Powershell:

Introduction and Practical Uses

Presentation URL: http://bit.ly/2ICK4Pt

HELLO!

I am **Chris Wieringa**

CS Lab Manager for Calvin College

cwieri39@calvin.edu

1.
Goals

What we will cover today...

Goals

- Define and startup PowerShell (PS)
- Basics of PS
- Local system management
- Remote Powershell
- Software installations
- Windows Update

The *real* goal is to learn something together today.

2.
Define and Startup Powershell

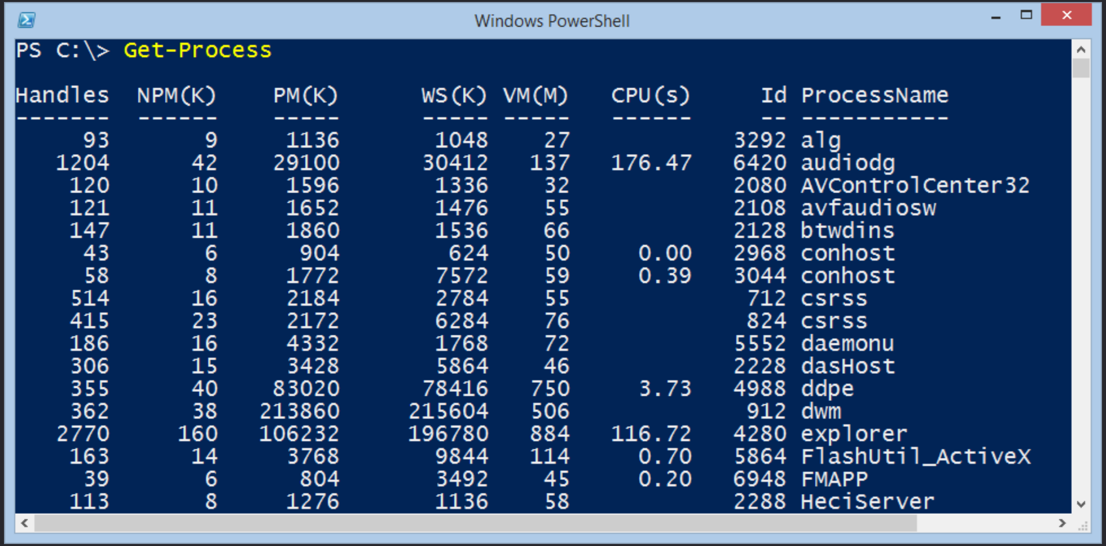
What is Powershell, and how to run it...

Windows PowerShell is an interactive object-oriented command environment with scripting language features that utilizes small programs called cmdlets to simplify configuration, administration, and management of heterogeneous environments in both standalone and networked typologies by utilizing standards-based remoting protocols.

- *Ed Wilson, "The Scripting Guy"*

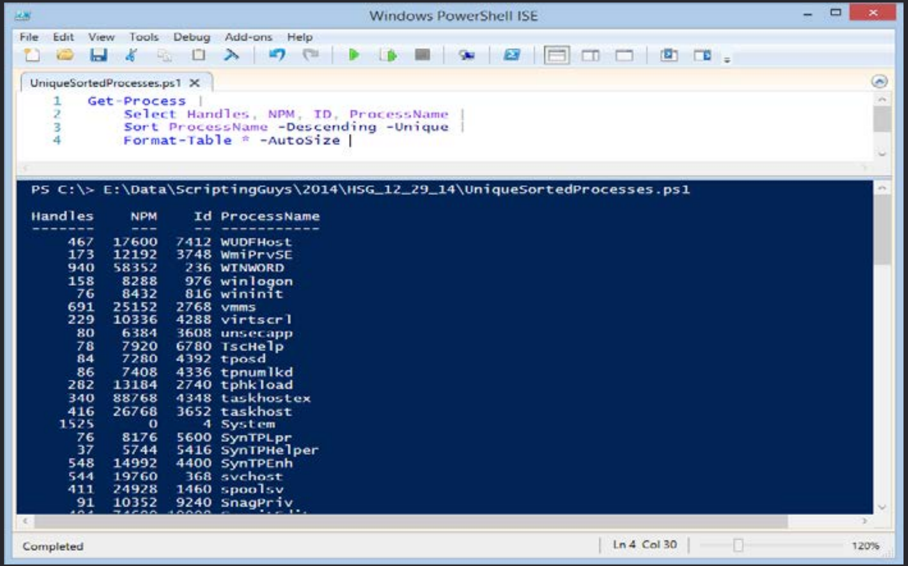
Define Powershell

- Interactive



Define Powershell

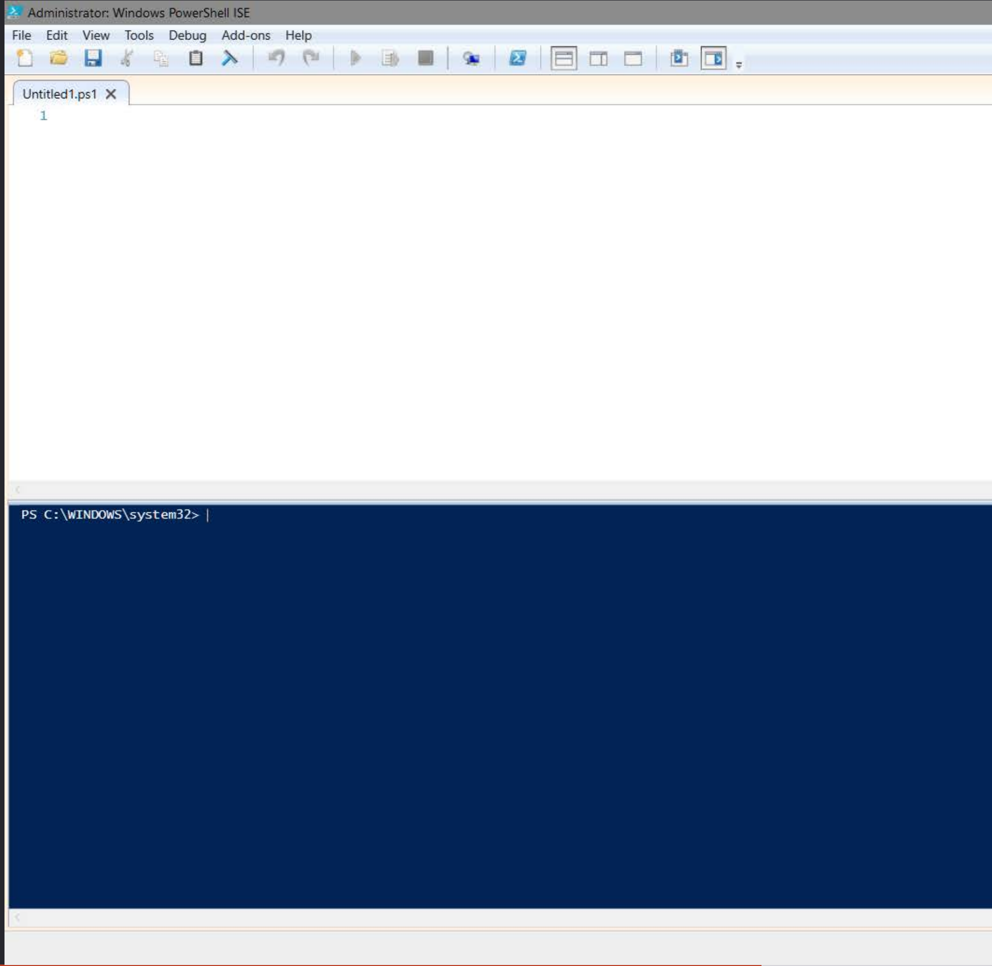
- Scripts



Startup

Today we'll be working with the Powershell ISE :

Start -> Windows Powershell ISE -> Right-click-> Run as Administrator



Security through Execution Policy

- Execution Policy
  - Default security settings to not run scripts.
  - http://bit.ly/2JRSYJN
  - Get-ExecutionPolicy ← if restricted, run:
    - Set-ExecutionPolicy unrestricted
    - (if non-admin, Set-ExecutionPolicy -Scope CurrentUser unrestricted)

3.
Basics

Start with the basics...

Basic Cmdlets

- View processes:
  - Get-Process
- Directory listing:
  - Get-ChildItem -Path C:\
  - Aliased to ‘ls’
- Services list:
  - Get-Service

Basic Cmdlets - More Information

- Aliases of popular cmdlets are available:
  - http://bit.ly/2H7iBV5
- Most cmdlets have arguments you can pass to them
  - -*ArgumentName value*
- Get-Help *cmdlet*
  - Get-Help *cmdlet* -examples
- Everything is treated as an objects, which allows objects to be passed to each other through...

Pipeline

Piping in Powershell is significant. PS will pass objects between commands virtually everywhere.

http://bit.ly/2vjH5Jx

Basic Cmdlets - Pipeline

- Services list - Pipelining!
  - Get-Service | Format-List *(or Get-Service | FL)*
  - Get-Service | Select-Object Name,DisplayName
  - Get-Service | Select-Object Name,DisplayName | Export-CSV -Path “$env:temp\servicelist.csv”
- In File Explorer, check your %TEMP% directory for a “servicelist.csv” file.

Basic Cmdlets - Pipeline example continued...

- Import-CSV -Path “$env:temp\servicelist.csv” |
  Foreach-Object {
    Get-Service -Name $_.Name |
      Where-Object { $_.Status -eq “Running” } |
      Select-Object Name,Status
  }
- $_ is a special variable, representing the current object in the pipeline. “Fill in the blank” variable.

4.
System Management and Administration

How to do something useful on your system...

- Event Viewer / Log
  - Get-EventLog -LogName *System* -Source “*GroupPolicy*”
  - Useful filters:
    - -EntryType { Error | Information | FailureAudit | SuccessAudit | Warning }
    - -LogName: { Application, Security, Setup, System, … }
    - -After or -Before
  - Get-Help Get-Event -examples

System Management and Administration - Start / Stop Processes

- Get-Process - shows current processes
- Start-Process - starts a new process
  - Start-Process -FilePath “notepad” -WindowStyle Maximized
  - Add “-Wait” to wait until the process ends.
- Stop-Process - stops/ends a process
  - Stop-Process -Name notepad
  - Add “-Force” to force kill it

- Get-Acl -Path “C:\Windows” | FL
  - $myGoodAcl = Get-Acl -Path “C:\Fish.txt”
- Set-Acl -Path “C:\Bird.txt” -AclObject $myGoodAcl

- Get-Acl -Path “C:\Fish.txt” | Set-Acl -Path “C:\Bird.txt”

Registry modification potential- Danger Will Robinson!

- Get-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\CurrentVersion"
- Get-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\CurrentVersion" -Name "ProgramFilesDir"

- New-Item -Path "HKCU:\Software\Labman"
- New-ItemProperty -Path "HKCU:\Software\Labman" -Name "NoAttendees" -Value 0
- Set-ItemProperty -Path "HKCU:\Software\Labman" -Name "NoAttendees" -Value 40
- Get-ItemProperty -Path "HKCU:\Software\Labman" -Name "NoAttendees"
- Remove-Item "HKCU:\Software\Labman" -Force -Recurse

Same commands for files!

Restart the computer:

- Restart-Computer
- Restart-Computer -ComputerName “labmachine01” -Wait -For PowerShell -Timeout 300 -Delay 2

Stop / Shutdown the computer:

- Stop-Computer

5.
Remote Powershell

Now do it all on remote machines.....

PS Remoting - Initial Setup

- Powershell can be used in both interactive and scripted modes. Cmdlets may also work on remote computers.
- Requires setup…https://tek.io/2wH6Zr9
  - Windows Remote Management- On/Allowed IPs
  - Windows Remote Management Service Started
  - Firewall settings
  - Fairly in depth… I'll leave this up to you.

Interactive Mode:

- Enter-PSSession -ComputerName “syslab01”
- [syslab01] PS C:\ > Get-ChildItem
- [syslab01] PS C:\ > Exit

PS Remoting - PSSession Remote Scripting

```
$s = New-PSSession -ComputerName “syslab01”
Invoke-Command -Session $s -ScriptBlock {
  Get-Process | Export-CSV -Path “C:\processlist.txt”
}
Copy-Item -FromSession $s -Path “C:\processlist.txt” -Destination ‘C:\reports\’
```

PS Remoting - PSSession Multiple Hosts Scripting

```
# syslab.txt is a file with a list of computer names, one per line
$s = New-PSSession -ComputerName (Get-Content syslab.txt)
Invoke-Command -Session $s -ScriptBlock {
  Get-ComputerInfo
}
```

PS Remoting - Cmdlet “ComputerName”

```
# syslab.txt is a file with a list of computer names, one per line
Invoke-Command -ComputerName (Get-Content syslab.txt)
    -FilePath C:\Scripts\myScript.ps1 -ArgumentList Process, Service
```

https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/invoke-command?view=powershell-6

5.
Software Installations

Get away from the batch scripts...

Software Installations - Get Rid of Batch

- Powershell is a natural replacement for your batch install scripts
- Benefits:
  - Full control over install window
  - Easy to check for existing software installs
  - Native ability to remote copy installer and start install
  - Built-In Provider Packages (aka Chocolatey)

Software Installations - Run those installers...

Basic installation of a file: http://bit.ly/2IDfCVi

- Start-Process -NoNewWindow -Wait -FilePath “C:\installers\7zip_x64.exe” -ArgumentWindow ‘/S’

Got a not-so-silent installer? Control your window…

- Start-Process -WindowStyle {Normal|Hidden|Minimized|Maximized}...

How to check for existing software installs: http://bit.ly/2rWfxFJ

1. (Bad/slow way) Get-WmiObject -Class Win32_Product
2. (Better) Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall* | Select-Object DisplayName, DisplayVersion, Publisher, InstallDate | Format-Table -AutoSize

Detect if the install is needed, only continue if it is old/non-existent...

Software Installations - Chocolatey integration

Use chocolatey? Easy Powershell integration: http://bit.ly/2J0tsol

- Find-Package -ProviderName Chocolatey -Name 7zip | Install-Package

6.
Windows Update

How to run your Windows Update...

Windows Update

- PSWindowsUpdate module from PowerShell Gallery
  - https://www.powershellgallery.com
- Install via the nuget system
- Install once, then load on use

Windows Update - PSWindowsUpdate install

- Install-PackageProvider NuGet -Force
- Import-PackageProvider NuGet -Force
- Set-PSRepository -Name PSGallery -InstallationPolicy Trusted
- Install-Module PSWindowsUpdate

Windows Update - PSWindowsUpdate Operations

- Get-Command -module PSWindowsUpdate
- Get updates:
  - Get-WUInstall -AcceptAll -AutoReboot
- List updates:
  - Get-WUList
- Results from last run:
  - Get-WULastResults

*The cmdlets may take a bit to run.

THANKS!

ANY QUESTIONS?

- cwieri39@calvin.edu
- 616-526-6785
- https://cs.calvin.edu

Presentation URL: http://bit.ly/2ICK4Pt

CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash