
**Full credit is given to all the above companies including the Operating System that this PDF file was generated!**

## *Windows PowerShell Get-Help on Cmdlet 'Split-Path'*

***PS:\>Get-HELP Split-Path -Full***

NAME

Split-Path

SYNOPSIS

Returns the specified part of a path.

SYNTAX

Split-Path [-Path] <System.String[]> [-Credential <System.Management.Automation.PSCredential>] [-IsAbsolute] [-Resolve] [-UseTransaction] [<CommonParameters>]

Split-Path [-Path] <System.String[]> [-Credential <System.Management.Automation.PSCredential>] [-Leaf] [-Resolve] [-UseTransaction] [<CommonParameters>]

Split-Path [-Credential <System.Management.Automation.PSCredential>] -LiteralPath <System.String[]> [-Resolve] [-UseTransaction] [<CommonParameters>]

Split-Path [-Path] <System.String[]> [-Credential <System.Management.Automation.PSCredential>] [-NoQualifier] [-Resolve] [-UseTransaction] [<CommonParameters>]

Split-Path [-Path] <System.String[]> [-Credential <System.Management.Automation.PSCredential>] [-Parent] [-Resolve] [-UseTransaction] [<CommonParameters>]

Split-Path [-Path] <System.String[]> [[-Qualifier]] [-Credential <System.Management.Automation.PSCredential>] [-Resolve] [-UseTransaction] [<CommonParameters>]

DESCRIPTION

The `Split-Path` cmdlet returns only the specified part of a path, such as the parent folder, a subfolder, or a filename. It can also get items that are referenced by

the split path and tell whether the path is relative or absolute.

You can use this cmdlet to get or submit only a selected part of a path.

PARAMETERS

-Credential <System.Management.Automation.PSCredential>

> [!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 (../Microsoft.PowerShell.Core/Invoke-Command.md).

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-IsAbsolute <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns `$True` if the path is absolute and `$False` if it's relative. An absolute path has a length greater than zero and doesn't use

a dot (`.`) to indicate the current path.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Leaf <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns only the last item or container in the path. For example, in the path `C:\Test\Logs\Pass1.log`, it returns only `Pass1.log`.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-LiteralPath <System.String[]>

Specifies the paths to be split. Unlike Path , the value of LiteralPath is used exactly as it is typed. No characters are interpreted as wildcard characters. 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.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-NoQualifier <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns the path without the qualifier. For the FileSystem or registry providers, the qualifier is the drive of the provider path, such

as `C:` or `HKCU:`. For example, in the path `C:\Test\Logs\Pass1.log`, it returns only `\Test\Logs\Pass1.log`.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Parent <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns only the parent containers of the item or of the container specified by the path. For example, in the path

`C:\Test\Logs\Pass1.log`, it returns `C:\Test\Logs`. The Parent parameter is the default split location parameter.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Path <System.String[]>

Specifies the paths to be split. Wildcard characters are permitted. If the path includes spaces, enclose it in quotation marks. You can also pipe a path to this

cmdlet.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? true

-Qualifier <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns only the qualifier of the specified path. For the FileSystem or registry providers, the qualifier is the drive of the provider

path, such as `C:` or `HKCU:`.

Required? false

Position? 1

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Resolve <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet displays the items that are referenced by the resulting split path instead of displaying the path elements.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-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

<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).

INPUTS

System.String

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

OUTPUTS

System.String

This cmdlet returns text strings. When you specify the Resolve parameter, it returns a string that describes the location of the items. It doesn't return objects

that represent the items, such as a FileInfo or RegistryKey object.

System.Boolean

When you specify the IsAbsolute parameter, this cmdlet returns a Boolean value.

NOTES

- The split location parameters ( Qualifier , Parent , Leaf , and NoQualifier ) are exclusive. You can use only one in each command.

- The cmdlets that contain the Path noun (the Path cmdlets) work with path names and return the names in a concise format that all PowerShell providers can

interpret. They're designed for use in programs and scripts where you want to display all or part of a path name in a particular format. Use them in the way

that you would use Dirname , Normpath , Realpath , Join , or other path manipulators.

- You can use the Path cmdlets together with several providers. These include the FileSystem, Registry, and Certificate providers.

- `Split-Path` 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: Get the qualifier of a path ----------

Split-Path -Path "HKCU:\Software\Microsoft" -Qualifier

HKCU:

This command returns only the qualifier of the path. The qualifier is the drive.

---------- Example 2: Display filenames ----------

Split-Path -Path "C:\Test\Logs*.log" -Leaf -Resolve

Pass1.log

Pass2.log

...

This command displays the files that are referenced by the split path. Because this path is split to the last item, also known as the leaf, the command displays only

the filenames.

The Resolve parameter tells `Split-Path` to display the items that the split path references, instead of displaying the split path.

Like all `Split-Path` commands, this command returns strings. It doesn't return FileInfo objects that represent the files.

---------- Example 3: Get the parent container ----------

Split-Path -Parent "C:\WINDOWS\system32\WindowsPowerShell\V1.0\about_*.txt"

C:\WINDOWS\system32\WindowsPowerShell\V1.0

This command returns only the parent containers of the path. Because it doesn't include any parameters to specify the split, `Split-Path` uses the split location

default, which is Parent .

------- Example 4: Determines whether a path is absolute -------

Split-Path -Path ".\My Pictures*.jpg" -IsAbsolute

False

This command determines whether the path is relative or absolute. In this case, because the path is relative to the current folder, which is represented by a dot

(`.`), it returns `$False`.

-------- Example 5: Change location to a specified path --------

PS C:\> Set-Location (Split-Path -Path $profile)

PS C:\Documents and Settings\User01\My Documents\WindowsPowerShell>

This command changes your location to the folder that contains the PowerShell profile.

The command in parentheses uses `Split-Path` to return only the parent of the path stored in the built-in `$Profile` variable. The Parent parameter is the default

split location parameter. Therefore, you can omit it from the command. The parentheses direct PowerShell to run the command first. This is a useful way to move to a

folder that has a long path name.

---------- Example 6: Split a path using the pipeline ----------

'C:\Documents and Settings\User01\My Documents\My Pictures' | Split-Path

C:\Documents and Settings\User01\My Documents

This command uses a pipeline operator (`|`) to send a path to `Split-Path`. The path is enclosed in quotation marks to indicate that it's a single token.

RELATED LINKS

Online Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.management/split-path?view=powershell-5.1&WT.mc_i

d=ps-gethelp

Convert-Path

Join-Path

Resolve-Path

Test-Path