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

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

NAME

Get-Location

SYNOPSIS

Gets information about the current working location or a location stack.

SYNTAX

Get-Location [-PSDrive <System.String[]>] [-PSProvider <System.String[]>] [-UseTransaction] [<CommonParameters>]

Get-Location [-Stack] [-StackName <System.String[]>] [-UseTransaction] [<CommonParameters>]

DESCRIPTION

The `Get-Location` cmdlet gets an object that represents the current directory, much like the print working directory (pwd) command.

When you move between PowerShell drives, PowerShell retains your location in each drive. You can use this cmdlet to find your location in each drive.

You can use this cmdlet to get the current directory at run time and use it in functions and scripts, such as in a function that displays the current directory in the

PowerShell prompt.

You can also use this cmdlet to display the locations in a location stack. For more information, see the Notes and the descriptions of the Stack and StackName

parameters.

PARAMETERS

-PSDrive <System.String[]>

Gets the current location in the specified PowerShell drive.

For instance, if you are in the `Cert:` drive, you can use this parameter to find your current location in the `C:` drive.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PSProvider <System.String[]>

Gets the current location in the drive supported by the specified PowerShell provider. If the specified provider supports more than one drive, this cmdlet returns

the location on the most recently accessed drive.

For example, if you are in the `C:` drive, you can use this parameter to find your current location in the drives of the PowerShell Registry provider.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Stack <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet displays the locations added to the current location stack. You can add locations to stacks by using the `Push-Location` cmdlet.

To display the locations in a different location stack, use the StackName parameter. For information about location stacks, see the Notes (#notes).

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-StackName <System.String[]>

Specifies, as a string array, the named location stacks. Enter one or more location stack names.

To display the locations in the current location stack, use the Stack parameter. To make a location stack the current location stack, use the `Set-Location`

cmdlet.

This cmdlet cannot display the locations in the unnamed default stack unless it is the current stack.

Required? false

Position? named

Default value None

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

(../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

None

You can't pipe objects to this cmdlet.

OUTPUTS

System.Management.Automation.PathInfo

By default, this cmdlet returns a PathInfo object.

System.Management.Automation.PathInfoStack

When you use the Stack or StackName parameters, this cmdlet returns a PathInfoStack object.

NOTES

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

- `gl`

- `pwd`

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. The `Get-Location` cmdlet returns the current directory of the current PowerShell runspace.

This cmdlet is designed to work with the data exposed by any provider. To list the providers in your session, type `Get-PSProvider`. For more information, see

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

The ways that the PSProvider , PSDrive , Stack , and StackName parameters interact depends on the provider. Some combinations will result in errors, such as

specifying both a drive and a provider that does not expose that drive. If no parameters are specified, this cmdlet returns the PathInfo object for the provider

that contains the current working location.

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 this 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

(`""`).

-------- Example 1: Display your current drive location --------

PS C:\Windows> Get-Location

Path

C:\Windows

For instance, if you are in the `Windows` directory of the `C:` drive, it displays the path to that directory.

Example 2: Display your current location for different drives

PS C:\> Set-Location C:\Windows

PS C:\Windows> Set-Location HKLM:\Software\Microsoft

PS HKLM:\Software\Microsoft> Set-Location "HKCU:\Control Panel\Input Method"

```
PS HKCU:\Control Panel\Input Method> Get-Location -PSDrive C

Path
----
C:\Windows

PS HKCU:\Control Panel\Input Method> Get-Location -PSDrive HKLM

Path
----
HKLM:\Software\Microsoft

PS HKCU:\Control Panel\Input Method> Set-Location C:
PS C:\Windows> Get-Location -PSProvider Registry

Path
----
HKCU:\Control Panel\Input Method


---------- Example 3: Get locations using stacks ----------

PS C:\> Push-Location C:\Windows
PS C:\Windows>Push-Location System32
PS C:\Windows\System32>Push-Location WindowsPowerShell -StackName Stack2
C:\Windows\System32\WindowsPowerShell>Get-Location -Stack

Path
----
C:\Windows
C:\

C:\Windows\System32\WindowsPowerShell>Get-Location -StackName Stack2
```

Path

C:\Windows\System32

---------- Example 4: Customize the PowerShell prompt ----------

PS C:\>

function prompt { 'PowerShell: ' + (Get-Location) + '> '}

PowerShell: C:\>

The function that defines the prompt includes a `Get-Location` command, which is run whenever the prompt appears in the console.

The format of the default PowerShell prompt is defined by a special function named `prompt`. You can change the prompt in your console by creating a new function

named `prompt`.

To see the current prompt function, type the following command: `Get-Content Function:\prompt`

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/get-location?view=powershell-5.1&WT.mc_id=ps-gethelp

Pop-Location

Push-Location

Set-Location