20 Windows PowerShell Commands You Must Know

Windows PowerShell is the new command-line shell, which is **more powerful and scriptable** than Command Prompt. In my three years of experience of using it, I found it really useful, especially if you are into **automating or scripting tasks**. However, most of us either do not know about it or do not prefer using it in place of the old command-line shell.

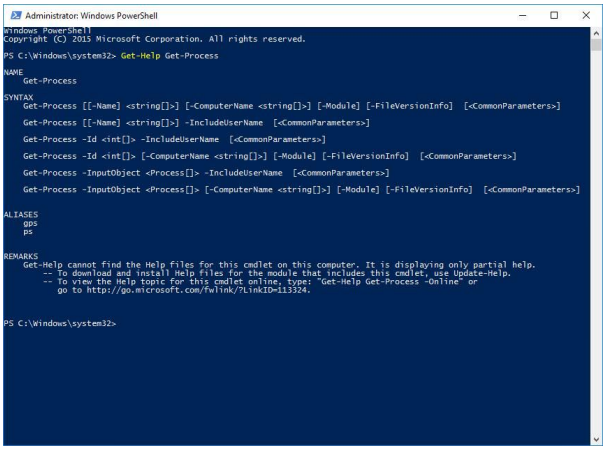
In this post, I am going to share **useful yet straightforward commands** (with examples) of Windows PowerShell. You can use these commands to accomplish numerous tasks — from getting help to starting processes. Shall we begin?

Note: Windows PowerShell is built with **backward compatibility** in mind, and thus supports many commands of the Command Prompt. That said, you can continue using the old commands in its new, colorful interface.

Get-Help [help]

If you are new to PowerShell, you may run into troubles; and in such situations, Get-Help becomes your savior. It provides **necessary information about cmdlets, commands**, functions, scripts, and workflows of the PowerShell.

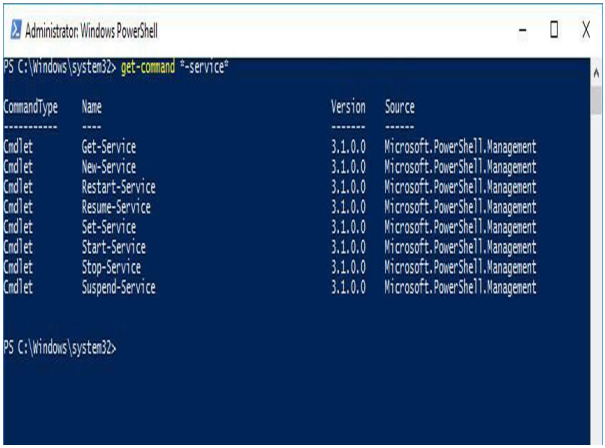
Moreover, it's easy: you need to type `Get-Help` followed by the command, of which, you seek the details. For example, you can get information about "Get-Process" using `Get-Help Get-Process`.



Get-Command [gcm]

Windows PowerShell allows discovering its commands and features using Get-Command. It displays the list of **commands of a specific feature** or for a specific purpose based on your search parameter.

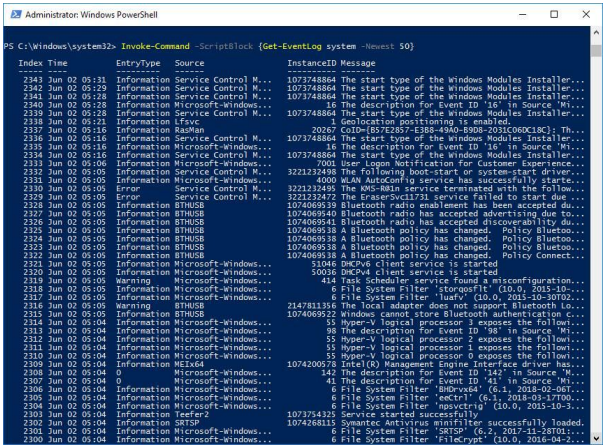
You only need to type `Get-Command` followed by your search query in the PowerShell. For example, `Get-Command *-service*` displays commands with "-service" in its name. Please remember to **use the asterisks on both sides of your query** because it is a wild card that helps to search for the unknown.



Invoke-Command [icm]

When you wish to **run a command or a script** of PowerShell — locally or remotely on single or multiple computer(s) — “Invoke-Command” is your friend. It is easy-to-use and helps you to batch-control computers.

You must type `Invoke-Command` followed by the command or the script with its complete path. For example, you can **run a command “Get-EventLog”**using `Invoke-Command -ScriptBlock {Get-EventLog system -Newest 50}` or on a remote computer “Server01” using `Invoke-Command -ScriptBlock {Get-EventLog system -Newest 50} -ComputerName Server01` .

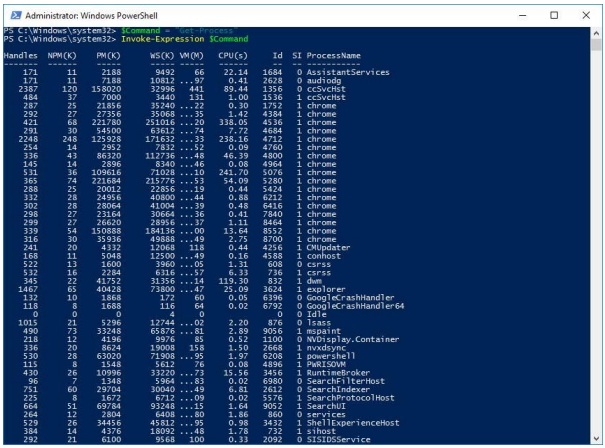


Invoke-Expression [iex]

Invoke-Expression **runs another command or expression**. If you are providing an expression or a string as its input, this command first evaluates it, then runs it, but also works only locally, unlike the previous command.

You must type `Invoke-Expression` followed by a command or an expression. For instance, you can assign a variable “$Command” with a string telling the command “Get-Process”. When you run `Invoke-Expression`

`$Command` , “Get-Process” gets run as a command on your local computer.

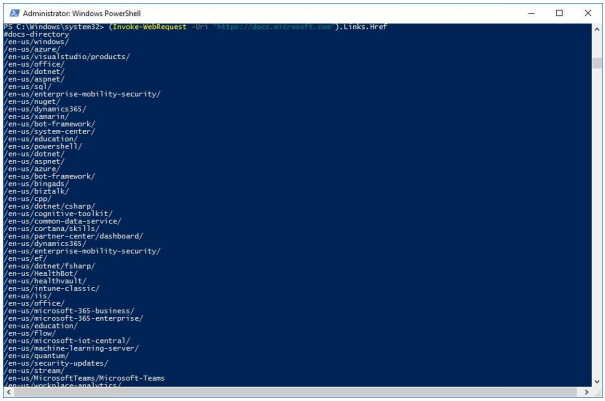


Invoke-WebRequest [iwr]

You can **download, log in, and scrape for information** on websites and web services while working on Windows PowerShell using the Invoke-WebRequest.

You must use it like `Invoke-WebRequest` followed by its parameters. For example, you can get the links on a given

web page by using the command as `(Invoke-WebRequest -Uri "https://docs.microsoft.com").Links.Href`.

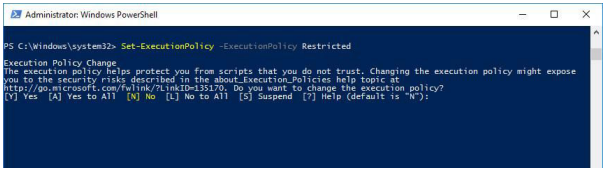


Set-ExecutionPolicy

Although creating and executing scripts (having extension “ps1”) in Windows PowerShell is possible; however, there are restrictions for security purposes. But you can **switch the security level** using the Set-ExecutionPolicy command.

You can type `Set-ExecutionPolicy` followed by one of the four security levels — **Restricted, Remote Signed, All Signed, or Unrestricted** to use the command. For example,

you can assign the restricted policy status using `Set-ExecutionPolicy -ExecutionPolicy Restricted`.

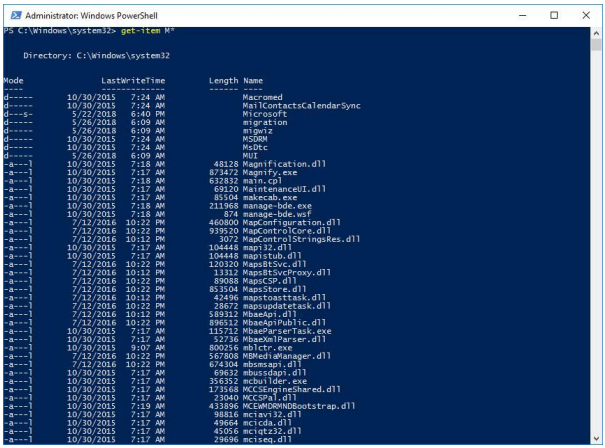


Get-Item [gi]

If you are looking for **information on an item** at any given location, say a file on your hard disk, Get-Item is the best way to acquire it in Windows PowerShell. You must know that **it does not get the contents of the item**, such as files and sub-directories in a given directory unless explicitly specified by you.

You must type `Get-Item` followed by a path or a string along with its parameters if any. For example, you can get all the items (files or folders) beginning with “M” in the current

directory using `Get-Item M*`. Along with content of directories, it can also get the content of registry keys.



Copy-Item [copy]

If you need to **copy files and directories** on your storage disk or registry entries and keys in the registry, you can use Copy-Item. It functions similar to the “cp” command we have in the Command Prompt, but it is much better.

You can make use of `Copy-Item` command to **copy and rename items in the same command** as well — give a new name as the destination. For instance, you can copy and

rename “Services.htm” to “MyServices.txt” using `Copy-Item "C:\Services.htm" -Destination "C:\MyData\MyServices.txt"` .

Remove-Item [del]

If you wish **to delete items** such as files, folders, functions, and registry keys and variables, Remove-Item is the command for you. What I found interesting is, it provides parameters to include and exclude items.

You can make use of `Remove-Item` command to **delete items from specific locations using parameters**. For example, you can delete the file “MyServices.txt” with the command `Remove-Item "C:\MyData\MyServices.txt"` .

Get-Content [cat]

When you need to view the content of a text file at a specific location, you open and read it in a code/text editor like

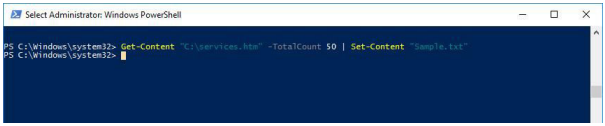
Notepad++. In Windows PowerShell, you can use Get-Content to **retrieve the content** without opening the file.

For example, you can retrieve 50 lines of content of “Services.htm”, then you can use `Get-Content "C:\Services.htm" -TotalCount 50`.

Set-Content [sc]

You can save text to files using Set-Content, similar to the “echo” command of the Bash Shell. In combination with the Get-Content, you can also **retrieve the content of one file and copy it into another file**using this command.

For example, you can type `Set-Content` to **write or replace the content** of a file with new content. Moreover, you can club it with the previous command’s example to save its output into a new file named “Sample.txt” using `Get-Content "C:\Services.htm" -TotalCount 50 | Set-Content "Sample.txt"`.



Get-Variable [gv]

If you are looking to use variables in Windows PowerShell, Get-Variable command helps you to **visualize the values of variables**. It shows them in a tabular form and allows including, excluding, and using wildcards.

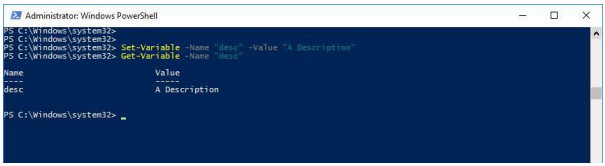
You can use this command by typing `Get-Variable` followed by its options and parameters. For example, you can retrieve the value for a variable named "desc" using the following code: `Get-Variable -Name "desc"`.

Set-Variable [set]

You can **assign or change/reset the value of a variable** using the command Set-Variable. As a shortcut, you

can also set a simple variable using the format ${ **$VarName = VarValue**}$, like `$desc = "A Description"`.

You can use the command `Set-Variable` followed by its parameters to set a variable. For instance, we can set the value for a variable named “desc” using the command `Set-Variable -Name "desc" -Value "A Description"`.



Get-Process [gps]

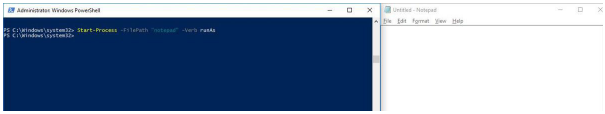
We usually use Task Manager to **find the running processes** on our computer. In Windows PowerShell, anyone can use Get-Process to get the list of currently running processes, which you can further process as well.

You can write the command as `Get-Process` along with your search query. For example, if you need information about the processes with “explore” in their name, you can type `Get-Process *explore*` (note the asterisks).

Start-Process [saps]

Windows PowerShell makes it easy to **start one or more processes** on your computer. I found this command is handy in scripting apps since it is one of the must-have commands you will need for automating a task.

You can type `Start-Process` followed by its parameters to use the command. For instance, you can start Notepad by typing `Start-Process -FilePath "notepad" -Verb runAs` in the Windows PowerShell.



Stop-Process [kill]

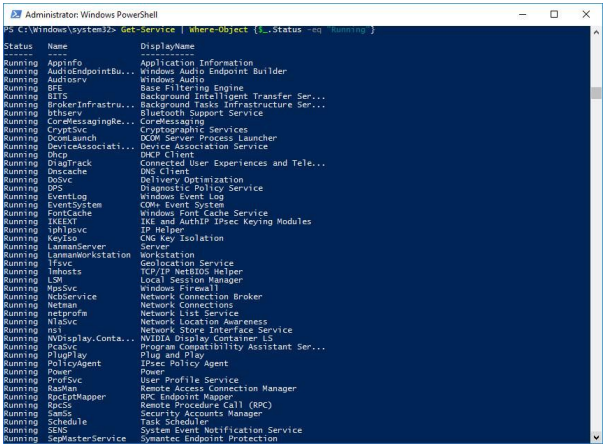
You can **stop specific or all instances of a process running** on your computer using its name or PID (Process ID), thanks to the command Stop-Process. What makes it compelling is, you can detect a process is stopped or not and you can even stop the processes not owned or started by the current user.

You can type the command `Stop-Process` followed by its parameters to stop the given processes. For example, you can stop all the processes of Notepad using the command `Stop-Process -Name "notepad"`.

Get-Service [gsv]

When you need **information on specific services** (running or stopped) on your computer, you can use Get-Service. It displays the services installed in your system and provides options to filter and include and exclude them.

If you wish to use this command, you can type `Get-Service` followed by its parameters. For example, type the following `Get-Service | Where-Object {$_.Status -eq "Running"}` to get the services “running” on your system.



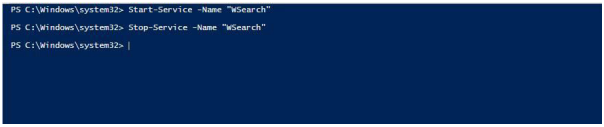
Start-Service [sasv]

If you wish to **start a service** on your computer, Start-Service command can help you do the same from Windows PowerShell. I found it is powerful enough to start a service even if that service is disabled on your computer.

You need to specify the name of the service while using the command `Start-Service`. For instance, `Start-Service -Name "WSearch"` starts the service “Windows Search” on your local computer.

Stop-Service [spsv]

If you wish to **stop services running**on your computer, Stop-Service command will prove helpful. You need to specify the name of the service along with `Stop-Service`. For instance, type `Stop-Service -Name "WSearch"` to stop the service “Windows Search” on your computer.

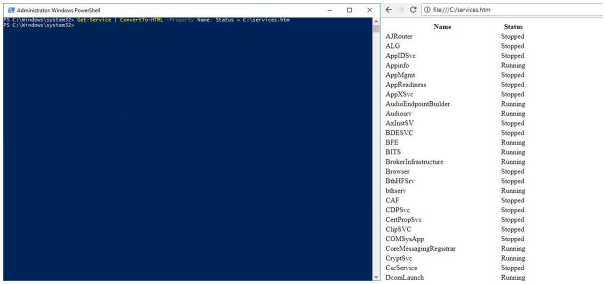


ConvertTo-HTML

PowerShell can provide amazing information about your system. However, it is mostly presented in an indigestible

format, but you can use ConvertTo-HTML to **create and format a report** to analyze it or send it to someone.

You can use `ConvertTo-HTML` along with the output of another command using piping. For example `Get-Service | ConvertTo-HTML -Property Name, Status > C:\Services.htm` displays the list of all the services and their status in the form of a web report, which is stored in the file “Services.htm”.



That is all about the essential commands you must know of the Windows PowerShell.