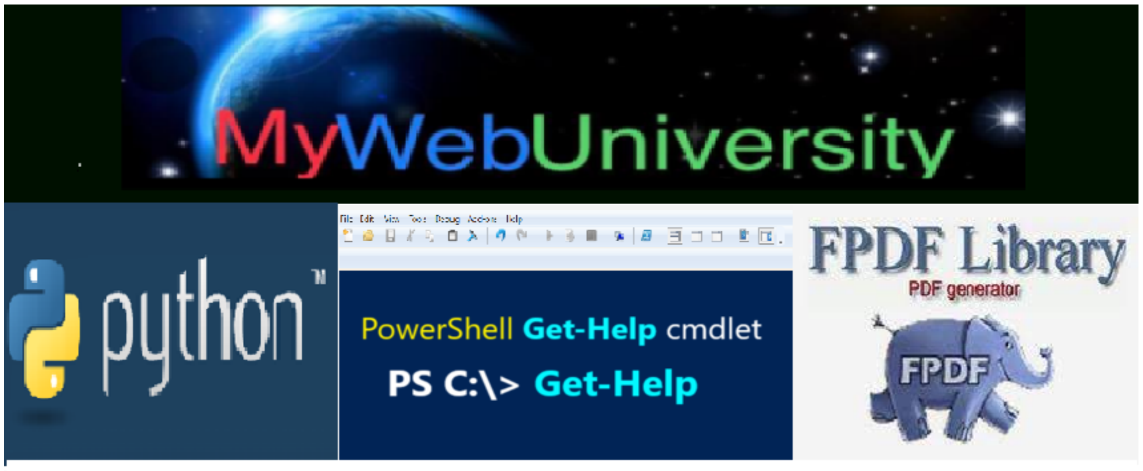

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'cd'

PS:\>Get-HELP cd -Full

NAME

Set-Location

SYNOPSIS

Sets the current working location to a specified location.

SYNTAX

Set-Location -LiteralPath <System.String> [-PassThru] [-UseTransaction] [<CommonParameters>]

Set-Location [[-Path] <System.String>] [-PassThru] [-UseTransaction] [<CommonParameters>]

Set-Location [-PassThru] [-StackName <System.String>] [-UseTransaction] [<CommonParameters>]

DESCRIPTION

The `Set-Location` cmdlet sets the working location to a specified location. That location could be a directory, a subdirectory, a registry location, or any provider

path.

You can also use the StackName parameter to make a named location stack the current location stack. For more information about location stacks, see the Notes.

PARAMETERS

-LiteralPath <System.String>

Specifies a path of the location. The value of the LiteralPath parameter 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
```

-PassThru <System.Management.Automation.SwitchParameter>

Returns a PathInfo object that represents the location. 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>

Specify the path of a new working location. If no path is provided, `Set-Location` defaults to the current user's home directory. When wildcards are used, the

cmdlet chooses the container (directory, registry key, certificate store) that matches the wildcard pattern. If the wildcard pattern matches more than one

container, the cmdlet returns an error.

Required?   false

Position?   0

Default value   None

Accept pipeline input?   True (ByPropertyName, ByValue)

Accept wildcard characters? true

-StackName <System.String>

Specifies an existing location stack name that this cmdlet makes the current location stack. Enter a location stack name. To indicate the unnamed default location

stack, type `$null` or an empty string (`""`).

Using this parameter does not change the current location. It only changes the stack used by the ` -Location` cmdlets. The ` -Location` cmdlets act on the current

stack unless you use the StackName parameter to specify a different stack. For more information about location stacks, see the Notes (#notes).

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

    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 with Path or LiteralPath , this cmdlet returns a PathInfo object representing the new location.

    System.Management.Automation.PathInfoStack

        When you use the PassThru parameter with StackName , this cmdlet returns a PathInfoStack object representing the new stack context.

NOTES

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

        - `cd`

        - `chdir`

- `sl`

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.

The `Set-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).

A stack is a last-in, first-out list in which only the most recently added item can be accessed. 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.

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 `*-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 `Get-Location`.

- To create a new location stack, use the StackName parameter of `Push-Location`. 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 `Set-Location`.

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 (`""`).

------------- Example 1: Set the current location -------------

```
PS C:\> Set-Location -Path "HKLM:\"
PS HKLM:\>
```

This command sets the current location to the root of the `HKLM:` drive.

Example 2: Set the current location and display that location

```
PS C:\> Set-Location -Path "Env:\" -PassThru

Path
----
Env:\

PS Env:\>
```

This command sets the current location to the root of the `Env:` drive. It uses the PassThru parameter to direct PowerShell to return a PathInfo object that

represents the `Env:` location.

Example 3: Set location to the current location in the C: drive

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

The first command sets the location to the root of the `HKLM:` drive in the Registry provider. The second command sets the location to the current location of the

`C:` drive in the FileSystem provider. When the drive name is specified in the form `<DriveName>:` (without backslash), the cmdlet sets the location to the current

location in the PSDrive. To get the current location in the PSDrive use `Get-Location -PSDrive <DriveName>` command.

----- Example 4: Set the current location to a named stack -----

```
PS C:\> Push-Location -Path 'C:\Program Files\PowerShell\' -StackName "Paths"
PS C:\Program Files\PowerShell\> Set-Location -StackName "Paths"
PS C:\Program Files\PowerShell\> Get-Location -Stack

Path
----
C:\
```

The first command adds the current location to the Paths stack. The second command makes the Paths location stack the current location stack. The third command

displays the locations in the current location stack.

The `*-Location` cmdlets use the current location stack unless a different location stack is specified in the command. For information about location stacks, see the

Notes (#notes).

RELATED LINKS

Online Version:

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

Get-Location

Pop-Location

Push-Location