

# 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.

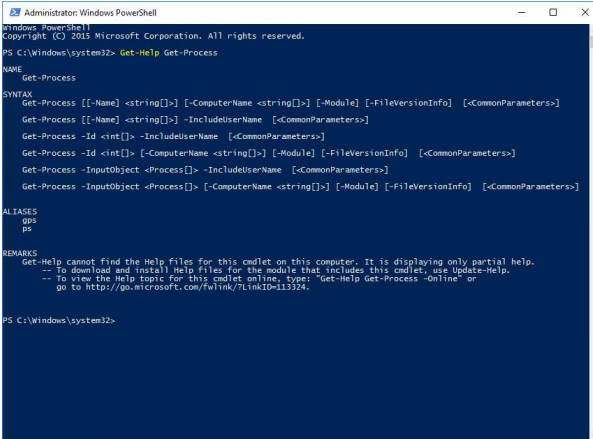
In this post, I am going to share **useful yet straightforward commands** (with examples) of Windows PowerShell. You can use these commands to accomplish numerous tasks — from getting help to starting processes. Shall we begin?

**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` .



```
Administrator Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Get-Help Get-Process

NAME
----
Get-Process

SYNOPSIS
Get-Process [-Name <string[]>] [-ComputerName <string[]>] [-Module <FileVersionInfo>] [<CommonParameters>]
Get-Process [-Name <string[]>] -IncludeUserName [<CommonParameters>]
Get-Process -Id <int[]> -IncludeUserName [<CommonParameters>]
Get-Process -Id <int[]> [-ComputerName <string[]>] [-Module <FileVersionInfo>] [<CommonParameters>]
Get-Process -InputObject <Process[]> -IncludeUserName [<CommonParameters>]
Get-Process -InputObject <Process[]> [-ComputerName <string[]>] [-Module <FileVersionInfo>] [<CommonParameters>]

ALIASES
-----
gps
ps

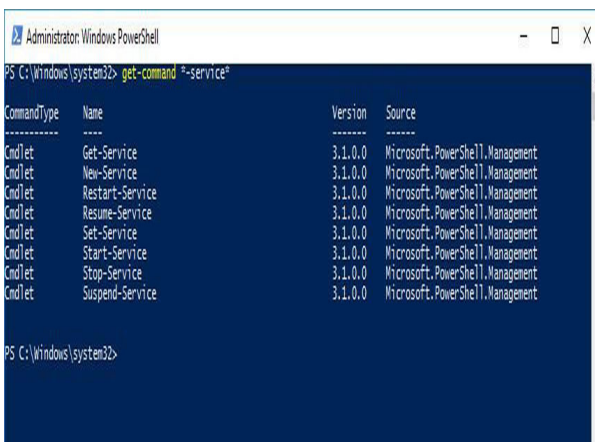
REMARKS
-----
Get-Help cannot find the Help files for this cmdlet on this computer. It is displaying only partial help.
-- To download and install Help files for the module that includes this cmdlet, use Update-Help.
-- To view the Help topic for this cmdlet online, type: "Get-Help Get-Process -Online" or
go to https://go.microsoft.com/fwlink/?LinkID=113324.

PS C:\Windows\system32>
```

# 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.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> get-command *-service*

CommandType Name                Version      Source
-----
Cmdlet      Get-Service         3.1.0.0     Microsoft.PowerShell.Management
Cmdlet      New-Service         3.1.0.0     Microsoft.PowerShell.Management
Cmdlet      Restart-Service     3.1.0.0     Microsoft.PowerShell.Management
Cmdlet      Resume-Service      3.1.0.0     Microsoft.PowerShell.Management
Cmdlet      Set-Service         3.1.0.0     Microsoft.PowerShell.Management
Cmdlet      Start-Service       3.1.0.0     Microsoft.PowerShell.Management
Cmdlet      Stop-Service        3.1.0.0     Microsoft.PowerShell.Management
Cmdlet      Suspend-Service     3.1.0.0     Microsoft.PowerShell.Management

PS C:\Windows\system32>
```

# 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`.

```
Administrator: Windows PowerShell

PS C:\Windows\system32> Invoke-Command -ScriptBlock {Get-Eventlog system -Newest 50}

Index Time EntryType Source InstanceID Message
-----
2343 Jun 02 05:13 Information Service Control M... 1027248864 The start type of the windows Modules Installer...
2342 Jun 02 05:12 Information Service Control M... 1027248864 The start type of the windows Modules Installer...
2341 Jun 02 05:12 Information Service Control M... 1027248864 The start type of the windows Modules Installer...
2340 Jun 02 05:12 Information Microsoft-Windows... 16 The description for Event ID 16 in Source 'Win...
2339 Jun 02 05:12 Information Service Control M... 1027248864 The start type of the windows Modules Installer...
2338 Jun 02 05:12 Information Ifxave 1 Collocation partitioning is enabled.
2337 Jun 02 05:10 Information RamMan 20267 C0D6(B572B57-61B5-49A0-8908-2031C060438C); Th...
2336 Jun 02 05:10 Information Service Control M... 1027248864 The start type of the windows Modules Installer...
2335 Jun 02 05:10 Information Microsoft-Windows... 16 The description for Event ID 16 in Source 'Win...
2334 Jun 02 05:10 Information Service Control M... 1027248864 The start type of the windows Modules Installer...
2333 Jun 02 05:10 Information Microsoft-Windows... 7001 User Logon Notification for Customer Experience...
2332 Jun 02 05:10 Information Microsoft-Windows... 4000 WLAN AutoConfig service has successfully starte...
2331 Jun 02 05:10 Error Service Control M... 3221232408 The RNS-800n service terminated with the follow...
2330 Jun 02 05:10 Error Service Control M... 3221232472 The EaserSvc32724 service failed to start due ...
2329 Jun 02 05:10 Information BTHUSB 1074069930 Bluetooth radio enablement has been accepted du...
2328 Jun 02 05:10 Information BTHUSB 1074069940 Bluetooth radio has accepted advertising due to...
2327 Jun 02 05:10 Information BTHUSB 1074069941 Bluetooth radio has accepted discoverability due...
2326 Jun 02 05:10 Information BTHUSB 1074069938 A Bluetooth policy has changed. Policy Bluetoo...
2325 Jun 02 05:10 Information BTHUSB 1074069938 A Bluetooth policy has changed. Policy Bluetoo...
2324 Jun 02 05:10 Information BTHUSB 1074069938 A Bluetooth policy has changed. Policy Bluetoo...
2323 Jun 02 05:10 Information BTHUSB 1074069938 A Bluetooth policy has changed. Policy Bluetoo...
2322 Jun 02 05:10 Information Microsoft-Windows... 11046 DPCV4 client service is started
2321 Jun 02 05:10 Information Microsoft-Windows... 50036 DPCV4 client service is started
2320 Jun 02 05:10 Information Microsoft-Windows... 414 Test Scheduler service found a misconfiguration...
2319 Jun 02 05:10 Warning Microsoft-Windows... 6 File System Filter 'storgsflt' (10.0, 2015-10-...
2318 Jun 02 05:10 Warning Microsoft-Windows... 6 File System Filter 'luffy' (10.0, 2015-10-30702...
2317 Jun 02 05:10 Warning BTHUSB 2147811336 The local adapter does not support Bluetooth lo...
2316 Jun 02 05:10 Information Microsoft-Windows... 1074069922 Windows cannot store Bluetooth authentication c...
2315 Jun 02 05:10 Information BTHUSB 55 Hyper-V logical processor 3 exposes the follow...
2314 Jun 02 05:10 Information Microsoft-Windows... 55 Hyper-V logical processor 3 exposes the follow...
2313 Jun 02 05:10 Information Microsoft-Windows... 55 Hyper-V logical processor 2 exposes the follow...
2312 Jun 02 05:10 Information Microsoft-Windows... 55 Hyper-V logical processor 2 exposes the follow...
2311 Jun 02 05:10 Information Microsoft-Windows... 55 Hyper-V logical processor 1 exposes the follow...
2310 Jun 02 05:10 Information Microsoft-Windows... 55 Hyper-V logical processor 1 exposes the follow...
2309 Jun 02 05:10 Information Microsoft-Windows... 1074200578 Intel(R) Management Engine Interface driver has...
2308 Jun 02 05:10 Information Microsoft-Windows... 42 The description for Event ID 42 in Source 'Win...
2307 Jun 02 05:10 Information Microsoft-Windows... 41 The description for Event ID 41 in Source 'Win...
2306 Jun 02 05:10 Information Microsoft-Windows... 6 File System Filter 'libvase4' (6.1, 2016-02-08)...
2305 Jun 02 05:10 Information Microsoft-Windows... 6 File System Filter 'aectrl' (6.1, 2018-03-17700...
2304 Jun 02 05:10 Information Microsoft-Windows... 6 File System Filter 'inspctrp' (10.0, 2015-10-...
2303 Jun 02 05:10 Information Teefor2 102754325 Service started successfully.
2302 Jun 02 05:10 Information SRTSP 1074208315 Symantec Antivirus minifilter successfully loaded...
2301 Jun 02 05:10 Information Microsoft-Windows... 6 File System Filter 'SRTSP' (6.2, 2017-11-28701...
2300 Jun 02 05:10 Information Microsoft-Windows... 6 File System Filter 'FileCrypt' (10.0, 2016-04-...
```

# 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.

Handles	NPM(K)	PM(K)	VS(K)	VM(K)	CPU(s)	Id	SI	ProcessName
171	11	2188	9492	66	22.14	1684	0	AssistantServices
171	11	7188	10812	..57	0.41	2628	0	audiodg
2187	120	153020	32996	441	89.44	1356	0	csrss
484	37	7000	3440	131	1.00	1536	1	csrss
287	25	21888	37400	..22	0.30	1752	1	chrome
292	27	27356	35068	..35	1.42	4384	1	chrome
421	68	221760	218016	..20	338.09	4536	1	chrome
291	10	34500	63612	..74	7.72	4684	1	chrome
2148	248	124920	171632	..23	236.10	4712	1	chrome
254	14	2852	7832	..52	0.09	4760	1	chrome
136	43	86310	113736	..48	46.39	4800	1	chrome
145	14	2896	8340	..46	0.08	4964	1	chrome
131	36	108816	21081	..10	248.70	1076	1	chrome
363	74	221884	215776	..53	34.09	5280	1	chrome
288	25	20012	23356	..19	0.48	5424	1	chrome
132	28	24936	40880	..44	0.88	6212	1	chrome
102	29	20064	41896	..39	0.48	6416	1	chrome
298	27	23164	30664	..36	0.41	7840	1	chrome
399	27	26620	28936	..37	1.11	8664	1	chrome
139	54	150888	164136	..00	13.64	8552	1	chrome
316	30	35936	49888	..49	2.75	8700	1	chrome
241	10	4372	12068	..18	0.44	9216	1	cmd
168	11	1048	12500	..49	0.10	4588	1	cmd
222	13	1600	3960	..05	1.31	608	0	csrss
532	16	2284	6316	..57	6.33	736	1	csrss
145	22	41752	11316	..14	119.30	832	1	cmd
1467	65	40428	73800	..47	25.09	3624	1	explorer
112	10	1868	1172	..60	0.09	1396	0	GoogleCrashHandler64
118	8	1888	116	..64	0.02	6792	0	GoogleCrashHandler64
0	0	0	0	..4	0	0	0	Idle
1015	21	5296	12744	..02	2.20	876	0	lsass
490	73	33248	61576	..81	1.89	8976	1	mspaint
218	12	4196	9976	..85	0.52	1100	0	NVDisplay.Container
136	20	4624	19008	..38	1.30	2688	1	Notepad
530	28	60320	71988	..95	1.97	6208	1	powershell
113	8	1348	368	..26	0.00	408	1	PowerShell
430	26	10996	33220	..73	15.58	3456	1	RunTimeBroker
96	7	1348	5964	..83	0.02	6880	0	SearchFilterHost
71	60	29704	10040	..49	6.81	2812	0	SearchIndexer
225	8	1672	6712	..09	0.02	5576	1	SearchProtocolHost
684	51	69784	137248	..15	1.64	9052	1	SearchUI
264	12	7804	6408	..80	1.86	860	0	services
729	26	94856	45812	..95	0.98	8432	1	ShellExperienceHost
384	14	4376	18092	..48	1.78	732	1	shost
72	21	6100	8368	..100	0.31	2092	0	StlBService

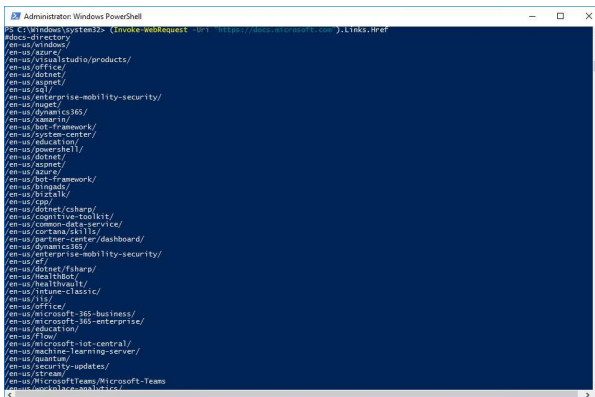
# Invoke-WebRequest

## [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 .



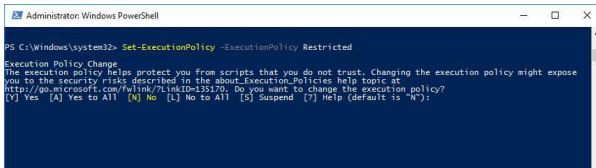
```
Administrator Windows PowerShell
PS C:\Windows\system32 (Invoke-WebRequest -Uri https://docs.microsoft.com).Links.Href
/microsoft/
/en-us/windows/
/en-us/azure/
/en-us/visualstudio/products/
/en-us/office/
/en-us/dotnet/
/en-us/aspnet/
/en-us/sql/
/en-us/enterprise-mobility-security/
/en-us/nuget/
/en-us/dynamics365/
/en-us/axar-in/
/en-us/azure-framework/
/en-us/system-center/
/en-us/education/
/en-us/powershell/
/en-us/dotnet/
/en-us/aspnet/
/en-us/azure/
/en-us/azure-framework/
/en-us/bringd/
/en-us/bringd/
/en-us/csp/
/en-us/dotnet/csharp/
/en-us/cognitive-toolkit/
/en-us/common-data-service/
/en-us/cortana/skills/
/en-us/partner-center/dashboard/
/en-us/dynamics365/
/en-us/enterprise-mobility-security/
/en-us/ef/
/en-us/dotnet/fsharp/
/en-us/health/
/en-us/healthcare/
/en-us/office-classic/
/en-us/office/
/en-us/microsoft-365-business/
/en-us/microsoft-365-enterprise/
/en-us/education/
/en-us/flow/
/en-us/microsoft-iot-central/
/en-us/machine-learning-server/
/en-us/quantum/
/en-us/security-updates/
/en-us/stream/
/en-us/microsoft-teams/microsoft-teams/
/en-us/microsoft-teams/
```

# 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`.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> Set-ExecutionPolicy -ExecutionPolicy Restricted

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at
http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"):
```

# 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.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> get-item M*

Directory: C:\Windows\system32

Mode                LastWriteTime         Length Name
----                -
d----- 10/30/2015 7:24 AM                Macromed
d----- 10/30/2015 7:24 AM                MailContactsCalendarSync
d----- 5/22/2018 6:40 PM                Microsoft
d----- 5/26/2018 6:09 AM                Migration
d----- 5/26/2018 6:09 AM                mspwiz
d----- 10/30/2015 7:24 AM                MSOM
d----- 10/30/2015 7:24 AM                MstDc
d----- 5/26/2018 6:09 AM                MUI
d----- 10/30/2015 7:18 AM                48128 Magnification.dll
d----- 10/30/2015 7:17 AM                82412 Magnify.exe
d----- 10/30/2015 7:18 AM                632832 main.cpl
d----- 10/30/2015 7:17 AM                69180 MaintenanceUI.dll
d----- 10/30/2015 7:17 AM                85194 malecab.exe
d----- 10/30/2015 7:18 AM                211968 manage-bde.exe
d----- 7/12/2016 10:22 PM                460800 MapConfiguration.dll
d----- 7/12/2016 10:22 PM                939200 MapControlCore.dll
d----- 7/12/2016 10:12 PM                3072 MapControlStringsRes.dll
d----- 10/30/2015 7:17 AM                164448 mapis32.dll
d----- 10/30/2015 7:17 AM                104448 mapistub.dll
d----- 7/12/2016 10:22 PM                100320 MapsApiSvc.dll
d----- 7/12/2016 10:12 PM                13112 MapsApiSvcProxy.dll
d----- 7/12/2016 10:22 PM                8008 MapsStore.dll
d----- 7/12/2016 10:12 PM                813104 MapsStore.dll
d----- 7/12/2016 10:12 PM                12416 MapsUpdateTask.dll
d----- 7/12/2016 10:22 PM                28672 MapsUpdateTask.dll
d----- 7/12/2016 10:12 PM                58912 Mbaapi.dll
d----- 7/12/2016 10:22 PM                89632 MbaapiPublic.dll
d----- 10/30/2015 7:17 AM                11712 MbaapiUpdateTask.exe
d----- 10/30/2015 7:17 AM                12736 MbaapiParser.dll
d----- 10/30/2015 6:07 AM                800256 mbicr.exe
d----- 7/12/2016 10:22 PM                567808 MBMediaManager.dll
d----- 7/12/2016 10:22 PM                67424 Mbsnap.dll
d----- 10/30/2015 7:17 AM                69632 Mbsnap.dll
d----- 10/30/2015 7:17 AM                35632 Mchui.ldr.exe
d----- 10/30/2015 7:17 AM                173408 MCCSuppInetShared.dll
d----- 10/30/2015 7:17 AM                12040 MCCSPA.dll
d----- 10/30/2015 7:19 AM                433896 MCCSuppOutstrap.dll
d----- 10/30/2015 7:17 AM                98816 Mclm32.dll
d----- 10/30/2015 7:17 AM                49664 Mclm32.dll
d----- 10/30/2015 7:17 AM                45056 Mclm32.dll
d----- 10/30/2015 7:17 AM                89632 Mclm32.dll
```

# 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](#)

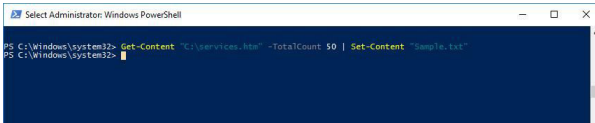
[Notepad++](#). In Windows PowerShell, you can use Get-Content to **retrieve the content** without opening the file.

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.

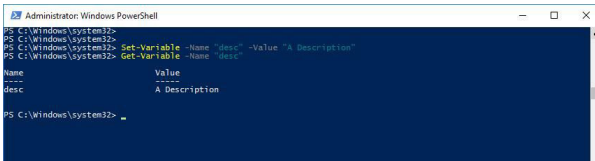
You can use this command by typing `Get-Variable` followed by its options and parameters. For example, you can retrieve the value for a variable named “desc” using the following code: `Get-Variable -Name "desc" .`

# 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"` .



```
Administrator: Windows PowerShell
PS C:\Windows\system32>
PS C:\Windows\system32>
PS C:\Windows\system32> Set-Variable -Name "desc" -Value "A Description"
PS C:\Windows\system32> Get-Variable -Name "desc"

Name      Value
----      -
desc      A Description

PS C:\Windows\system32> _
```

# 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.

---





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.

```
PS C:\Windows\system32> Start-Service -Name "WSearch"  
PS C:\Windows\system32> Stop-Service -Name "WSearch"  
PS C:\Windows\system32> |
```

## 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”.

The screenshot shows the Windows Task Manager application with the 'Services' tab selected. The 'Status' column is highlighted in blue. The list of services and their statuses are as follows:

Service Name	Status
AFDiscuss	Stopped
ALG	Stopped
Arc42Svc	Stopped
AppHost	Running
AppMgmt	Stopped
AppReadiness	Stopped
AppXSvc	Stopped
AsyncHostBuilder	Running
AudioSrv	Running
BDESVC	Stopped
BITS	Running
Browser	Running
BSMF	Stopped
CAA	Running
CDPSvc	Stopped
CryptSvc	Stopped
CryptSp	Stopped
COMSysApp	Running
CryptspSrv	Running
Cryptui	Running
CscService	Stopped
DismHost	Running
DismSvc	Running

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