



# PowerShell & PowerCLI Starting from Scratch

INDIANAPOLIS VMUG UserCon

JUNE 29, 2022

# Get-Mike

 mikenelsonio

 nelmedia

 mikenelson-io

- ▶ 35+ years in tech
- ▶ Principal Technologist
- ▶ Experience from Helpdesk to Architect
- ▶ Scripter, not a coder
- ▶ Passion for community, teaching, learning
- ▶ Beer, BBQ, & Gadgets



# Get-Content

- ▶ PowerShell Basics
- ▶ PowerCLI
- ▶ Demos

## PowerShell (PoSH)

Started as a scripting framework for automation & evolved into a command line interface (CLI) and a scripting language

Native executables, cmdlets, scripts, functions, aliases, modules, help, profiles, parameters, and more

# Get-History

- Jeffrey Snover, Bruce Payette, & James Truher
- Project Monad in 2002
- The “Monad Manifesto”
- Snover was demoted after



 SIGN IN

The  Register®

{\* OFFBEAT \*}

## Jeffrey Snover claims Microsoft demoted him for inventing PowerShell

Twitter thread reveals Redmondian machinations

## Get-SnoverQuips

“This is rock science, not rocket science.”

---

“PowerShell is a such a great product because I am a deeply flawed human being.”

---

“When in trouble, fear, or doubt - run in circles, scream, and shout.”

# Get-Versions

.Net Framework



$\leq 5.1$

Windows Server  
Windows Client

.NET Core



6.x



7.x

Windows Client  
Windows Server  
Linux  
MacOS

# Start-Install

- ▶ Pre-installed - powershell.exe or pwsh.exe
- ▶ Microsoft Store - Get Windows Terminal
- ▶ PowerShell GitHub Repository - [aka.ms/powershell](https://aka.ms/powershell)
- ▶ Windows Update (7.2.1+)
- ▶ Chocolatey
- ▶ Snap (\*nix)

```
PowerShell 7.2.5
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Windows\System32>
```

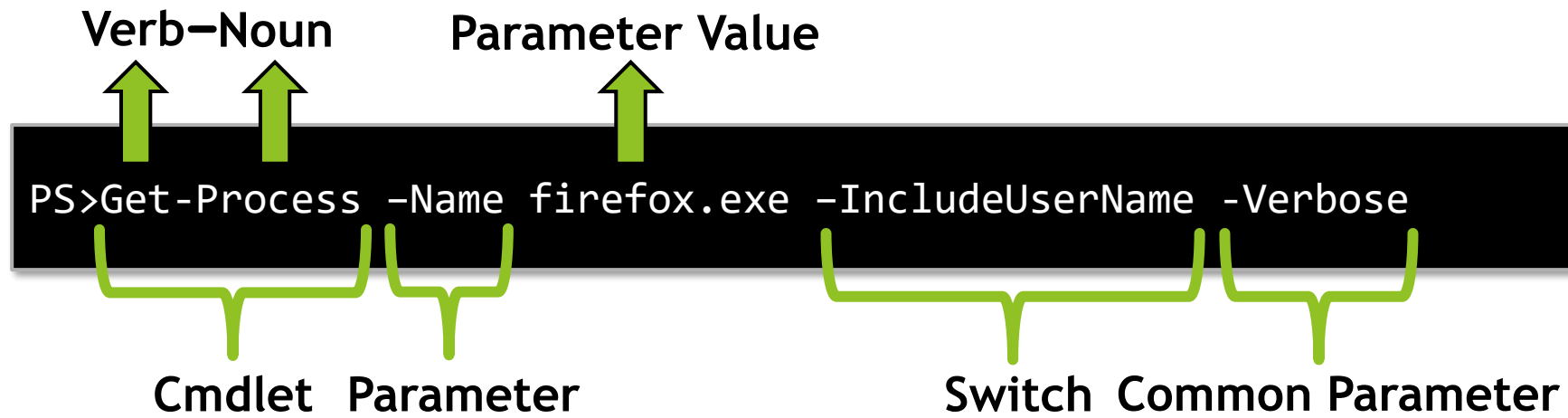


# Cmdlets

“commandlets”

- ▶ A type of command in PowerShell
- ▶ Common syntax & options
- ▶ Usually take object input & return objects
- ▶ Repeatable and auditable
- ▶ Stored in **.ps1** file for scripts

# Get-Syntax



# Parameters

Named

Positional

Dynamic

Common

Sets

- ▶ Allow for users to provide input or options
- ▶ Used by cmdlets, functions, and scripts
- ▶ A hyphen (“-”) signals PoSH to a parameter
- ▶ Some parameters do not accept or require a value
- ▶ Parameter Sets

# Pipelines

Pipeline operator



```
PS>Get-Process -Name firefox.exe -IncludeUserName | Stop-Process
```

Object returned by first cmdlet sent to second cmdlet

“One-liner”

# Functions

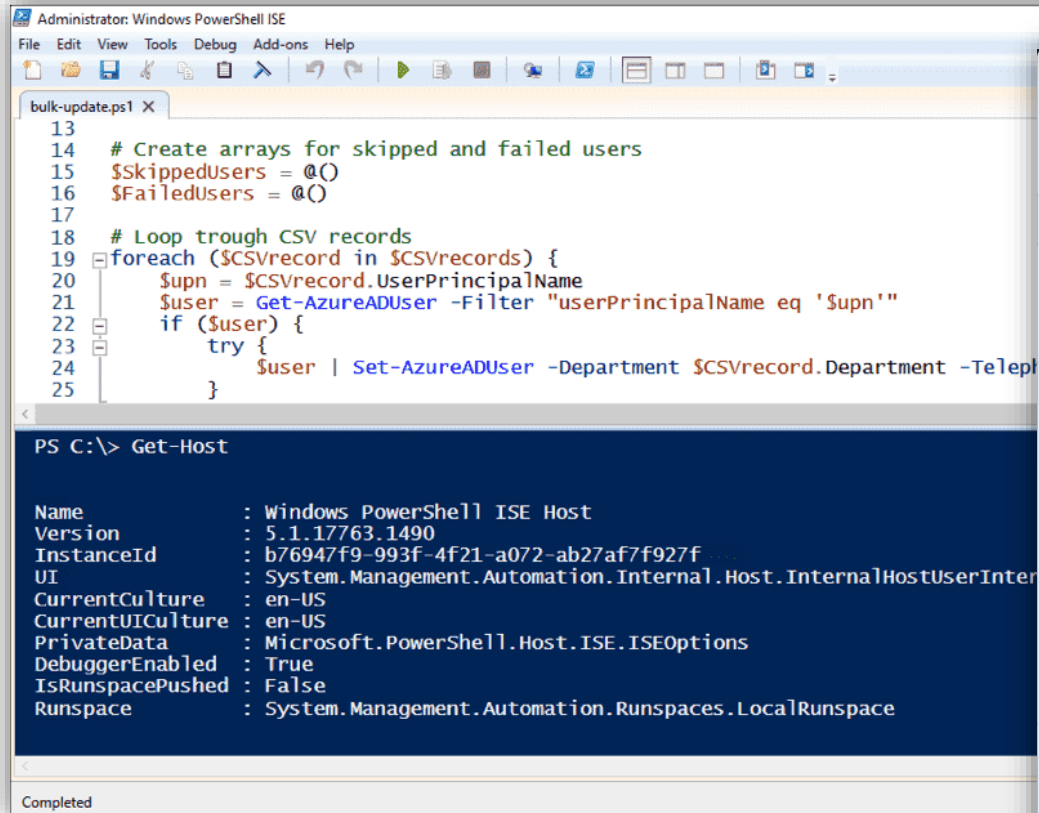
A list of PowerShell statements that run like you had entered them on the command line

```
function Get-FirefoxProcess { Get-Process firefox.exe }  
function Get-FirefoxProcess {  
    $a = Get-Process firefox.exe  
    if ($a -eq $null) {  
        Write-Host "No Firefox process present"}  
    return $a  
}
```

# Modules

- **.psm1** file
  - Imported as itself or launched from **.psd1** file (a module manifest file)
  - Commands, providers, variables, functions, help context, aliases, workflows
- Auto-Loading capability
- Stored in \$env:PSModulePath
- Easy lifecycle management with PowerShell Gallery (Install-Module & Update-Module)

# Creating / Editing / Running



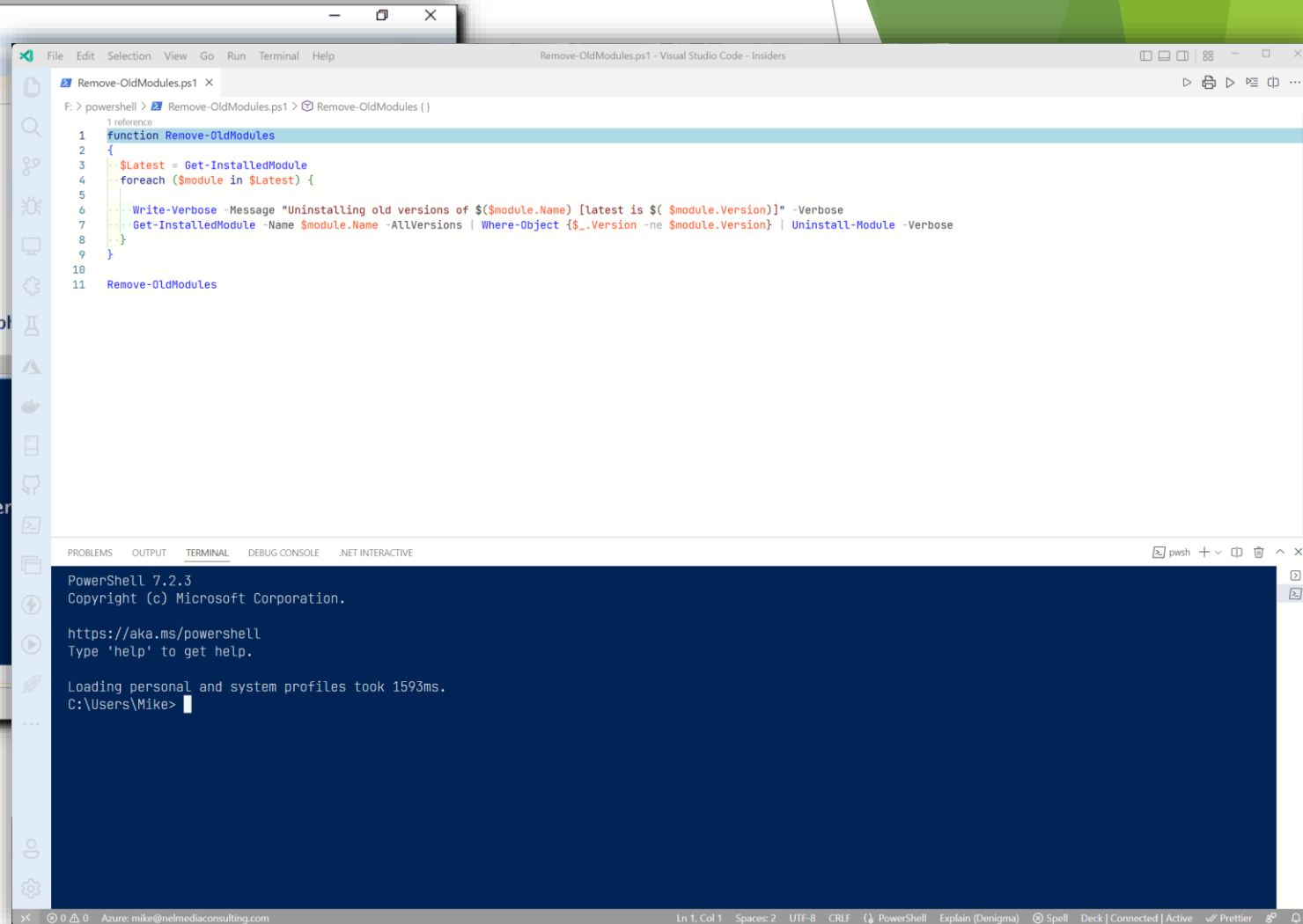
Administrator: Windows PowerShell ISE

```
bulk-update.ps1 X
13
14 # Create arrays for skipped and failed users
15 $SkippedUsers = @()
16 $FailedUsers = @()
17
18 # Loop through CSV records
19 foreach ($CSVrecord in $CSVrecords) {
20     $supn = $CSVrecord.UserPrincipalName
21     $user = Get-AzureADUser -Filter "userPrincipalName eq '$supn'"
22     if ($user) {
23         try {
24             $user | Set-AzureADUser -Department $CSVrecord.Department -Teleph
25         }
26     }
27 }
```

PS C:\> Get-Host

```
Name           : Windows PowerShell ISE Host
Version        : 5.1.17763.1490
InstanceId     : b76947f9-993f-4f21-a072-ab27af7f927f
UI             : System.Management.Automation.Internal.Host.InternalHostUserInter
CurrentCulture : en-US
CurrentUICulture : en-US
PrivateData    : Microsoft.PowerShell.Host.ISE.ISEOptions
DebuggerEnabled : True
IsRunspacePushed : False
Runspace       : System.Management.Automation.Runspaces.LocalRunspace
```

Completed



Remove-OldModules.ps1 - Visual Studio Code - Insiders

```
Remove-OldModules.ps1 X
F: > powershell > Remove-OldModules.ps1 > Remove-OldModules {}

1 reference
1 function Remove-OldModules
2 {
3     $Latest = Get-InstalledModule
4     foreach ($module in $Latest) {
5
6         Write-Verbose -Message "Uninstalling old versions of $($module.Name) [latest is $($module.Version)]" -Verbose
7         Get-InstalledModule -Name $module.Name -AllVersions | Where-Object {$_.Version -ne $module.Version} | Uninstall-Module -Verbose
8     }
9 }
10
11 Remove-OldModules
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE .NET INTERACTIVE

```
PowerShell 7.2.3
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

Loading personal and system profiles took 1593ms.
C:\Users\Mike>
```

Azure: mike@nelmediaconsulting.com

Ln 1, Col 1 Spaces: 2 UTF-8 CR/LF PowerShell Explain (Denigma) Spell Deck | Connected | Active Prettier

## Core Commands to Know

**Get-Command**  
**Get-Help**  
**Update-Help**  
**Update-Module**  
**Get-Member**  
**Show-Command**



# PowerCLI

- Install-Module VMware.PowerCLI
  - -Allow-Clobber, -Force & -SkipPublisherCheck may be necessary

```
Install-Package: The following commands are already available on this system: 'Export-VM, Get-VM, Get-VMHost, Move-VM, New-VM, Remove-VM, Restart-VM, Set-VM, Set-VMHost, Start-VM, Stop-VM, Suspend-VM'. This module 'VMware.VimAutomation.Core' may override the existing commands. If you still want to install this module 'VMware.VimAutomation.Core', use -AllowClobber parameter.
```

- If certificate error, Update-Module will not work
- Cmdlet collisions such as Get-VM
- 800+ cmdlets in 15+ modules
- Set-PowerCLIConfiguration creates PowerCLI\_Settings.xml file
  - For Connect-VIServer certificate errors, use -InvalidCertificateAction Ignore

```
# Modules that must be imported into the global environment prior to importing this module
RequiredModules = @(
@{"ModuleName"="VMware.VimAutomation.Sdk";"ModuleVersion"="12.5.0.19093564"}
@{"ModuleName"="VMware.VimAutomation.Common";"ModuleVersion"="12.6.0.19600917"}
@{"ModuleName"="VMware.Vim";"ModuleVersion"="7.0.3.19601056"}
@{"ModuleName"="VMware.VimAutomation.Core";"ModuleVersion"="12.6.0.19601570"}
@{"ModuleName"="VMware.VimAutomation.Srm";"ModuleVersion"="12.6.0.19609133"}
@{"ModuleName"="VMware.VimAutomation.License";"ModuleVersion"="12.0.0.15939670"}
@{"ModuleName"="VMware.VimAutomation.Vds";"ModuleVersion"="12.5.0.19167830"}
@{"ModuleName"="VMware.CloudServices";"ModuleVersion"="12.6.0.19606210"}
@{"ModuleName"="VMware.VimAutomation.Vmc";"ModuleVersion"="12.6.0.19609014"}
@{"ModuleName"="VMware.VimAutomation.Nsxt";"ModuleVersion"="12.5.0.19168180"}
@{"ModuleName"="VMware.VimAutomation.vROps";"ModuleVersion"="12.5.0.19167825"}
@{"ModuleName"="VMware.VimAutomation.Cis.Core";"ModuleVersion"="12.6.0.19601368"}
@{"ModuleName"="VMware.VimAutomation.HorizonView";"ModuleVersion"="12.5.0.19033914"}
@{"ModuleName"="VMware.VimAutomation.Cloud";"ModuleVersion"="12.0.0.15940183"}
@{"ModuleName"="VMware.DeployAutomation";"ModuleVersion"="7.0.3.19599828"}
@{"ModuleName"="VMware.ImageBuilder";"ModuleVersion"="7.0.3.19599828"}
@{"ModuleName"="VMware.VimAutomation.Storage";"ModuleVersion"="12.6.0.19609013"}
@{"ModuleName"="VMware.VimAutomation.StorageUtility";"ModuleVersion"="1.6.0.0"}
@{"ModuleName"="VMware.VumAutomation";"ModuleVersion"="12.1.0.16941488"}
@{"ModuleName"="VMware.VimAutomation.Security";"ModuleVersion"="12.3.0.17833870"}
@{"ModuleName"="VMware.VimAutomation.Hcx";"ModuleVersion"="12.6.0.19606303"}
@{"ModuleName"="VMware.VimAutomation.WorkloadManagement";"ModuleVersion"="12.4.0.18627055"}
@{"ModuleName"="VMware.Sdk.Runtime";"ModuleVersion"="1.0.106.18628394"}
@{"ModuleName"="VMware.Sdk.vSphere";"ModuleVersion"="1.0.104.18678708"}
@{"ModuleName"="VMware.PowerCLI.VCenter";"ModuleVersion"="12.6.0.19600125"}
@{"ModuleName"="VMware.Sdk.Nsx.Policy";"ModuleVersion"="3.2.0.19610335"}
)
```

https://developer.vmware.com/docs/powercli/latest/products/

vmw VMware Developer Documentation BETA

API Reference

PowerCLI Reference

[← Back Home](#)

VMware PowerCLI Cmdlets

By Products ▾

VMware vSphere and vSAN

VMware Cloud Director

vRealize Operations Manager

VMware Cloud Services

VMware Cloud on AWS

VMware HCX


VMware Horizon

VMware NSX-T Data Center

VMware Site Recovery Manager


Search PowerCLI

VMware PowerCLI Cmdlets by Product




VMware vSphere and vSAN

Provides cmdlets for automated administration of the vSphere environment.




VMware Cloud Director

Provides cmdlets for automating vCloud Director features.




vRealize Operations Manager

Provides cmdlets for automating vRealize Operations Manager features.




VMware Cloud Services

Provides cmdlets for managing VMware Cloud Services.




VMware Cloud on AWS

Provides cmdlets for managing VMware Cloud on AWS features.




VMware HCX

Provides cmdlets for managing VMware HCX features.




VMware Horizon

Provides cmdlets for automating VMware Horizon features.



VMware NSX-T Data Center

Provides cmdlets for managing NSX-T servers.



VMware Site Recovery Manager

Provides cmdlets for managing VMware Site Recovery Manager features.



# Demos

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

# Thank you!

@mikenelsonio