



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 'Show-Command'

PS C:\Users\wahid> Get-Help Show-Command

NAME

Show-Command

SYNOPSIS

Displays PowerShell command information in a graphical window.

SYNTAX

```
Show-Command [[-Name] <System.String>] [-ErrorPopup] [-Height <System.Double>]  
[-NoCommonParameter] [-PassThru] [-Width <System.Double>] [<CommonParameters>]
```

DESCRIPTION

The `Show-Command` cmdlet lets you create a PowerShell command in a command window. You can use the features of the command window to run the command or have it return the command to you.

`Show-Command` is a very useful teaching and learning tool. `Show-Command` works on all command types, including cmdlets, functions, workflows and CIM commands.

Without parameters, `Show-Command` displays a command window that lists all available commands in all installed modules. To find the commands in a module, select the module from the Modules drop-down list. To select a command, click the command name.

To use the command window, select a command, either by using the Name or by clicking the command name in the Commands list. Each parameter set is displayed on a separate tab. Asterisks indicate the mandatory parameters. To enter values for a parameter, type the value in the text box or select the value from the drop-down box. To add a switch parameter, click to select the parameter check box.

When you're ready, you can click Copy to copy the command that you've created to the clipboard or click Run to run the command. You can also use the PassThru parameter to return the command to the host program, such as the PowerShell console. To cancel the command selection and return to the view that displays all commands, press Ctrl and click the selected command.

In the PowerShell Integrated Scripting Environment (ISE), a variation of the `Show-Command` window is displayed by default. For information about using this command window, see the PowerShell ISE Help topics.

This cmdlet was introduced in PowerShell 3.0.

Because this cmdlet requires a user interface, it does not work on Windows Server Core or Windows Nano Server. This cmdlet is only available on Windows systems that support the Windows Desktop.

PARAMETERS

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

Indicates that the cmdlet displays errors in a pop-up window, in addition

to displaying them at the command line. By default, when a command that is

run in a `Show-Command` window generates an error, the error is displayed only at the command line.

Also, when you run the command (by using the Run button in the `Show-Command` window), the `ErrorPopup` parameter returns the command results to the current command, instead of running the command and returning its output to a new command. You can use this feature to save the command results in a variable.

`-Height <System.Double>`

Specifies the height of the `Show-Command` window in pixels. Enter a value between 300 and the number of pixels in the screen resolution. If the value is too large to display the command window on the screen, `Show-Command` generates an error. The default height is 600 pixels. For a `Show-Command` command that includes the `Name` parameter, the default height is 300 pixels.

`-Name <System.String>`

Displays a command window for the specified command. Enter the name of one command, such as the name of a cmdlet, function, or CIM command. If you omit this parameter, `Show-Command` displays a command window that lists all of the PowerShell commands in all modules installed on the computer.

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

Indicates that this cmdlet omits the Common Parameters section of the command display. By default, the Common Parameters appear in an expandable section at the bottom of the command window.

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

Returns an object representing the item with which you are working. By default, this cmdlet does not generate any output. To run the command string, copy and paste it at the command prompt or save it in a variable and use the `Invoke-Expression` cmdlet to run the string in the variable.

`-Width <System.Double>`

Specifies the width of the ``Show-Command`` window in pixels. Enter a value between 300 and the number of pixels in the screen resolution. If the value is too large to display the command window on the screen, ``Show-Command`` generates an error. The default width is 300 pixels.

`<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: Open the Commands window -----

```
Show-Command
```

----- Example 2: Open a cmdlet in the Commands window -----

```
Show-Command -Name "Invoke-Command"
```

----- Example 3: Open a cmdlet with specified parameters -----

```
Show-Command -Name "Connect-PSSession" -Height 700 -Width 1000 -ErrorPopup
```

The `Height` and `Width` parameters specify the dimension of the command window. The `ErrorPopup` parameter displays the error command window.

When you click `Run`, the ``Connect-PSSession`` command runs, just as would if you typed the ``Connect-PSSession`` command at the command line.

- Example 4: Specify new default parameter values for a cmdlet -

```
$PSDefaultParameterValues = @{  
    "Show-Command:Height" = 700  
    "Show-Command:Width" = 1000  
    "Show-Command:ErrorPopup" = $True  
}
```

Now when you run a `Show-Command` command, the new defaults are applied automatically. To use these default values in every PowerShell session, add the `\$PSDefaultParameterValues` variable to your PowerShell profile. For more information, see [about_Profiles](#)

(../Microsoft.PowerShell.Core/About/about_Profiles.md)and

[about_Parameters_Default_Values](#)

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

----- Example 5: Send output to a grid view -----

```
Show-Command Get-ChildItem | Out-GridView
```

The command uses the `Show-Command` cmdlet to open a command window for the `Get-ChildItem` cmdlet. When you click the Run button, the `Get-ChildItem` command runs and generates output. The pipeline operator (|) sends the output of the `Get-ChildItem` command to the `Out-GridView` cmdlet, which displays the `Get-ChildItem` output in an interactive window.

Example 6: Display a command that you create in the Commands window

```
Show-Command -PassThru
```

```
Get-EventLog -LogName "Windows PowerShell" -Newest 5
```

For example, if you use the `Show-Command` window to create a `Get-EventLog` command that gets the five newest events in the Windows PowerShell event log, and then click OK , the command returns the output shown above. Viewing the command string helps you learn PowerShell.

----- Example 7: Save a command to a variable -----

```
$C = Show-Command -PassThru
```

```
$C
```

```
Invoke-Expression $C
```

```
Get-EventLog -LogName "PowerShell" -Newest 5
```

Index	Time	EntryType	Source	InstanceID	Message
11520	Dec 16 16:37	Information	Windows PowerShell	400	Engine state is changed from None to Available...
11519	Dec 16 16:37	Information	Windows PowerShell	600	Provider "Variable" is Started. ...
11518	Dec 16 16:37	Information	Windows PowerShell	600	Provider "Registry" is Started. ...
11517	Dec 16 16:37	Information	Windows PowerShell	600	Provider "Function" is Started. ...
11516	Dec 16 16:37	Information	Windows PowerShell	600	Provider "FileSystem" is Started. ...

The first command uses the PassThru parameter of the `Show-Command` cmdlet and saves the results of the command in the `\$C` variable. In this case, we use the `Show-Command` window to create a `Get-EventLog` command that gets the five newest events in the Windows PowerShell event log. When you click OK, `Show-Command` returns the command string, which is saved in the `\$C` variable.

---- Example 8: Save the output of a command to a variable ----

```
$P = Show-Command Get-Process -ErrorPopup
```

```
$P
```

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
---------	--------	-------	-------	-------	--------	----	-------------

In addition to displaying errors in a window, `ErrorPopup` returns command output to the current command, instead of creating a new command. When you run this command, the `Show-Command` window opens. You can use the window features to set parameter values. To run the command, click the Run button in the `Show-Command` window.

REMARKS

To see the examples, type: `"get-help Show-Command -examples"`.

For more information, type: `"get-help Show-Command -detailed"`.

For technical information, type: `"get-help Show-Command -full"`.

For online help, type: `"get-help Show-Command -online"`