.txt"
```

This example attempts to rename the `project.txt` file in the current directory to `old-project.txt` in the `D:\Archive` directory. The result is the error shown in the output.

Use the `Move-Item` cmdlet, instead.

----- Example 3: Rename a registry key -----

```
Rename-Item -Path "HKLM:\Software\MyCompany\Advertising" -NewName "Marketing"
```

----- Example 4: Rename multiple files -----

```
Get-ChildItem *.txt
```

```
Directory: C:\temp\files
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	10/3/2019 7:47 AM	2918	Friday.TXT
-a----	10/3/2019 7:46 AM	2918	Monday.Txt
-a----	10/3/2019 7:47 AM	2918	Wednesday.txt

```
Get-ChildItem *.txt | Rename-Item -NewName { $_.Name -replace '.txt','.log' }
```

```
Get-ChildItem *.log
```

```
Directory: C:\temp\files
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	10/3/2019 7:47 AM	2918	Friday.log
-a----	10/3/2019 7:46 AM	2918	Monday.log
-a----	10/3/2019 7:47 AM	2918	Wednesday.log

The `Get-ChildItem` cmdlet gets all the files in the current folder that have a `.txt` file extension then pipes them to `Rename-Item`. The value of `NewName` is a script block that runs before the value is submitted to the `NewName` parameter.

In the script block, the `$_` automatic variable represents each file object as it comes to the command through the pipeline. The script block uses the `-replace` operator to replace the file extension of each file with `.log`.

Notice that matching using the `-replace` operator is not case sensitive.

REMARKS

To see the examples, type: "get-help Rename-Item -examples".

For more information, type: "get-help Rename-Item -detailed".

For technical information, type: "get-help Rename-Item -full".

For online help, type: "get-help Rename-Item -online"