Putting Windows PowerShell to Work

WORK WITH WINDOWS POWERSHELL COMMANDS

Jeff Hicks
AUTHOR/TEACHER

@jeffhicks | https://jdhitsolutions.com/blog

How to Follow Along

Windows 10/11 Desktop

Administrator Privileges

Optional

Virtual Machine or Windows Sandbox

Demonstration Files

Download course files

Open ps1 files in a script editor

Run demo commands individually

Discovering Windows PowerShell Commands

Get-Command
(alias gcm)

Get-Help
(alias help)

Command Types

Cmdlet

Function

Script

Application

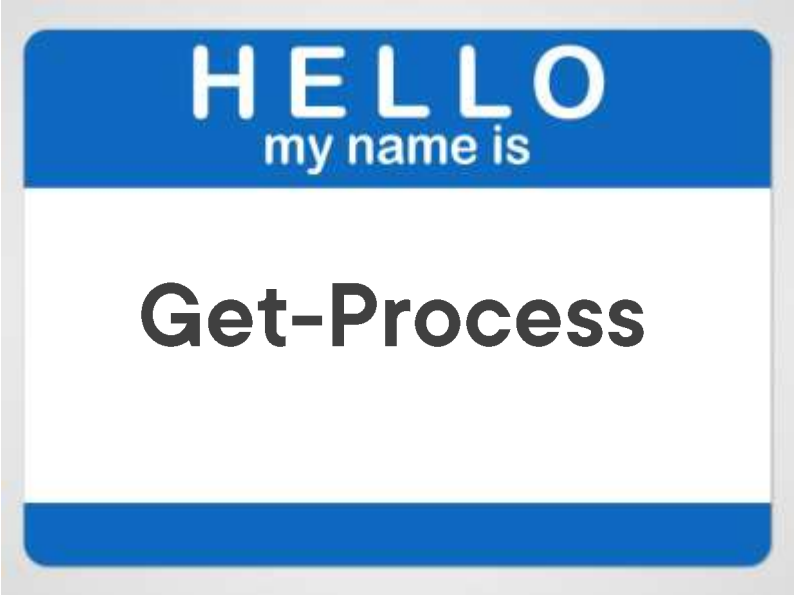
Interactive Console

Windows PowerShell is an interactive shell (console)

You can run any command-line tool

Even other scripting languages

Command Naming Conventions



Applies to PowerShell cmdlets and functions

Verb-Noun

Singular Noun

.NET Standard Verb

Get-Verb

Command Aliases

- Alternate command name
- Technically another command type
- Often shorter
- Used as transition tools
- Get-Alias

```
PS C:\> Get-Volume
```

Running Windows PowerShell Commands

Type the command and press enter

```
PS C:\> Get-Volume

Drive  SizeGB  FreeGB PercentFree HealthStatus
-----  ------  ------ ----------- ------------
              1       0       36.89 Healthy
              0       0        81.9 Healthy
            237     106       44.68 Healthy
C           237     106       44.68 Healthy
              8       8       94.77 Healthy
D           931     512          55 Healthy
```

Running Windows PowerShell Commands

Type the command and press Enter

Use tab completion

```
PS C:\> Get-Volume
```

Using Command Parameters

Customize the command

```
PS C:\> Get-Volume -DriveLetter D
```

Using Command Parameters

Customize the command

Tab completion

PSReadline completion

```
PS C:\> Get-Volume -DriveLetter D

Drive  SizeGB  FreeGB PercentFree HealthStatus
-----  ------  ------ ----------- ------------
D         931     512          55 Healthy
```

Using Command Parameters

Parameter name preceded by a dash

Space

Parameter value

```
PS C:\> Get-Service bits,winrm,spooler
```

Using Command Parameters

Some parameters are positional

Some parameters are mandatory

Separate multiple values with commas

```
PS C:\> Get-CimInstance -ClassName win32_bios -Verbose
```

Using Command Parameters

-Verbose

```
PS C:\> Get-CimInstance -ClassName win32_bios -Verbose
VERBOSE: Perform operation 'Enumerate CimInstances' with following
parameters, ''namespaceName' = root\cimv2,'className' = win32_bios'.

SMBIOSBIOSVersion : M2WKT40A
Manufacturer      : LENOVO
Name              : M2WKT40A
SerialNumber      : MJ0D9JCA
Version           : LENOVO - 1280

VERBOSE: Operation 'Enumerate CimInstances' complete.
```

Using Command Parameters

-Verbose

Must be built-in to the command

```
PS C:\> Stop-Service -Name WinRM -WhatIf
```

Using Command Parameters

-WhatIf

```
PS C:\> Stop-Service -Name WinRM -WhatIf

What if: Performing the operation "Stop-Service" on target "Windows
Remote Management (WS-Management) (WinRM)".
```

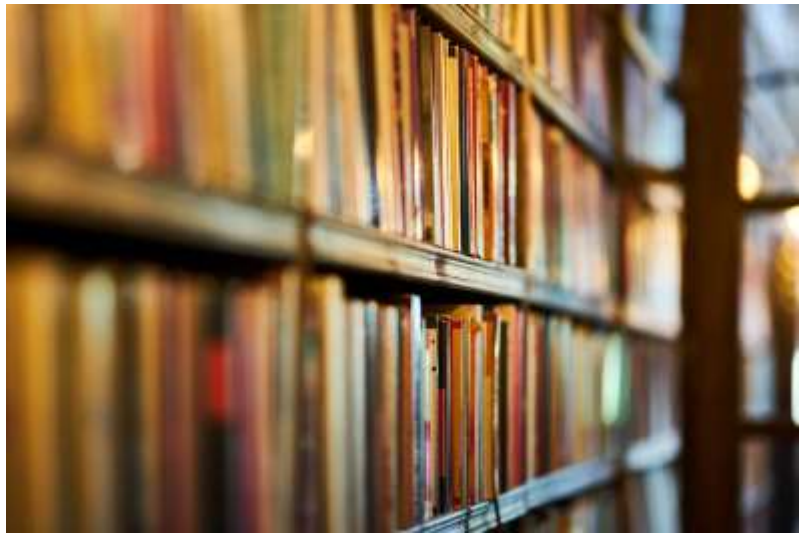
Using Command Parameters

-WhatIf

Shows what PowerShell would have done

***Should* apply to commands that change things**

Windows PowerShell Command History



Scroll through command history

Get-History

Invoke-History

Search command history with PSReadline

- InlinePrediction

Demo

Discovering Windows PowerShell Commands

Windows PowerShell Security

- You can only run what you have permissions and rights to run
- Some PowerShell actions require an elevated console
- Some commands support alternate credentials
- PowerShell scripting security is a separate thing

```
PS C:\> Get-WinEvent -FilterHashtable @{Logname="Windows PowerShell";ID=400} -MaxEvents
1 | Select-Object -expandProperty message
```

Activity Logging

Windows PowerShell activity is logged

- Windows PowerShell
- Microsoft-Windows-PowerShell/Operational

Run Show-EventLog to launch the GUI

```
Engine state is changed from None to Available.

Details:
        NewEngineState=Available
        PreviousEngineState=None

        SequenceNumber=13

        HostName=ConsoleHost
        HostVersion=5.1.19041.1237
        HostId=0486cd64-fe54-42ac-9969-26c279cf9e5c
        HostApplication=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
        EngineVersion=5.1.19041.1237
        RunspaceId=2dbb3cc0-8a43-4f68-b51f-8684765137fd
        PipelineId=
        CommandName=
        CommandType=
        ScriptName=
        CommandPath=
        CommandLine=
```

Enable Additional Logging

HKLM:\SOFTWARE\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging

HKLM:\SOFTWARE\WOW6432Node\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging

EnableScriptBlockLogging DWORD 1

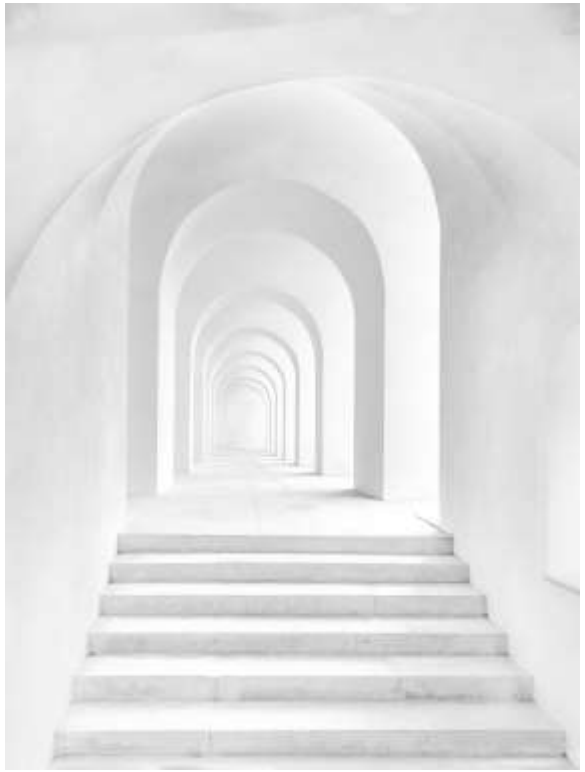
https://devblogs.microsoft.com/powershell/powershell-the-blue-team/

Learn More

Demo

Windows PowerShell Security

Understanding the PowerShell Host



Windows PowerShell alone is an *engine*

Windows PowerShell is *hosted* in an application

Windows PowerShell commands write objects to the *pipeline*

You can read and write directly to the hosting application

Pipeline Streams

PowerShell has multiple pipeline streams

- Error
- Verbose
- Warning

Identified in command output

You can redirect them to a file

Demo

The Windows PowerShell Host

- Commands in the Pipeline
- Using Write-Host
- Using Read-Host
- Pipeline Streams