

PowerShell Administrator Reference - 245 Commands

A practical Windows administration reference covering discovery, files, services, CIM, networking, accounts, remoting, event logs, Defender, storage, packages, and automation.

DOCUMENT CENTER

Compiled September 2026

CAUTION: Many administrator cmdlets change services, accounts, firewall rules, storage, registry values, and security settings. Use read-only `Get/Test` commands first, review scope, and use `-WhatIf/-Confirm` where the command supports them.

Use this guide as a practical reference. Verify commands, targets, permissions, paths, versions, and organizational policy before making changes. Examples with placeholders must be adapted to the environment.

Contents

#	Section	Entries
1	Discovery, help, pipeline, and objects	25
2	Files, content, paths, CSV, JSON, and archives	25
3	Processes, services, scheduled tasks, and jobs	25
4	System, hardware, CIM, registry, and environment	25
5	Networking, DNS, routes, sockets, and firewall	25
6	Accounts, groups, permissions, remoting, and sessions	25
7	Event logs, auditing, Defender, and security	25
8	Storage, disks, volumes, updates, packages, and features	25
9	Automation, formatting, filtering, error handling, and scripts	25
10	Additional administrator essentials	20

Total reference entries: 245

1. Discovery, help, pipeline, and objects

Command	Purpose / Notes
Get-Command	List available commands.
Get-Command -Name *Service*	Find commands by name.
Get-Command -Verb Get	List commands using the Get verb.
Get-Help Get-Service	Show help for a cmdlet.
Get-Help Get-Service -Examples	Show examples for a cmdlet.
Get-Help Get-Service -Online	Open online help when supported.
Update-Help	Update locally installed help content; often requires elevation/network access.
Get-Member	Inspect the type and members of pipeline objects.
Get-Alias	List aliases.
Get-Variable	List variables.
Get-PSDrive	List PowerShell drives/providers.
Get-PSProvider	List PowerShell providers.
\$PSVersionTable	Show PowerShell/runtime version information.
Get-Verb	List approved PowerShell verbs.
Get-History	Show command history for the session.
Invoke-History 3	Run a command from session history.
Clear-History	Clear PowerShell session history.
Get-Location	Show the current location.
Set-Location C:\Windows	Change location.
Push-Location C:\Temp	Push a location onto the stack.
Pop-Location	Return to the previous pushed location.
Get-ChildItem	List items in the current location.
Get-ChildItem -Force	Include hidden/system items where applicable.
Select-Object -First 10	Select the first ten pipeline objects.
Sort-Object Name	Sort pipeline objects by a property.

2. Files, content, paths, CSV, JSON, and archives

Command	Purpose / Notes
Get-Item C:\Path\File.txt	Get a specific filesystem item.
Get-ChildItem C:\Logs -Recurse -File	Enumerate files recursively.
Test-Path C:\Path	Test whether a path exists.
Resolve-Path .\file.txt	Resolve a path to its provider path.
New-Item -ItemType Directory -Path C:\Temp\Work	Create a directory.
New-Item -ItemType File -Path .\notes.txt	Create a file.
Copy-Item .\source.txt .\backup.txt	Copy an item.
Copy-Item C:\Data C:\Backup -Recurse	Copy a directory tree.
Move-Item .\old.txt .\new.txt	Move or rename an item.
Rename-Item .\old.txt new.txt	Rename an item.
Remove-Item .\old.txt	Remove a file. Verify the target first.
Remove-Item C:\Temp\Old -Recurse -Confirm	Remove a directory tree with confirmation.
Get-Content .\app.log	Read a text file.
Get-Content .\app.log -Tail 50	Read the last 50 lines.
Get-Content .\app.log -Tail 20 -Wait	Follow appended lines.
Set-Content .\file.txt "text"	Replace file content.
Add-Content .\file.txt "more"	Append content to a file.
Select-String -Path *.*.log -Pattern "error"	Search text files with regex matching.
Get-FileHash .\file.iso -Algorithm SHA256	Calculate a SHA-256 hash.
Import-Csv .\data.csv	Import CSV rows as objects.
Export-Csv .\out.csv -NoTypeInfo	Export objects to CSV.
ConvertTo-Json -Depth 5	Convert pipeline objects to JSON.
ConvertFrom-Json	Parse JSON into PowerShell objects.
Compress-Archive -Path .\Folder -DestinationPath .\backup.zip	Create a ZIP archive.
Expand-Archive .\backup.zip -DestinationPath .\restore	Extract a ZIP archive.

3. Processes, services, scheduled tasks, and jobs

Command	Purpose / Notes
Get-Process	List processes.
Get-Process -Name powershell	Get processes by name.
Get-Process Sort-Object CPU -Descending Select-Object -First 10	Show highest cumulative CPU consumers.
Stop-Process -Id <pid>	Stop a process by PID.
Stop-Process -Name notepad -Confirm	Stop a named process with confirmation.
Start-Process notepad.exe	Start a process.
Start-Process powershell.exe -Verb RunAs	Request an elevated PowerShell process.
Wait-Process -Id <pid>	Wait for a process to exit.
Get-Service	List services.
Get-Service -Name wuauserv	Show Windows Update service status.
Get-Service Where-Object Status -eq Running	List running services.
Start-Service wuauserv	Start a service.
Stop-Service wuauserv	Stop a service.
Restart-Service spooler	Restart a service.
Set-Service wuauserv -StartupType Manual	Set a service startup type. Review dependencies first.
Get-ScheduledTask	List scheduled tasks.
Get-ScheduledTask -TaskName *Update*	Filter scheduled tasks by name.
Get-ScheduledTaskInfo -TaskName <name>	Show task runtime information.
Start-ScheduledTask -TaskName <name>	Run a scheduled task.
Disable-ScheduledTask -TaskName <name>	Disable a task. Confirm business impact.
Enable-ScheduledTask -TaskName <name>	Enable a scheduled task.
Start-Job { Get-Process }	Run a background job.
Get-Job	List background jobs.
Receive-Job -Id 1	Receive job output.
Remove-Job -Id 1	Remove a completed job object.

4. System, hardware, CIM, registry, and environment

Command	Purpose / Notes
Get-ComputerInfo	Get broad computer and Windows information.
Get-CimInstance Win32_OperatingSystem	Show operating-system information.
Get-CimInstance Win32_ComputerSystem	Show computer-system information.
Get-CimInstance Win32_BIOS	Show BIOS/firmware information.
Get-CimInstance Win32_Processor	Show processor information.
Get-CimInstance Win32_PhysicalMemory	Show physical memory modules.
Get-CimInstance Win32_LogicalDisk	Show logical disks.
Get-CimInstance Win32_PnPEntity Where-Object ConfigManagerErrorCode -ne 0	Find PnP devices reporting errors.
Get-CimInstance Win32_QuickFixEngineering	Show hotfix records exposed through WMI/CIM.
Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion	Read registry values.
Get-ChildItem HKLM:\SOFTWARE	Enumerate registry keys.
New-ItemProperty -Path HKCU:\Software\Example -Name Test -Value 1 -PropertyType DWord	Create a registry value. Back up and verify the path.
Set-ItemProperty -Path HKCU:\Software\Example -Name Test -Value 0	Change a registry value.
Remove-ItemProperty -Path HKCU:\Software\Example -Name Test	Remove a registry value.
[Environment]::GetEnvironmentVariables()	Read environment variables.
\$env:PATH	Show the process PATH variable.
\$env:TEMP	Show the temporary directory path.
[Environment]::SetEnvironmentVariable("Name" ,"Value", "User")	Set a user environment variable.
Get-TimeZone	Show the configured time zone.
Set-TimeZone -Id "Eastern Standard Time"	Set the system time zone. Requires appropriate privileges.
Get-Date	Show date and time.
Get-Culture	Show current culture settings.
Get-UICulture	Show current UI culture.
Get-ExecutionPolicy -List	Show execution-policy scopes.
Get-PSRepository	List registered PowerShell repositories.

5. Networking, DNS, routes, sockets, and firewall

Command	Purpose / Notes
Get-NetAdapter	List network adapters.
Get-NetAdapter Where-Object Status -eq Up	List active network adapters.
Get-NetIPConfiguration	Show IP configuration.
Get-NetIPAddress	List IP addresses.
Get-NetRoute	Show routing entries.
Get-NetRoute -DestinationPrefix 0.0.0.0/0	Show IPv4 default routes.
Get-NetNeighbor	Show neighbor/ARP cache entries.
Get-DnsClient	Show DNS client interface configuration.
Get-DnsClientServerAddress	Show configured DNS servers.
Resolve-DnsName example.com	Resolve a DNS name.
Resolve-DnsName example.com -Server 1.1.1.1	Query a specific DNS server.
Clear-DnsClientCache	Clear the Windows DNS client cache.
Test-NetConnection example.com	Test basic connectivity.
Test-NetConnection example.com -Port 443	Test TCP connectivity to a port.
Test-NetConnection example.com -TraceRoute	Run route diagnostics.
Get-NetTCPConnection	List TCP connections.
Get-NetTCPConnection -State Established	List established TCP connections.
Get-NetUDPEndpoint	List UDP endpoints.
Get-NetTCPConnection Sort-Object RemotePort	Sort TCP connections by remote port.
Get-NetIPInterface	Show IP interface configuration.
Get-NetConnectionProfile	Show network profiles.
Set-NetConnectionProfile -InterfaceAlias Ethernet -NetworkCategory Private	Change a network profile category after verifying the network.
Get-NetFirewallProfile	Show firewall profile settings.
Get-NetFirewallRule	List firewall rules.
Get-NetFirewallRule -Enabled True Select-Object -First 25	Show a sample of enabled firewall rules.

6. Accounts, groups, permissions, remoting, and sessions

Command	Purpose / Notes
Get-LocalUser	List local user accounts.
Get-LocalGroup	List local groups.
Get-LocalGroupMember Administrators	List local Administrators group members.
New-LocalUser -Name analyst -NoPassword	Create a local account without a password; normally set a secure credential instead.
Disable-LocalUser -Name olduser	Disable a local user.
Enable-LocalUser -Name analyst	Enable a local user.
Add-LocalGroupMember -Group Administrators -Member analyst	Add an account to local Administrators. Grants broad privileges.
Remove-LocalGroupMember -Group Administrators -Member analyst	Remove an account from local Administrators.
Get-Acl C:\Data	Show an ACL.
Get-Acl C:\Data Format-List	Display ACL details.
\$acl = Get-Acl C:\Data	Load an ACL object for review or editing.
Get-PSSession	List PowerShell remoting sessions.
New-PSSession -ComputerName <host>	Create a remoting session where WinRM is configured.
Enter-PSSession -ComputerName <host>	Enter an interactive remote session.
Exit-PSSession	Leave an interactive remote session.
Invoke-Command -ComputerName <host> -ScriptBlock { hostname }	Run a command remotely where authorized.
Remove-PSSession -Id 1	Close a remoting session.
Test-WSMan <host>	Test WinRM availability.
Get-CimSession	List CIM sessions.
New-CimSession -ComputerName <host>	Create a CIM session.
Remove-CimSession -CimSession \$session	Close a CIM session.
Get-SmbSession	Show SMB server sessions.
Get-SmbOpenFile	Show files opened through the SMB server.
Get-SmbShare	List SMB shares.
Get-SmbShareAccess -Name <share>	Show share permissions.

7. Event logs, auditing, Defender, and security

Command	Purpose / Notes
Get-WinEvent -ListLog *	List event logs.
Get-WinEvent -LogName System -MaxEvents 50	Read recent System events.
Get-WinEvent -LogName Security -MaxEvents 50	Read recent Security events; elevation is often required.
Get-WinEvent -FilterHashtable @{LogName="System";Level=2}	Filter System log errors.
Get-WinEvent -FilterHashtable @{LogName="Security";Id=4625}	Find failed-logon events.
Get-WinEvent -FilterHashtable @{LogName="Security";Id=4688}	Find process-creation events if auditing is enabled.
wevtutil el	List event log names through wevtutil.
Get-MpComputerStatus	Show Microsoft Defender Antivirus health.
Update-MpSignature	Update Defender security intelligence.
Start-MpScan -ScanType QuickScan	Start a Defender quick scan.
Start-MpScan -ScanType FullScan	Start a Defender full scan.
Get-MpThreat	Show Defender threat records.
Get-MpThreatDetection	Show Defender detection details.
Get-MpPreference	Show Defender preferences.
Set-MpPreference -PUAProtection Enabled	Enable potentially unwanted application protection where policy permits.
Get-BitLockerVolume	Show BitLocker volume state.
manage-bde -status	Show BitLocker status using the command-line utility.
Get-Tpm	Show Trusted Platform Module status.
Confirm-SecureBootUEFI	Check Secure Boot state on supported UEFI systems.
Get-AuthenticodeSignature .\script.ps1	Inspect a file signature.
Set-AuthenticodeSignature	Apply a code-signing signature when a suitable certificate is available.
Get-Certificate -Template *	Query certificates with PKI cmdlets where available.
Get-ChildItem Cert:\LocalMachine\My	List local-machine personal certificates.
Get-FileHash .\download.bin -Algorithm SHA256	Calculate a file hash.
Get-NetFirewallRule -Enabled True	List enabled Defender Firewall rules.

8. Storage, disks, volumes, updates, packages, and features

Command	Purpose / Notes
Get-Disk	List physical disks.
Get-PhysicalDisk	List Storage Spaces physical disks.
Get-Partition	List partitions.
Get-Volume	List volumes.
Get-Volume Sort-Object SizeRemaining	Sort volumes by remaining space.
Get-StoragePool	List storage pools.
Get-StorageReliabilityCounter -PhysicalDisk (Get-PhysicalDisk Select-Object -First 1)	Read available storage reliability counters.
Repair-Volume -DriveLetter C -Scan	Scan a volume for filesystem issues.
Optimize-Volume -DriveLetter C -Analyze -Verbose	Analyze volume optimization state.
Optimize-Volume -DriveLetter C -Verbose	Optimize a volume using the appropriate media method.
Get-WindowsOptionalFeature -Online	List optional Windows features.
Get-WindowsOptionalFeature -Online -FeatureName <name>	Show a feature state.
Enable-WindowsOptionalFeature -Online -FeatureName <name> -All	Enable an optional feature. Review dependencies and restart requirements.
Disable-WindowsOptionalFeature -Online -FeatureName <name>	Disable an optional feature. Review dependencies.
Get-WindowsCapability -Online	List Windows capabilities.
Add-WindowsCapability -Online -Name <capability>	Install a Windows capability.
Remove-WindowsCapability -Online -Name <capability>	Remove a capability after verifying impact.
Get-AppxPackage	List AppX/MSIX packages for the current user.
Get-AppxPackage -AllUsers	List AppX/MSIX packages for all users.
Get-AppxPackage Microsoft.WindowsStore Reset-AppxPackage	Reset the Microsoft Store package where supported.
Get-HotFix	List hotfix records exposed by Win32_QuickFixEngineering.
Get-WindowsUpdateLog	Convert Windows Update ETL traces into a readable log.
DISM /Online /Cleanup-Image /ScanHealth	Call DISM from PowerShell to scan component-store health.
DISM /Online /Cleanup-Image /RestoreHealth	Call DISM to repair the component store.
sfc /scannow	Call System File Checker to scan and repair protected system files.

9. Automation, formatting, filtering, error handling, and scripts

Command	Purpose / Notes
\$items Where-Object Status -eq "Running"	Filter objects by a property.
\$items Select-Object Name,Status	Project selected properties.
\$items Group-Object Status	Group objects by a property.
\$items Measure-Object	Count and measure objects.
\$items Format-Table -AutoSize	Format output as a table.
\$items Format-List *	Format all object properties as a list.
\$items Out-File .\report.txt	Write formatted output to a file.
\$items Tee-Object -FilePath .\report.txt	Save pipeline output while passing it onward.
ForEach-Object { \$_.Name }	Process each pipeline object.
1..10 ForEach-Object { \$_ }	Generate and process a numeric range.
if (\$condition) { "yes" } else { "no" }	Conditional execution.
switch (\$value) { "A" { } default { } }	Multi-branch conditional processing.
try { Get-Item C:\Missing -ErrorAction Stop } catch { \$_.Exception.Message }	Handle terminating errors.
\$Error[0]	Inspect the most recent error record.
\$LASTEXITCODE	Read the exit code from the last native executable.
\$?	Show whether the last operation succeeded.
Write-Verbose "detail" -Verbose	Emit verbose diagnostic output.
Write-Warning "warning"	Emit a warning message.
Write-Error "problem"	Write a non-terminating error.
Start-Transcript -Path .\session.txt	Record console session output.
Stop-Transcript	Stop session transcription.
Set-StrictMode -Version Latest	Enable stricter scripting checks in the current scope.
Get-ScriptAnalyzerRule	List PSScriptAnalyzer rules when the module is installed.
Invoke-ScriptAnalyzer .\script.ps1	Analyze a PowerShell script when PSScriptAnalyzer is installed.
powershell.exe -NoProfile -File .\script.ps1	Run a Windows PowerShell script without loading profiles.

10. Additional administrator essentials

Command	Purpose / Notes
Get-Content \$PROFILE	Read the current PowerShell profile if it exists.
Test-Path \$PROFILE	Check whether the current profile exists.
New-Item -ItemType File -Path \$PROFILE -Force	Create the current profile file.
Get-Module	List imported modules.
Get-Module -ListAvailable	List modules available for import.
Import-Module <module>	Import a module.
Remove-Module <module>	Remove a module from the session.
Find-Module <name>	Search a registered repository for a module.
Install-Module <name> -Scope CurrentUser	Install a module for the current user after verifying source and trust.
Save-Module <name> -Path C:\Temp\Modules	Download a module without installing it.
Get-PSDrive -PSProvider FileSystem	List filesystem PowerShell drives.
New-PSDrive -Name X -PSProvider FileSystem -Root \server\share	Map a temporary PowerShell drive.
Remove-PSDrive X	Remove a PowerShell drive mapping.
Get-Clipboard	Read text from the clipboard.
Set-Clipboard "text"	Write text to the clipboard.
Get-Counter "\\Processor(_Total)\% Processor Time"	Sample a performance counter.
Get-Process Export-Clixml .\processes.xml	Serialize PowerShell objects to CLIXML.
Import-Clixml .\processes.xml	Restore serialized objects from CLIXML.
Compare-Object \$a \$b	Compare two collections.
Get-Random -Minimum 1 -Maximum 100	Generate a pseudorandom number for scripting purposes.

Authoritative references

Use vendor and standards documentation for exact syntax, current behavior, version-specific details, licensing, and policy requirements.

Microsoft PowerShell documentation

<https://learn.microsoft.com/powershell/>

Get-Command

<https://learn.microsoft.com/powershell/module/microsoft.powershell.core/get-command>

Get-Help

<https://learn.microsoft.com/powershell/module/microsoft.powershell.core/get-help>

Windows Commands

<https://learn.microsoft.com/windows-server/administration/windows-commands/windows-commands>