



Windows PowerShell Get-Help on Cmdlet 'Push-Location'

PS:\>Get-HELP Push-Location -Full

NAME

Push-Location

SYNOPSIS

Adds the current location to the top of a location stack.

SYNTAX

```
Push-Location [-LiteralPath <System.String>] [-PassThru] [-StackName <System.String>] [-UseTransaction]
[<CommonParameters>]
```

```
Push-Location [[-Path] <System.String>] [-PassThru] [-StackName <System.String>] [-UseTransaction]
[<CommonParameters>]
```

DESCRIPTION

The `Push-Location` cmdlet adds ("pushes") the current location onto a location stack. If you specify a path, `Push-Location` pushes the current location onto a

location stack and then changes the current location to the location specified by the path. You can use the `Pop-Location` cmdlet to get locations from the location

stack.

By default, the ``Push-Location`` cmdlet pushes the current location onto the current location stack, but you can use the `StackName` parameter to specify an alternate

location stack. If the stack does not exist, ``Push-Location`` creates it.

For more information about location stacks, see the Notes (#notes).

PARAMETERS

`-LiteralPath <System.String>`

Specifies the path to the new location. Unlike the `Path` parameter, the value of the `LiteralPath` parameter 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.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

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

Passes an object representing the location to the pipeline. 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>`

Changes your location to the location specified by this path after it adds (pushes) the current location onto the top of the stack. Enter a path to any location

whose provider supports this cmdlet. Wildcards are permitted. The parameter name is optional.

Required?	false
Position?	1
Default value	None
Accept pipeline input?	True (ByPropertyName, ByValue)
Accept wildcard characters?	true

-StackName <System.String>

Specifies the location stack to which the current location is added. Enter a location stack name. If the stack does not exist, ``Push-Location`` creates it.

Without this parameter, ``Push-Location`` adds the location to the current location stack. By default, the current location stack is the unnamed default location

stack that PowerShell creates. To make a location stack the current location stack, use the `StackName` parameter of the ``Set-Location`` cmdlet. For more information about location stacks, see the Notes (#notes).

> [!NOTE] > ``Push-Location`` cannot add a location to the unnamed default stack unless it is the current > location stack.

Required?	false
Position?	named
Default value	Default stack
Accept pipeline input?	True (ByPropertyName)
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](#)

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 (but not a literal path) to this cmdlet.

OUTPUTS

None

By default, this cmdlet returns no output.

System.Management.Automation.PathInfo

When you use the PassThru parameter, this cmdlet returns a PathInfo object representing the current path after the cmdlet's operation.

NOTES

Windows PowerShell includes the following aliases for `Push-Location`:

- `pushd`

PowerShell supports multiple runspaces per process. Each runspace has its own current directory . This is not the same as

`[System.Environment]::CurrentDirectory``. This behavior can be an issue when calling .NET APIs or running native applications without providing explicit directory paths.

Even if the location cmdlets did set the process-wide current directory, you can't depend on it because another runspace might change it at any time. You should

use the location cmdlets to perform path-based operations using the current working directory specific to the current runspace.

A stack is a last-in, first-out list in which only the most recently added item is accessible. You add items to a stack in the order that you use them, and then

retrieve them for use in the reverse order. PowerShell lets you store provider locations in location stacks.

PowerShell creates an unnamed default location stack and you can create multiple named location stacks. If you do not specify a stack name, PowerShell uses the

current location stack. By default, the unnamed default location is the current location stack, but you can use the ``Set-Location`` cmdlet to change the current location stack.

To manage location stacks, use the PowerShell Location cmdlets, as follows.

- To add a location to a location stack, use the ``Push-Location`` cmdlet.
- To get a location from a location stack, use the ``Pop-Location`` cmdlet.
- To display the locations in the current location stack, use the Stack parameter of the ``Get-Location`` cmdlet.
- To display the locations in a named location stack, use the StackName parameter of the ``Get-Location`` cmdlet.
- To create a new location stack, use the StackName parameter of the ``Push-Location`` cmdlet. If you specify a stack

that does not exist, ``Push-Location`` creates the stack.

- To make a location stack the current location stack, use the `StackName` parameter of the ``Set-Location`` cmdlet.

The unnamed default location stack is fully accessible only when it is the current location stack. If you make a named location stack the current location stack,

you can no longer use the ``Push-Location`` or ``Pop-Location`` cmdlets to add or get items from the default stack or use the ``Get-Location`` cmdlet to display the

locations in the unnamed stack. To make the unnamed stack the current stack, use the `StackName` parameter of the ``Set-Location`` cmdlet with a value of ``$null`` or an empty string (`""`).

You can also refer to ``Push-Location`` by its built-in alias, ``pushd``. For more information, see `about_Aliases` (`../Microsoft.PowerShell.Core/About/about_Aliases.md`).

The ``Push-Location`` 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 -----

```
PS C:\> Push-Location C:\Windows
```

----- Example 2 -----

```
PS C:\> Push-Location HKLM:\Software\Policies -StackName RegFunction
```

You can use the Location cmdlets in any PowerShell drive (PSDrive).

----- Example 3 -----

```
PS C:\> Push-Location
```

----- Example 4 - Create and use a named stack -----

```
PS C:\> Push-Location ~ -StackName Stack2
```

```
PS C:\Users\User01> Pop-Location -StackName Stack2
```

```
PS C:\>
```

The first command pushes the current location onto a new stack named Stack2, and then changes the current location to the home directory, represented in the command

by the tilde symbol (~), which when used on a FileSystem provider drives is equivalent to '\$HOME'.

If Stack2 does not already exist in the session, 'Push-Location' creates it. The second command uses the 'Pop-Location' cmdlet to pop the original location ('C:')

from the Stack2 stack. Without the StackName parameter, 'Pop-Location' would pop the location from the unnamed default stack.

For more information about location stacks, see the Notes (#notes).

RELATED LINKS

Online

Version:

[https://learn.microsoft.com/powershell/module/microsoft.powershell.management/push-location?view=powershell-5.1&WT.](https://learn.microsoft.com/powershell/module/microsoft.powershell.management/push-location?view=powershell-5.1&WT.mc_id=ps-gethelp)

[mc_id=ps-gethelp](#)

[Get-Location](#)

[Pop-Location](#)

[Set-Location](#)

[about_Aliases](#)

[about_Providers](#)