# GET-CLI | OUT-POWERSHELL

TURNING CLI TOOLS INTO POWERSHELL TOOLS

JEFF HICKS

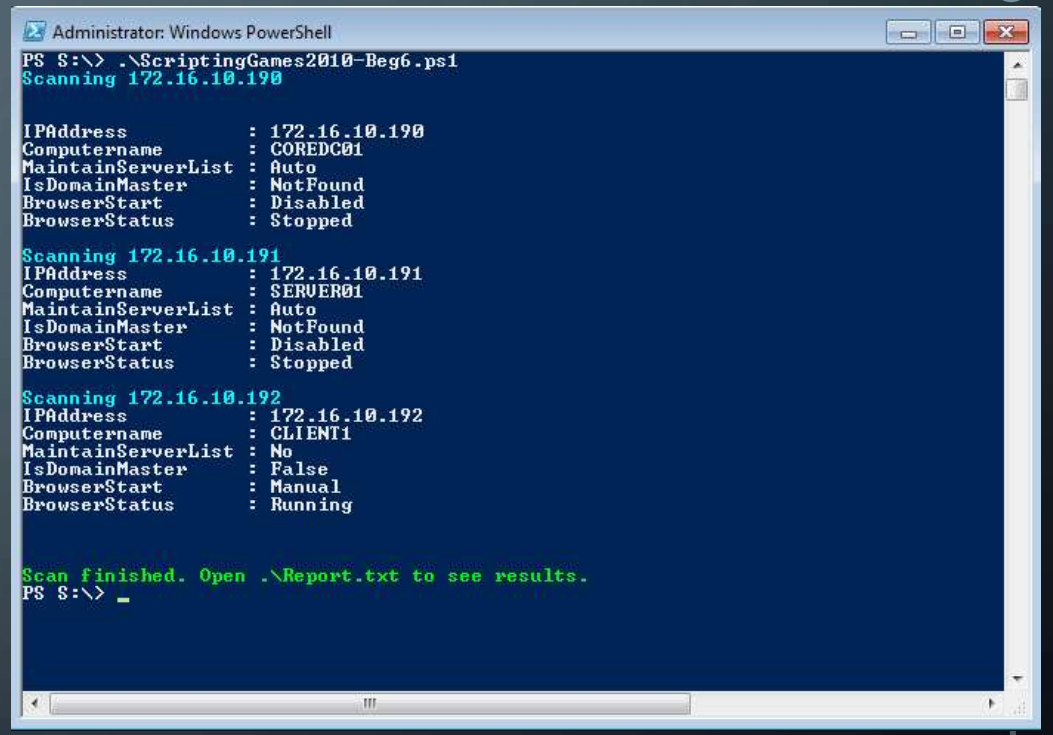
@JEFFHICKS

# THE PROBLEM

- PowerShell is all about the objects
- Leveraging the pipeline is critical to maximum efficiency
- PowerShell solutions are still arriving
- …legacy command line tools fill the gap
- ….how can we turn them into PowerShell tools?

# THE PROBLEM

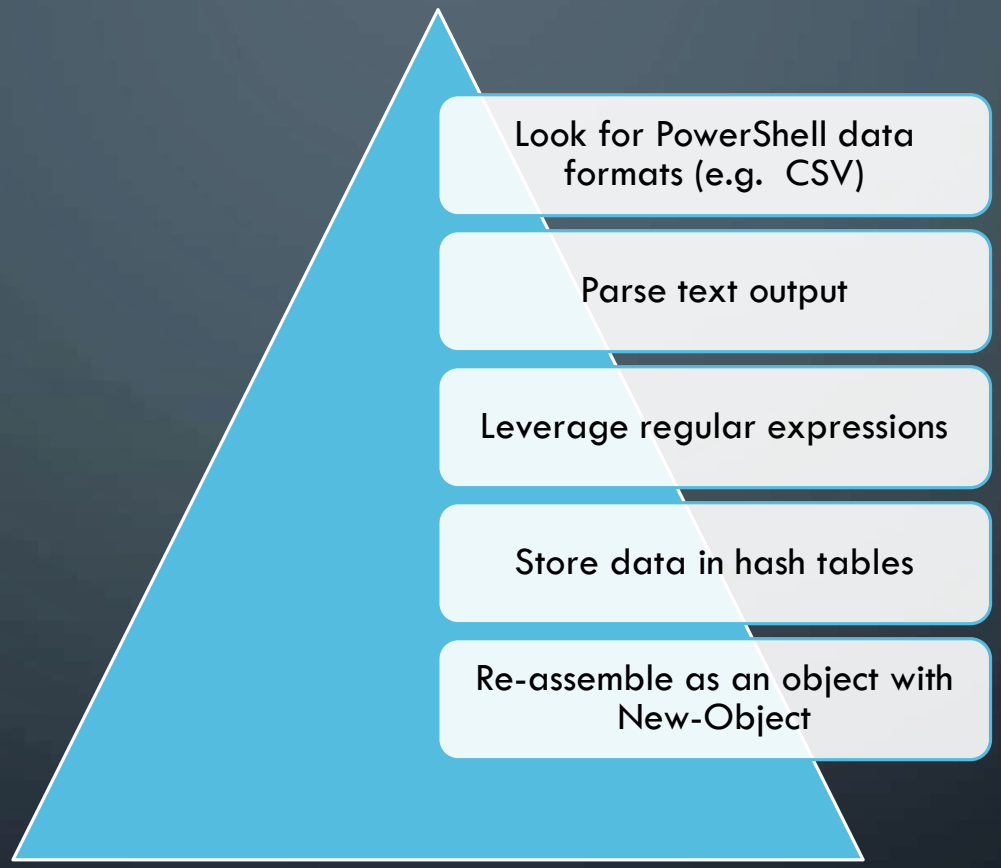

C:\>ipconfig/all
Windows NT IP Configuration
Host Name . . . . . : somehost.somedomain.net
DNS Servers . . . . . : 24.128.1.80
24.128.1.81
Node Type . . . . . : Broadcast
NetBIOS Scope ID. . . . . :
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
NetBIOS Resolution Uses DNS : No
Ethernet adapter Elnk32:
Description . . . . . : ELNK3 Ethernet Adapter.
Physical Address. . . . . : 00-A0-24-D9-DF-46
DHCP Enabled. . . . . : Yes
IP Address. . . . . : 24.128.1.59
Subnet Mask . . . . . : 255.255.255.224
Default Gateway . . . . . : 24.128.1.33
DHCP Server . . . . . : 24.128.1.34
Lease Obtained. . . . . : Tuesday, September 23, 1997 10:14:37 AM
Lease Expires . . . . . : Thursday, October 23, 1997 10:14:37 AM
C:\>


Administrator: Windows PowerShell
PS S:\> .\ScriptingGames2010-Beg6.ps1
Scanning 172.16.10.190
IPAddress : 172.16.10.190
Computername : COREDC01
MaintainServerList : Auto
IsDomainMaster : NotFound
BrowserStart : Disabled
BrowserStatus : Stopped
Scanning 172.16.10.191
IPAddress : 172.16.10.191
Computername : SERVER01
MaintainServerList : Auto
IsDomainMaster : NotFound
BrowserStart : Disabled
BrowserStatus : Stopped
Scanning 172.16.10.192
IPAddress : 172.16.10.192
Computername : CLIENT1
MaintainServerList : No
IsDomainMaster : False
BrowserStart : Manual
BrowserStatus : Running
Scan finished. Open .\Report.txt to see results.
PS S:\>

# REQUIREMENTS

- CLI tool should use StdOut
- CLI tool should offer predictable or formatted output
- Avoid interactive tools like netsh
- …unless they offer a “scripted” solution
- These techniques are better suited for scripting

# TECHNIQUES


Look for PowerShell data formats (e.g. CSV)
Parse text output
Leverage regular expressions
Store data in hash tables
Re-assemble as an object with New-Object

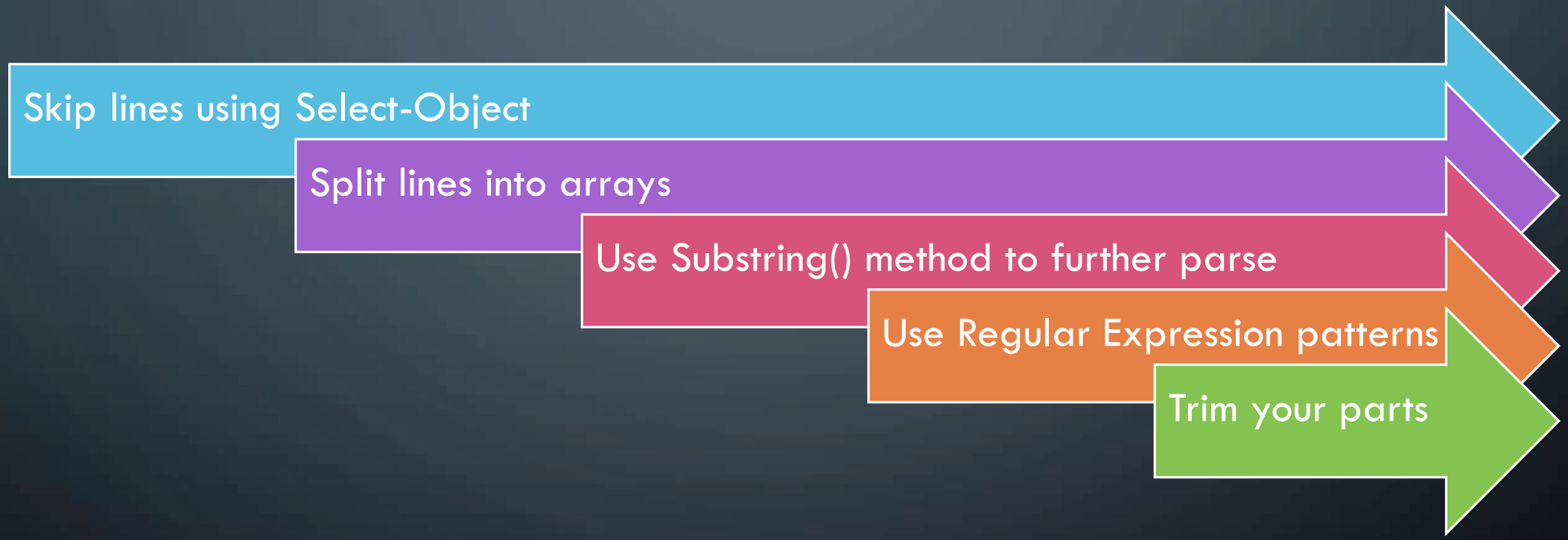
# LOOK FOR POWERSHELL DATA FORMATS

- Does the CLI command offer data formats?
- Can you save formatted results to a file?
- Some output you can turn directly into objects

```
PS C:\> driverquery /fo csv | convertFrom-csv
```

- Be careful creating property names with spaces
- Use your own header

# PARSE TEXT


Skip lines using Select-Object
Split lines into arrays
Use Substring() method to further parse
Use Regular Expression patterns
Trim your parts

# CREATE NEW OBJECTS

Create hash table of “properties”

- Make your hash table ordered in PowerShell 3.0

Create object with New-Object

- …or [pscustomobject]

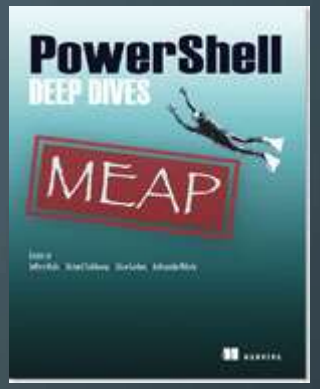
Use Add-Member to add properties and methods

# EXAMPLES AND DEMOS

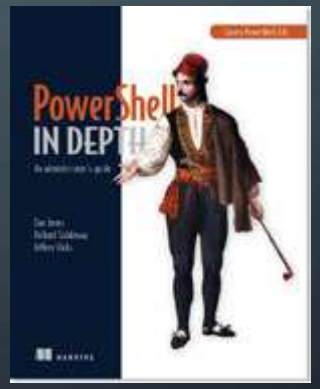
*Focus on technique examples not commands*

# QUESTIONS/ANSWERS

# RESOURCES AND LINKS




- PowerShell Deep Dives
- PowerShell in Depth: An Administrator's Guide
- http://jdhitsolutions.com/blog
- http://bit.ly/13X8IiV
- http://bit.ly/1a2E6Bc



# THANK YOU

http://jdhitsolutions.com/blog

jhicks@jdhitsolutions.com

@JeffHicks

http://gplus.to/jeffhicks