

Linux & PowerShell Professional Mastery

PowerShell Command Reference & Cheat Sheet

Master the Command Line. Understand Your Systems. Secure Your Network. Automate Your Work.

30 Days: From Beginner to Advanced IT Professional
Ubuntu 24.04 LTS · Windows 11 · Windows PowerShell 5.1 · PowerShell 7.x

Edition 2026.11.08 · Days 01–02 of 30 · Built 2026-11-08

Copyright © 2026 Linux & PowerShell Professional Mastery contributors. Released under the MIT License.

Contents

1. PowerShell Command Reference 3
2. Bash ↔ PowerShell Side-by-Side 10

PowerShell Command Reference

Platforms: **Windows PowerShell 5.1** (Windows 11) and **PowerShell 7.4+** (Windows, Linux, macOS). Version and OS restrictions are noted per entry.

Navigation & System Information

Get-Location

Field	Detail
Version / OS	5.1, 7.x · all OS
Description	Returns the current location as a `PathInfo` object. Alias: `pwd`, `gl`.
Syntax	`Get-Location [-PSProvider <String[]>] [-PSDrive <String[]>] [-Stack]`
Parameters	`-Stack` shows the `Push-Location` stack
Example	`(Get-Location).Path`
Expected output	`C:\Users\student`
Equivalent	`pwd`
Common errors	Using the object where a string is required. Use `.Path`.
Use case	Building absolute paths in scripts: `Join-Path (Get-Location) 'out.csv'`.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-location

Get-ChildItem

Field	Detail
Version / OS	5.1, 7.x · all OS (aliases `ls`, `dir`, `gci`; `ls` only on Windows)
Description	Gets the items in one or more locations (files, folders, registry keys, certificates).
Syntax	`Get-ChildItem [[-Path] <String[]>] [[-Filter] <String>] [-Include] [-Exclude] [-Recurse] [-Depth <UInt32>] [-Force] [-Name] [-File] [-Directory] [-Hidden]`
Parameters	`-Filter *.log` fast filter · `-Recurse`/`-Depth` · `-Force` hidden/system · `-File`/`-Directory` · `-Name` strings only
Example	`Get-ChildItem C:\Windows\Logs -Recurse -File | Sort-Object Length -Descending | Select-Object -First 5 FullName, Length`
Expected output	A table of the five largest log files with `FullName` and `Length` (bytes)
Equivalent	`ls -la`, `find`, `ls -R`

Field	Detail
Common errors	`Access to the path ... is denied` during `-Recurse`: add `-ErrorAction SilentlyContinue` only when you understand what is skipped
Use case	Inventory files by size or date; find recently changed files.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-childitem

Get-ComputerInfo

Field	Detail
Version / OS	5.1+, 7.x · **Windows only**
Description	Returns a consolidated object with ~180 OS, hardware, BIOS and configuration properties. Alias: `gin`.
Syntax	`Get-ComputerInfo [[-Property] <String[]>]`
Parameters	`-Property` limit to named properties (wildcards allowed)
Example	`Get-ComputerInfo -Property OsName, OsVersion, OsArchitecture, CsTotalPhysicalMemory`
Expected output	`Microsoft Windows 11 Home Single Language 10.0.26200 64-bit ...`
Equivalent	`uname -a`, `cat /etc/os-release`, `hostnamectl`
Common errors	Slow without `-Property` · not available on Linux PowerShell
Use case	Help-desk triage and patch-compliance inventory.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-computerinfo

Get-Date

Field	Detail
Version / OS	5.1, 7.x · all OS (`-AsUTC` needs 7.1+)
Description	Returns the current (or a specified) date and time as a `DateTime` object, or a formatted string.
Syntax	`Get-Date [[-Date] <DateTime>] [-Format <String>] [-UFormat <String>] [-AsUTC]`
Parameters	`-Format 'yyyy-MM-dd HH:mm:ss'` · `-Format o` ISO 8601 · `-UFormat %Y-%m-%d` · `-AsUTC`
Example	`(Get-Date).AddDays(-30).ToString('yyyy-MM-dd')`
Expected output	`2026-10-09`
Equivalent	`date`, `date -d '30 days ago' +%F`
Common errors	Using `-Format` and then date arithmetic (the result is a string)
Use case	Filtering logs to "last 24 hours": `(Get-Date).AddHours(-24)`.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-date

$PSVersionTable · $env: variables

Field	Detail
Version / OS	5.1, 7.x · all OS
Description	`$PSVersionTable` holds engine version and edition. `$env:NAME` reads environment variables (`USERNAME`, `COMPUTERNAME`, `TEMP`, `USERPROFILE`).
Example	`$PSVersionTable.PSEdition; $env:COMPUTERNAME`
Expected output	`Desktop` / `WIN11-LAB`
Equivalent	`echo $BASH_VERSION`, `echo $USER $HOSTNAME`
Common errors	On Linux use `$env:USER` (not `USERNAME`); `$env:COMPUTERNAME` is empty there
Use case	Scripts that branch on edition or OS (`if ($IsLinux) { ... }` in 7).
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_automatic_variables

Help & Discovery

Get-Help

Field	Detail
Version / OS	5.1, 7.x · all OS
Description	Displays help for cmdlets, functions and `about_` topics. Alias: `help`.
Syntax	`Get-Help [[-Name] <String>] [-Examples | -Detailed | -Full | -Online | -Parameter <String>]`
Example	`Get-Help Remove-Item -Parameter WhatIf`
Expected output	Parameter description, type, position and whether it is required
Equivalent	`man`, `--help`, `help`
Common errors	Only syntax shown, because help is not downloaded. Run `Update-Help` (elevated in 5.1; `-Scope CurrentUser` in 7).
Use case	Learning a cmdlet offline.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/get-help

Get-Command

Field	Detail
Version / OS	5.1, 7.x · all OS. Alias: `gcm`
Description	Finds commands (cmdlets, functions, aliases, executables).
Syntax	`Get-Command [[-Name] <String[]>] [-Verb <String[]>] [-Noun <String[]>] [-Module <String[]>] [-All]`

Field	Detail
Example	`Get-Command -Noun Item`
Expected output	`Clear-Item, Copy-Item, Get-Item, Invoke-Item, Move-Item, New-Item, Remove-Item, Rename-Item, Set-Item`
Equivalent	`type`, `which`, `apropos`, `compgen -c`
Common errors	None common
Use case	Discover commands by noun; resolve which `ls` will actually run.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/get-command

Get-Alias

Field	Detail
Version / OS	5.1, 7.x (fewer aliases on Linux/macOS)
Description	Lists aliases or resolves one.
Example	`Get-Alias -Definition Remove-Item`
Expected output	`del, erase, rd, ri, rm, rmdir`
Equivalent	`alias`
Use case	Translating someone's alias-heavy one-liner into a readable script.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-alias

Files & Directories

Set-Location

Field	Detail
Version / OS	5.1, 7.x · all OS. Aliases: `cd`, `chdir`, `sl`
Description	Changes the current location (file system, registry, certificate store...).
Syntax	`Set-Location [[-Path] <String>] [-LiteralPath <String>] [-PassThru]`
Parameters	`-` / `+` history navigation (6.2+) · `-LiteralPath` · `-PassThru`
Example	`Set-Location HKLM:\SOFTWARE\Microsoft`
Expected output	Prompt becomes `PS HKLM:\SOFTWARE\Microsoft>`
Equivalent	`cd`
Common errors	`Cannot find path ... because it does not exist`
Use case	Navigating the registry like a file system.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/set-location

New-Item

Field	Detail
Version / OS	5.1, 7.x · all OS. Alias: `ni` (`mkdir` is a wrapper function on Windows)
Description	Creates files, directories, links and registry keys.
Syntax	`New-Item [-Path] <String[]> [-ItemType <String>] [-Name <String>] [-Value <Object>] [-Force]`
Parameters	`-ItemType File/Directory/SymbolicLink/Junction/HardLink` · `-Value` initial content · `-Force` (⚠ overwrites existing **files**)
Example	`New-Item -Path C:\Lab\logs -ItemType Directory -Force`
Expected output	`Directory: C:\Lab` … `d----- ... logs`
Equivalent	`mkdir -p`, `touch`
Common errors	`An item with the specified name ... already exists` (no `-Force`)
Use case	Idempotent folder creation in setup scripts.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/new-item

Copy-Item

Field	Detail
Version / OS	5.1, 7.x · all OS. Aliases: `copy`, `cp` (Windows), `cpi`
Description	Copies items.
Syntax	`Copy-Item [-Path] <String[]> [[-Destination] <String>] [-Recurse] [-Force] [-Filter] [-Include] [-Exclude] [-PassThru] [-WhatIf]`
Example	`Copy-Item C:\App\config.xml "C:\App\config.xml.bak.$(Get-Date -Format yyyyMMdd)"`
Expected output	None (add `-PassThru` to see the new item)
Equivalent	`cp`, `cp -r`; `robocopy` to preserve ACLs/attributes
Common errors	Nested folder when the destination exists · `Access denied` on read-only items without `-Force`
Use case	Configuration backups.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/copy-item

Move-Item / Rename-Item

Field	Detail
Version / OS	5.1, 7.x · all OS. Aliases: `move`, `mv` (Windows), `mi` / `ren`, `rni`
Description	`Move-Item` moves (and can rename); `Rename-Item` renames in place.

Field	Detail
Syntax	`Move-Item [-Path] <String[]> [[-Destination] <String>] [-Force]` · `Rename-Item [-Path] <String> [-NewName] <String>`
Example	`Get-ChildItem *.log | Rename-Item -NewName { $.BaseName + '-old' + $.Extension }`
Expected output	`app.log` → `app-old.log` (no output by default)
Equivalent	`mv`
Common errors	`Cannot create a file when that file already exists` (`Move-Item` without `-Force`) · `-NewName` given a path
Use case	Bulk renaming with a script block.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/move-item · https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/rename-item

Remove-Item

Field	Detail
Version / OS	5.1, 7.x · all OS. Aliases: `del`, `erase`, `rd`, `rmdir`, `ri`, `rm` (Windows)
Description	Deletes items. **Permanent: no Recycle Bin.**
Syntax	`Remove-Item [-Path] <String[]> [-LiteralPath] [-Recurse] [-Force] [-Filter] [-Include] [-Exclude] [-WhatIf] [-Confirm]`
Example	`Remove-Item C:\Temp\build -Recurse -WhatIf`
Expected output	`What if: Performing the operation "Remove Directory" on target "C:\Temp\build".`
Equivalent	`rm`, `rm -r`, `rmdir`
Common errors	`... has children and the Recurse parameter was not specified` · `being used by another process` · `-rf` is not a parameter
Use case	Previewed cleanup of temporary folders.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/remove-item

Test-Path

Field	Detail
Version / OS	5.1, 7.x · all OS
Description	Returns `$true`/`$false` for whether a path exists (optionally as file or folder).
Syntax	`Test-Path [-Path] <String[]> [-PathType <Any|Container|Leaf>]`
Example	`Test-Path C:\Windows\System32\drivers\etc\hosts -PathType Leaf`

Field	Detail
Expected output	`True`
Equivalent	`[ -e path ]`, `[ -f file ]`, `[ -d dir ]`
Use case	Guard clauses before create/delete operations.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/test-path

Join-Path / Split-Path

Field	Detail
Version / OS	5.1, 7.x · all OS
Description	Build or take apart paths with the correct separator.
Example	`Join-Path $HOME 'lpm-lab'` · `Split-Path C:\a\b\file.txt -Leaf`
Expected output	`C:\Users\student\lpm-lab` · `file.txt`
Equivalent	`"$HOME/lpm-lab"`, `basename`, `dirname`
Use case	Cross-platform scripts.
Docs	https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/join-path

Bash ↔ PowerShell Side-by-Side

Rows are added as each day is released. ⚠ marks behaviour that differs in an important way.

Identity & system

Task	Bash (Ubuntu)	PowerShell	Notes
Current user	`whoami`	`whoami` · `$env:USERNAME`	
Host name	`hostname`	`hostname` · `$env:COMPUTERNAME`	
OS version	`cat /etc/os-release`	`Get-ComputerInfo -Property OsName,OsVersion`	Windows-only cmdlet
Kernel / build	`uname -r`	`[Environment]::OSVersion.Version`	
Architecture	`uname -m`	`$env:PROCESSOR_ARCHITECTURE`	
Shell version	`echo $BASH_VERSION`	`$PSVersionTable.PSVersion`	
Date / UTC	`date` · `date -u`	`Get-Date` · `(Get-Date).ToUniversalTime()`	
ISO timestamp	`date -Iseconds`	`Get-Date -Format o`	

Help

Task	Bash	PowerShell	Notes
Full help	`man cmd`	`Get-Help cmd -Full`	`Update-Help` first
Examples	`man cmd` (EXAMPLES section)	`Get-Help cmd -Examples`	
Find commands	`apropos word`	`Get-Command *word*`	
What runs?	`type -a name`	`Get-Command name -All`	

Navigation & files

Task	Bash	PowerShell	Notes
Where am I	`pwd`	`Get-Location`	
List (incl. hidden)	`ls -la`	`Get-ChildItem -Force`	⚠ `ls -la` fails in PowerShell
Recursive list	`ls -R`	`Get-ChildItem -Recurse`	
Change dir	`cd dir`	`Set-Location dir`	
Previous dir	`cd -`	`Set-Location -`	6.2+ only
Make dirs	`mkdir -p a/b`	`New-Item a\b -ItemType Directory -Force`	
New empty file	`touch f`	`New-Item f -ItemType File`	⚠ `-Force` overwrites

Task	Bash	PowerShell	Notes
Copy	`cp a b`	`Copy-Item a b`	
Copy tree	`cp -r a b`	`Copy-Item a b -Recurse`	Same nesting rule
Move / rename	`mv a b`	`Move-Item a b` · `Rename-Item a -NewName b`	
Delete	`rm f`	`Remove-Item f`	No recycle bin
Delete tree	`rm -r d`	`Remove-Item d -Recurse`	
Force	`rm -rf d`	`Remove-Item d -Recurse -Force`	⚠ `-rf` is not a PS parameter
Simulate	*(preview with ls)*	`-WhatIf`	
Exists?	`[ -e p ]`	`Test-Path p`	
Literal name	`'a[1].txt'`	`-LiteralPath 'a[1].txt'`	