CNT 4603: System Administration Spring 2012

Scripting – Windows PowerShell – Part 1

Instructor :   Dr. Mark Llewellyn

markl@cs.ucf.edu

HEC 236, 4078-823-2790

http://www.cs.ucf.edu/courses/cnt4603/spr2012

Department of Electrical Engineering and Computer Science

Computer Science Division

University of Central Florida

Scripting – Windows PowerShell

- A shell is software that provides a customized interface designed for executing commands or scripts. (The term originated from OS nomenclature where the outer layer of a layered architecture OS was the interface between the user and the kernel of the OS.)

- Most OS shells generally fall into one of two categories: command-line and graphical . Command-line shells provide a command-line interface (CLI) to the OS, while graphical shells provide a graphical user interface (GUI).

- In either category the primary purpose of the shell is to invoke or "launch" other programs. In most modern environments, shells frequently have additional capabilities such as viewing the contents of directories.

Scripting – Windows PowerShell

- Windows PowerShell is a command-line interface (CLI).

- Two important features of PowerShell are scripts and cmdlets.

- A script is a file of commands that is run when you execute or invoke the script.

- A cmdlet (short for command-let, its pronounced like the long version) is a specialized (lightweight – they are technically instances of .NET Framework classes and are not stand-alone executables) command for completing common tasks in the PowerShell environment.

- There are about 130 built-in cmdlets already defined in PowerShell and you can also define (create) your own custom cmdlets as well as import third-party cmdlets.

Scripting – Windows PowerShell

- Windows PowerShell is particularly well suited for situations in which there are multiple servers and it is more efficient to manage them using a consistent set of scripts.

- It is also ideal for managing servers with the Application Server role installed in situations where the applications need to be configured the same way and regular updates are applied.

- Windows Server 2008 comes with PowerShell and it can be installed via the Server Manager (it also comes with Windows 7) and it can be easily downloaded into older server versions.

Scripting – Windows PowerShell

- Some of the more common server administration tasks that can easily be handled through PowerShell include:

    - Managing files and folders (directories).

    - Managing network tasks.

    - Managing fixed and removable storage devices.

    - Configuring printing services.

    - Managing software applications and updates.

    - Managing Terminal Services.

    - Managing server services and features.

    - Managing Web server services

    - Working with the system registry.

Scripting – Windows PowerShell

- PowerShell is not installed by default in Server 2008 (although it should be in Server 2008 R2 editions).

- To install PowerShell from the Server Manager:

  - Scroll down to the Features Summary.

  - Click Add Features.

  - Under Features, scroll to find Windows PowerShell and check its box.

  - Click Next and then click Install.

  - Click Close.

  - Close the Server Manager.

- The next few screen shots step you through this simple process.

Scripting – Windows PowerShell

- Once you've installed PowerShell on the server, you're reading to take advantage of some of the cmdlets.

- With PowerShell installed, you should be able to find it on the server under the Start menu, click All Programs, click Accessories, Click Windows PowerShell, and Windows PowerShell should be there (see the next page).

  - Note: there will also be a Windows PowerShell ISE, which is the Integrated Scripting Environment. We'll look at this later.

- Once you click on Windows PowerShell, you should see a screen like the one shown on page 14.

You'll find PowerShell under the Accessories list from the Start, All Programs listing

File ▼ Virtual Machine ▼ Help ▼

Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> _
```

This is the default appearance with the window maximized. To change this setting see pages 43-44.

Start   Administrator: Windo...   2:30 PM

To direct input to this virtual machine, press Ctrl+G.   **vm**ware

Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\users\Administrator> _
```

To direct input to this virtual machine, press Ctrl+G.

Start   Administrator: Windo...   2:50 PM

vmware

Scripting – Windows PowerShell

- To view the files in the current folder (the default folder will be the Users/Administrator folder), one page of files at a time, enter the command:

```
dir | more
```

- Press Enter after typing in the command (pressing the spacebar will give you the next page if there is one – probably not on our servers, since we don't have much out there yet).

- What you're doing here is executing the directory command and piping its output through to the more command which displays input one page at a time.

- The next page shows the execution of this command on one of my virtual servers.

Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\users\Administrator> dir | more


    Directory: C:\users\Administrator


Mode                LastWriteTime     Length Name
----                -------------            ------ ----
d----        10/3/2011   10:13 AM            .grasp_settings
d----       11/16/2011    4:41 PM            .idlerc
d----        3/15/2012   11:28 AM            .m2
d-r--        9/21/2011    5:02 PM            Contacts
d-r--        2/22/2012    2:15 PM            Desktop
d-r--         3/1/2012    8:21 AM            Documents
d-r--        2/22/2012    2:16 PM            Downloads
d-r--        9/21/2011    5:02 PM            Favorites
d-r--        9/21/2011    5:02 PM            Links
d-r--        9/21/2011    5:02 PM            Music
d----       11/16/2011   12:15 PM            MyScripts
d-r--        9/21/2011    5:02 PM            Pictures
d----       11/16/2011    4:30 PM            Python Scripts
d-r--        9/21/2011    5:02 PM            Saved Games
d-r--        9/21/2011    5:02 PM            Searches
d-r--        9/21/2011    5:02 PM            Videos
d----       11/29/2011    4:41 PM            workspace


PS C:\users\Administrator> _
```

To direct input to this virtual machine, press Ctrl+G.

Start   Administrator: Windo...   2:50 PM

Scripting – Windows PowerShell

- To get a listing of the services currently running on your server, enter the command `get-service`, at the command prompt. A partial listing of the output of this cmdlet is shown on page 19.

- To view a listing of all the currently defined cmdlets, enter the command `get-command | more`, at the command prompt. Here you will see the cmdlets one screen at a time, so press the spacebar to advance to the next screen. Simply repeat this until you've seen all the pages, or alternatively, press `q`, to quite and exit back to the command line if you don't want to view all the pages. This command is illustrated on page 20.

Administrator: Windows PowerShell

```
PS C:\users\Administrator> get-service

Status     Name               DisplayName
------     ----               -----------
Running    AeLookupSvc        Application Experience
Stopped    ALG                Application Layer Gateway Service
Stopped    Apache2.2          Apache2.2
Stopped    Appinfo            Application Information
Stopped    AppMgmt            Application Management
Stopped    AudioEndpointBu... Windows Audio Endpoint Builder
Stopped    Audiosrv           Windows Audio
Running    BFE                Base Filtering Engine
Running    BITS               Background Intelligent Transfer Ser...
Stopped    Browser            Computer Browser
Stopped    CertPropSvc        Certificate Propagation
Stopped    clr_optimizatio... Microsoft .NET Framework NGEN v2.0....
Stopped    clr_optimizatio... Microsoft .NET Framework NGEN v4.0....
Running    COMSysApp          COM+ System Application
Running    CryptSvc           Cryptographic Services
Stopped    CscService         Offline Files
Running    DcomLaunch         DCOM Server Process Launcher
Running    Dhcp               DHCP Client
Running    Dnscache           DNS Client
Stopped    dot3svc            Wired AutoConfig
Running    DPS                Diagnostic Policy Service
Stopped    EapHost            Extensible Authentication Protocol
Running    EventLog           Windows Event Log
Running    EventSystem        COM+ Event System
Stopped    FCRegSvc           Microsoft Fibre Channel Platform Re...
Stopped    fdPHost            Function Discovery Provider Host
Running    FDResPub           Function Discovery Resource Publica...
Running    FontCache          Windows Font Cache Service
Stopped    FontCache3.0.0.0   Windows Presentation Foundation Fon...
Running    gpsvc              Group Policy Client
Stopped    hidserv            Human Interface Device Access
Stopped    hkmsvc             Health Key and Certificate Management
Stopped    idsvc              Windows CardSpace
Running    IKEEXT             IKE and AuthIP IPsec Keying Modules
Stopped    IPBusEnum          PnP-X IP Bus Enumerator
Running    iphlpsvc           IP Helper
Stopped    KeyIso             CNG Key Isolation
Running    KtmRm              KtmRm for Distributed Transaction C...
Running    LanmanServer       Server
Running    LanmanWorkstation  Workstation
```

Start   Administrator: Windo...   2:51 PM

To direct input to this virtual machine, press Ctrl+G.   vmware

Administrator: Windows PowerShell

```
PS C:\users\Administrator> get-command | more

CommandType     Name                      Definition
-----------     ----                      ----------
Alias           %                         ForEach-Object
Alias           ?                         Where-Object
Function        A:                        Set-Location A:
Alias           ac                        Add-Content
Cmdlet          Add-Computer              Add-Computer [-DomainName] <String> [-Credenti
Cmdlet          Add-Content               Add-Content [-Path] <String[]> [-Value] <Objec
Cmdlet          Add-History               Add-History [[-InputObject] <PSObject[]>] [-Pa
Cmdlet          Add-Member                Add-Member [-MemberType] <PSMemberTypes> [-Nam
Cmdlet          Add-PSSnapin              Add-PSSnapin [-Name] <String[]> [-PassThru] [-
Cmdlet          Add-Type                  Add-Type [-TypeDefinition] <String> [-Language
Alias           asnp                      Add-PSSnapIn
Function        B:                        Set-Location B:
Function        C:                        Set-Location C:
Alias           cat                       Get-Content
Alias           cd                        Set-Location
Function        cd..                      Set-Location ..
Function        cd\                       Set-Location \
Alias           chdir                     Set-Location
Cmdlet          Checkpoint-Computer       Checkpoint-Computer [-Description] <String> [[
Alias           clc                       Clear-Content
Alias           clear                     Clear-Host
Cmdlet          Clear-Content             Clear-Content [-Path] <String[]> [-Filter <Str
Cmdlet          Clear-EventLog            Clear-EventLog [-LogName] <String[]> [[-Comput
Cmdlet          Clear-History             Clear-History [[-Id] <Int32[]>] [[-Count] <Int
Function        Clear-Host                $space = New-Object System.Management.Automati
Cmdlet          Clear-Item                Clear-Item [-Path] <String[]> [-Force] [-Filte
Cmdlet          Clear-ItemProperty        Clear-ItemProperty [-Path] <String[]> [-Name]
Cmdlet          Clear-Variable            Clear-Variable [-Name] <String[]> [-Include <S
Alias           clhy                      Clear-History
Alias           cli                       Clear-Item
Alias           clp                       Clear-ItemProperty
Alias           cls                       Clear-Host
Alias           clv                       Clear-Variable
Alias           compare                   Compare-Object
Cmdlet          Compare-Object            Compare-Object [-ReferenceObject] <PSObject[]>
Cmdlet          Complete-Transaction      Complete-Transaction [-Verbose] [-Debug] [-Err
Cmdlet          Connect-WSMan             Connect-WSMan [[-ComputerName] <String>] [-App
Cmdlet          ConvertFrom-Csv           ConvertFrom-Csv [-InputObject] <PSObject[]> [[
Cmdlet          ConvertFrom-SecureString  ConvertFrom-SecureString [-SecureString] <Secu
Cmdlet          ConvertFrom-StringData    ConvertFrom-StringData [-StringData] <String>
```

Start   Administrator: Windo...   2:58 PM

To direct input to this virtual machine, press Ctrl+G.   vmware
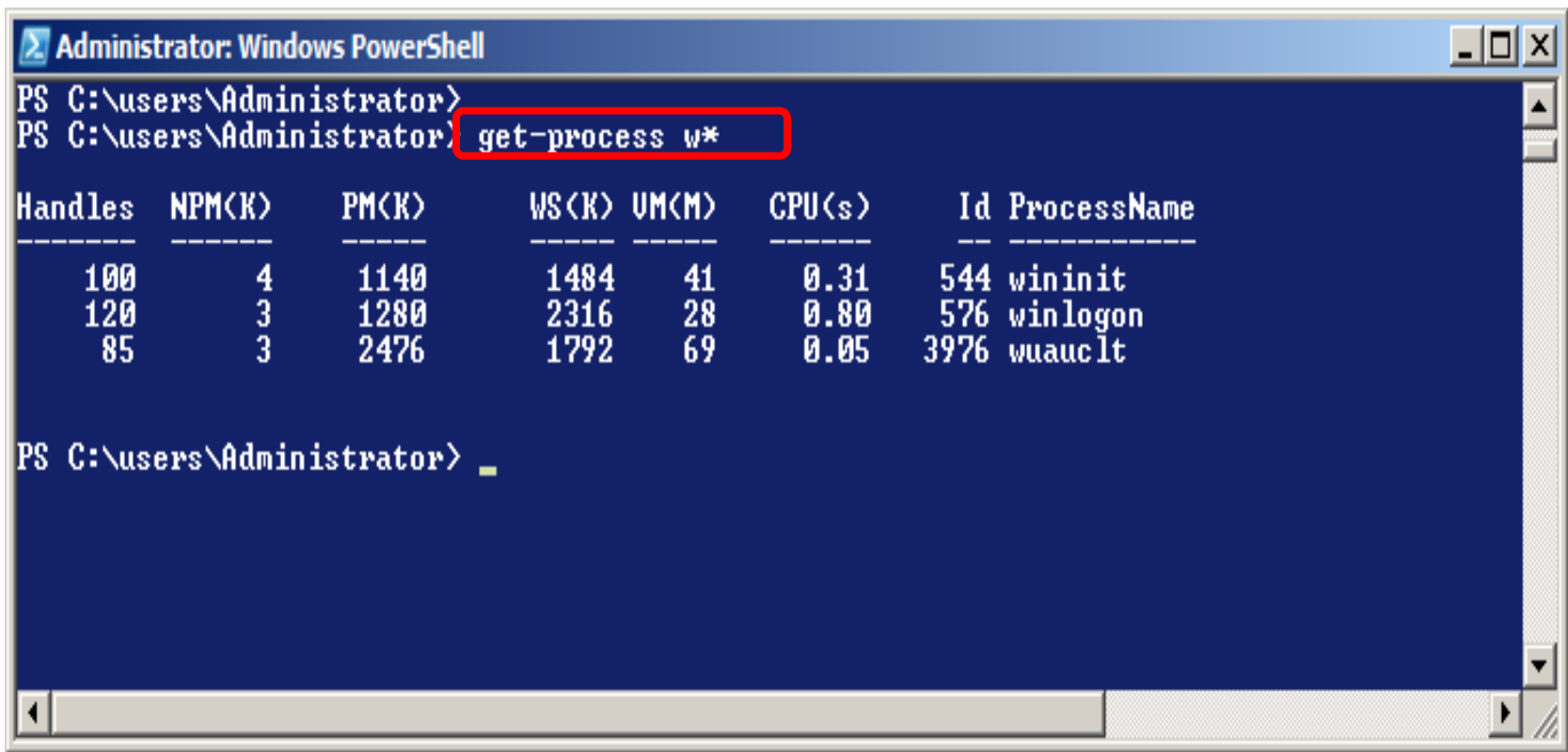
Scripting – Windows PowerShell

- One big plus of PowerShell is consistency. With many shells, the commands can vary in complexity; however, given the object-oriented nature of PowerShell, most cmdlets are fairly basic in their usage and are highly consistent.

- The power comes is using combinations of cmdlets.

- The cmdlets naming convention is for the first part to be a verb (for example, `get-`, `format-`, `out-`, or `set-`) that dictates what the cmdlet does (such as get information, format information, direct information, or set information).

- The next part is a noun, which specifies what is being acted on.

Scripting – Windows PowerShell

- Everything is based around this verb-noun pair; for example, `get-process w*` retrieves information about processes whose names start with the letter w, as shown below.
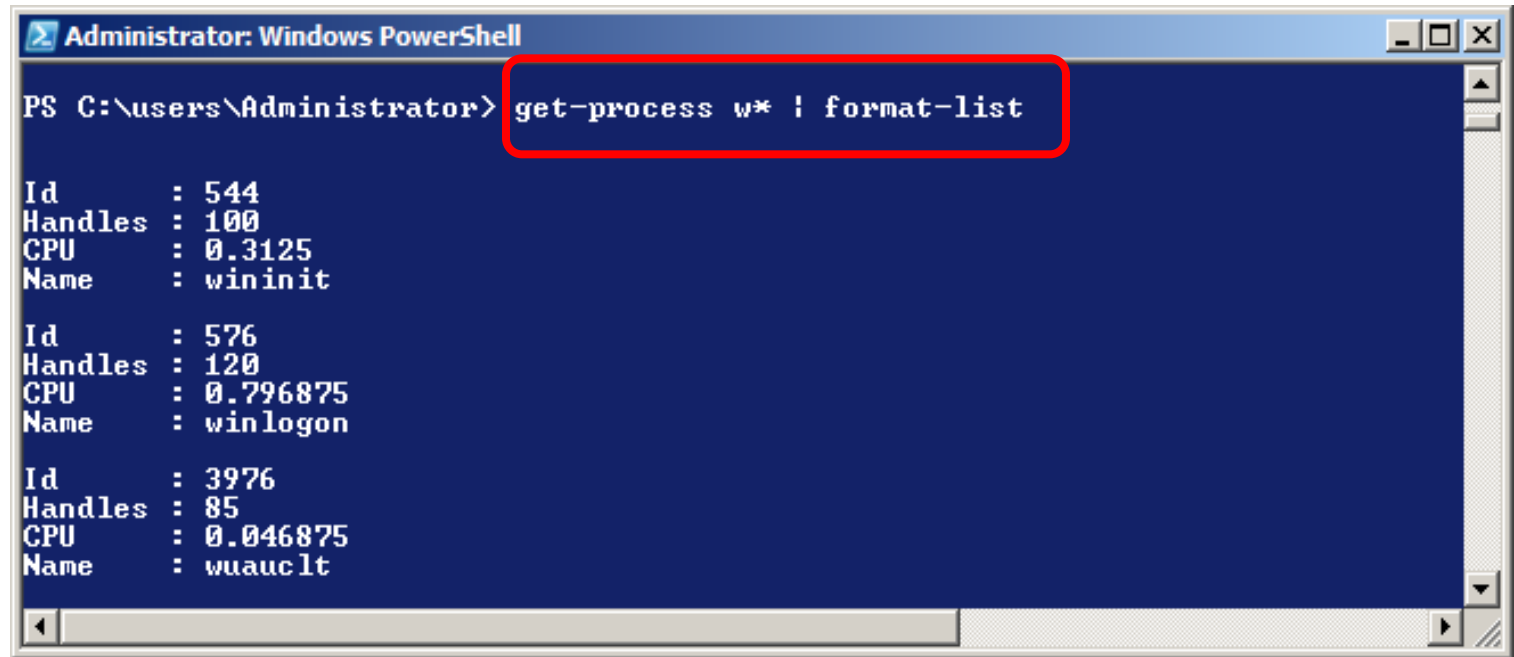
```
Administrator: Windows PowerShell                                    _ □ ✕

PS C:\users\Administrator>
PS C:\users\Administrator> get-process w*

Handles  NPM(K)    PM(K)     WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----     ----- -----   ------     -- -----------
    100       4     1140      1484    41     0.31    544 wininit
    120       3     1280      2316    28     0.80    576 winlogon
     85       3     2476      1792    69     0.05   3976 wuauclt


PS C:\users\Administrator> _
```

Scripting – Windows PowerShell

- Although the output, as shown on the previous page, is tabular, this is not how the data is returned in PowerShell. It's referenced in its .NET object format, but the default display format is a table.

- You can easily output in other formats, such as a list by piping the output of the `get-process` cmdlet to the `format-list` cmdlet.

Scripting – Windows PowerShell

- Probably the greatest cmdlet (as well as the best verb-noun combination) that you'll ever use is `get-help`.

- On its own, `get-help` gives you just basic information, but it can show you the names of other cmdlets, so you can detailed help on them.

- For example, `get-help format-*` will list all the cmdlets starting with format- to help you see the options available to you.

Administrator: Windows PowerShell

```
PS C:\users\Administrator> get-help format-*

Name                          Category    Synopsis
----                          --------    --------
Format-List                   Cmdlet      Formats the output as a list of properties in which each property appe
Format-Custom                 Cmdlet      Uses a customized view to format the output.
Format-Table                  Cmdlet      Formats the output as a table.
Format-Wide                   Cmdlet      Formats objects as a wide table that displays only one property of eac

PS C:\users\Administrator>
```

Start   Administrator: Windo...   3:07 PM

To direct input to this virtual machine, press Ctrl+G.   vmware

Scripting – Windows PowerShell

- In addition to getting detailed help about a cmdlet, use the `get-help` command with the name of the cmdlet followed by `-detailed` to get all available help.

- Add `-full` to just view a portion of the help, or add `-examples` to have examples of use listed for you.

- Note that when the `-detailed` option is selected, the examples are also listed.

- The following screen shots illustrates these cases. Note that the detailed case requires several pages of output and I only show the first one here. The same is often true for full and examples.

Administrator: Windows PowerShell

```
PS C:\users\Administrator> get-help format-list -detailed

NAME
    Format-List

SYNOPSIS
    Formats the output as a list of properties in which each property appears on a new line.


SYNTAX
    Format-List [[-Property] <Object[]>] [-DisplayError] [-Expand <string>] [-Force] [-GroupBy <Object>] [-InputOb
    <psobject>] [-ShowError] [-View <string>] [<CommonParameters>]


DESCRIPTION
    The Format-List cmdlet formats the output of a command as a list of properties in which each property is displ
    on a separate line. You can use Format-List to format and display all or selected properties of an object as a
     (format-list *).

    Because more space is available for each item in a list than in a table, Windows PowerShell displays more prop
    s of the object in the list, and the property values are less likely to be truncated.


PARAMETERS
    -DisplayError [<SwitchParameter>]
        Displays errors at the command line.

    -Expand <string>
        Formats the collection object, as well as the objects in the collection. This parameter is designed to for
        bjects that support the ICollection (System.Collections) interface. The default value is EnumOnly.

        Valid values are:
        -- EnumOnly: Displays the properties of the objects in the collection.
        -- CoreOnly: Displays the properties of the collection object.
        -- Both: Displays the properties of the collection object and the properties of objects in the collection.

    -Force [<SwitchParameter>]
        Directs the cmdlet to display all of the error information. Use with the DisplayError or ShowError paramet
        By default, when an error object is written to the error or display streams, only some of the error inform
         is displayed.

    -GroupBy <Object>
        Formats the output in groups based on a shared property or value. Enter an expression or a property of the
        ut.
```

Start | 📚 💻 🅮 🅾 | Administrator: Windo...   « 🌀 📺 ⏳ 🔋🔊 3:08 PM

To direct input to this virtual machine, press Ctrl+G.

Administrator: Windows PowerShell

```
PS C:\users\Administrator> get-help format-list -examples

NAME
    Format-List

SYNOPSIS
    Formats the output as a list of properties in which each property appears on a new line.

    --------------------------- EXAMPLE 1 ---------------------------

    C:\PS>get-service | format-list

    Description
    -----------
    This command formats information about services on the computer as a list. By default, the services are format
    s a table. The Get-Service cmdlet gets objects representing the services on the computer. The pipeline operato
    passes the results through the pipeline to Format-List. Then, the Format-List command formats the service inf
    ion in a list and sends it to the default output cmdlet for display.

    --------------------------- EXAMPLE 2 ---------------------------

    C:\PS>$a = get-childitem $pshome\*.ps1xml

    Description
    -----------
    These commands display information about the PS1XML files in the Windows PowerShell directory as a list. The f
    command gets the objects representing the files and stores them in the $a variable. The second command uses Fo
    List to format information about objects stored in $a. This command uses the InputObject parameter to pass the
    able to Format-List, which then sends the formatted output to the default output cmdlet for display.

    --------------------------- EXAMPLE 3 ---------------------------

    C:\PS>get-process | format-list -property name, basepriority, priorityclass

    Description
    -----------
```

Scripting – Windows PowerShell

- We've already seen the cmdlet `get-command`. If you want to see all the commands that begin with a certain verb, such as `get`, issue the command `get-command -verb get`.

- The output of this command is shown on the next page, but you might want to experiment a bit and try out some other options. For example, try listing all of the commands that use the verbs `add` or `new`.

Administrator: Windows PowerShell

```
PS C:\users\Administrator> get-command -verb get

CommandType     Name                              Definition
-----------     ----                              ----------
Cmdlet          Get-Acl                           Get-Acl [[-Path] <String[]>] [-Audit] [-Filter
Cmdlet          Get-Alias                         Get-Alias [[-Name] <String[]>] [-Exclude <Stri
Cmdlet          Get-AuthenticodeSignature         Get-AuthenticodeSignature [-FilePath] <String[
Cmdlet          Get-ChildItem                     Get-ChildItem [[-Path] <String[]>] [[-Filter]
Cmdlet          Get-Command                       Get-Command [[-ArgumentList] <Object[]>] [-Ver
Cmdlet          Get-ComputerRestorePoint          Get-ComputerRestorePoint [[-RestorePoint] <Int
Cmdlet          Get-Content                       Get-Content [-Path] <String[]> [-ReadCount <In
Cmdlet          Get-Counter                       Get-Counter [[-Counter] <String[]>] [-SampleIn
Cmdlet          Get-Credential                    Get-Credential [-Credential] <PSCredential> [-
Cmdlet          Get-Culture                       Get-Culture [-Verbose] [-Debug] [-ErrorAction
Cmdlet          Get-Date                          Get-Date [[-Date] <DateTime>] [-Year <Int32>]
Cmdlet          Get-Event                         Get-Event [[-SourceIdentifier] <String>] [-Ver
Cmdlet          Get-EventLog                      Get-EventLog [-LogName] <String> [[-InstanceId
Cmdlet          Get-EventSubscriber               Get-EventSubscriber [[-SourceIdentifier] <Stri
Cmdlet          Get-ExecutionPolicy               Get-ExecutionPolicy [[-Scope] <ExecutionPolicy
Cmdlet          Get-FormatData                    Get-FormatData [[-TypeName] <String[]>] [-Verb
Cmdlet          Get-Help                          Get-Help [[-Name] <String>] [-Path <String>] [
Cmdlet          Get-History                       Get-History [[-Id] <Int64[]>] [[-Count] <Int32
Cmdlet          Get-Host                          Get-Host [-Verbose] [-Debug] [-ErrorAction <Ac
Cmdlet          Get-HotFix                        Get-HotFix [[-Id] <String[]>] [-ComputerName <
Cmdlet          Get-Item                          Get-Item [-Path] <String[]> [-Filter <String>]
Cmdlet          Get-ItemProperty                  Get-ItemProperty [-Path] <String[]> [[-Name] <
Cmdlet          Get-Job                           Get-Job [[-Id] <Int32[]>] [-Verbose] [-Debug]
Cmdlet          Get-Location                      Get-Location [-PSProvider <String[]>] [-PSDriv
Cmdlet          Get-Member                        Get-Member [[-Name] <String[]>] [-InputObject
Cmdlet          Get-Module                        Get-Module [[-Name] <String[]>] [-All] [-Verbo
Cmdlet          Get-PfxCertificate                Get-PfxCertificate [-FilePath] <String[]> [-Ve
Cmdlet          Get-Process                       Get-Process [[-Name] <String[]>] [-ComputerNam
Cmdlet          Get-PSBreakpoint                  Get-PSBreakpoint [[-Script] <String[]>] [-Verb
Cmdlet          Get-PSCallStack                   Get-PSCallStack [-Verbose] [-Debug] [-ErrorAct
Cmdlet          Get-PSDrive                       Get-PSDrive [[-Name] <String[]>] [-Scope <Stri
Cmdlet          Get-PSProvider                    Get-PSProvider [[-PSProvider] <String[]>] [-Ve
Cmdlet          Get-PSSession                     Get-PSSession [[-ComputerName] <String[]>] [-U
Cmdlet          Get-PSSessionConfiguration        Get-PSSessionConfiguration [[-Name] <String[]>
Cmdlet          Get-PSSnapin                      Get-PSSnapin [[-Name] <String[]>] [-Registered
Cmdlet          Get-Random                        Get-Random [[-Maximum] <Object>] [-SetSeed <Nu
Cmdlet          Get-Service                       Get-Service [[-Name] <String[]>] [-ComputerNam
Cmdlet          Get-TraceSource                   Get-TraceSource [[-Name] <String[]>] [-Verbose
Cmdlet          Get-Transaction                   Get-Transaction [-Verbose] [-Debug] [-ErrorAct
Cmdlet          Get-UICulture                     Get-UICulture [-Verbose] [-Debug] [-ErrorActio
```

Start | Administrator: Windo...   3:10 PM

To direct input to this virtual machine, press Ctrl+G.

Scripting – Windows PowerShell

- Now that you've have some basic familiarity with PowerShell, let's do something more useful with it… let's try starting and stopping a process.

    - What you might want to do before going any further is first run the `get-help *-process` to list all the available commands that deal with a process. You should discover that there are five of these cmdlets.

- What we're going to do over the next few pages is start Notepad as a process running on our server and then use it and then stop the process. This will be illustrated by a sequence of screen shots from the server illustrating what is happening.

- First off, we'll see a screen shot of the current processes on the server. Notice that its alphabetically listed and Notepad is not running (Notepad++ is on my server).

Administrator: Windows PowerShell

PS C:\users\Administrator> get-process

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
405	5	1680	2544	105	1.19	492	csrss
210	7	7112	5472	51	17.34	536	csrss
230	7	5700	2832	68	0.44	2552	dllhost
79	3	1212	2732	45	0.72	2276	dwm
531	14	22388	27676	164	42.33	2300	explorer
0	0	0	24	0		0	Idle
201	6	3244	3368	63	0.14	2092	jucheck
209	6	2372	2988	69	0.28	3048	jusched
588	9	3204	4748	45	2.23	636	lsass
159	3	1596	2152	29	0.22	644	lsm
162	7	2812	2472	60	0.28	2740	msdtc
516	6	50612	25136	101	0.50	3284	mysqld
89	5	7100	11436	66	0.50	2220	notepad++
368	8	48348	50692	182	2.77	3300	powershell
243	7	2388	3808	37	13.55	624	services
96	3	5756	3176	40	1.47	1040	SLsvc
28	1	256	484	4	0.22	424	smss
309	10	7492	5464	100	2.39	1540	spoolsv
300	5	2724	3396	38	7.94	828	svchost
262	7	2640	3292	35	0.73	888	svchost
306	10	6236	5684	50	21.41	964	svchost
148	4	2876	3392	35	0.30	1008	svchost
1053	36	58280	50004	200	37.66	1028	svchost
576	17	6624	5888	60	1.06	1084	svchost
255	8	7320	4884	66	0.73	1148	svchost
431	14	14788	6836	88	4.20	1172	svchost
268	22	6100	5216	47	1.06	1336	svchost
125	5	1860	1352	36	0.14	1792	svchost
73	2	832	944	23	0.05	1804	svchost
81	3	1016	1388	23	0.20	1936	svchost
228	7	3220	1684	44	0.11	3096	svchost
64	2	1284	1508	25	0.11	3720	svchost
513	0	0	728	4		4	System
137	5	1868	2156	53	0.25	1448	taskeng
254	7	2816	4632	74	0.69	2200	taskeng
47	2	1028	752	45	0.03	3068	Tomcat7.0.22w
66	2	1252	2480	49	0.22	3084	Tomcat7.0.25w
111	4	2272	3792	63	22.67	2704	TPAutoConnect
127	4	2304	3488	55	0.36	2380	TPAutoConnSvc
254	7	6384	4968	88	13.44	1860	vmtoolsd

Currently running processes do not include Notepad.

Start | Administrator: Windo... | new 1 - Notepad++   3:12 PM

To direct input to this virtual machine, press Ctrl+G.   vmware

Administrator: Windows PowerShell

```
 588      9    3204    4748    45   2.23    636 lsass
 159      3    1596    2152    29   0.22    644 lsm
 162      7    2812    2472    60   0.28   2740 msdtc
 516      6   50612   25136   101   0.50   3284 mysqld
```

Untitled - Notepad

File   Edit   Format   View   Help

> Enter the command: `start notepad` on the command line and press enter. Notepad immediately launches.

```
 254      7    6384    4968    88  13.44   1860 vmtoolsd
  92      4    2744    2028    46   0.05   2012 VMUpgradeHelper
  68      3    2480    2328    54   0.30   2940 VMwareTray
 179      5   10664   10848   103   3.72   2964 VMwareUser
 100      4    1140    1484    41   0.31    544 wininit
 120      3    1280    2316    28   0.80    576 winlogon
  85      3    2476    1792    69   0.05   3976 wuauclt
```

```
PS C:\users\Administrator> start notepad
PS C:\users\Administrator>
```

To direct input to this virtual machine, press Ctrl+G.

Start | Administrator: Windows ... | new 1 - Notepad++ | Untitled - Notepad   3:13 PM

vmware

Administrator: Windows PowerShell

```
PS C:\users\Administrator> start notepad
PS C:\users\Administrator> get-process
```

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
398	5	1680	2544	105	1.19	492	csrss
214	7	7112	5480	51	17.53	536	csrss
230	7	5700	2832	68	0.44	2552	dllhost
79	3	1212	2732	45	0.72	2276	dwm
532	14	22388	27672	164	42.34	2300	explorer
0	0	0	24	0		0	Idle
201	6	3244	3368	63	0.14	2092	jucheck
209	6	2372	2988	69	0.28	3048	jusched
589	9	3060	4620	44	2.23	636	lsass
159	3	1596	2152	29	0.22	644	lsm
162	7	2812	2472	60	0.28	2740	msdtc
516	6	50612	25136	101	0.52	3284	mysqld
47	2	1044	3444	47	0.03	3680	notepad
84	5	7080	11424	65	0.50	2220	notepad++
449	8	48548	51512	186	2.94	3300	powershell
243	7	2388	3808	37	13.55	624	services
96	3	5756	3176	40	1.47	1040	SLsvc
28	1	256	484	4	0.22	424	smss
309	10	7492	5464	100	2.39	1540	spoolsv
298	4	2696	3372	38	7.94	828	svchost
260	7	2616	3280	35	0.73	888	svchost
312	10	6264	5700	50	21.42	964	svchost
148	4	2876	3392	35	0.30	1008	svchost
1051	36	58224	49984	199	37.66	1028	svchost
573	17	6620	5872	60	1.06	1084	svchost
255	8	7320	4884	66	0.73	1148	svchost
428	14	14748	6824	88	4.20	1172	svchost
268	22	6128	5228	47	1.06	1336	svchost
125	5	1860	1352	36	0.14	1792	svchost
73	2	832	944	23	0.05	1804	svchost
81	3	1016	1388	23	0.20	1936	svchost
228	7	3220	1684	44	0.11	3096	svchost
64	2	1284	1508	25	0.11	3720	svchost
513	0	0	728	4		4	System
137	5	1868	2156	53	0.25	1448	taskeng
253	7	2796	4620	74	0.69	2200	taskeng
47	2	1028	752	45	0.03	3068	Tomcat7.0.22w
66	2	1252	2480	49	0.22	3084	Tomcat7.0.25w
111	4	2272	3792	63	22.73	2704	TPAutoConnect

Reissue the command `get-process` and notice that now Notepad is listed.

Notice too in the tool tray that the you can still see Notepad is there.

Start   Administrator: Windo...   new 1 - Notepad++   Untitled - Notepad   3:13 PM

To direct input to this virtual machine, press Ctrl+G.   vmware

Mark - TestBed Server - VMware Player   File ▾ Virtual Machine ▾ Help ▾

Administrator: Windows PowerShell

```
PS C:\users\Administrator> stop-process -id 3680
PS C:\users\Administrator> get-process

Handles  NPM(K)    PM(K)    WS(K) VM(M)  CPU(s)    Id ProcessName
-------  ------    -----    ----- -----  ------    -- -----------
    397       5     1680     2544   105    1.19   492 csrss
    210       7     7112     5468    51   17.70   536 csrss
    230       7     5700     2832    68    0.44  2552 dllhost
     79       3     1212     2732    45    0.72  2276 dwm
    525      14    22312    27616   164   42.34  2300 explorer
      0       0        0       24     0          0 Idle
    201       6     3244     3368    63    0.14  2092 jucheck
    209       6     2372     2988    69    0.28  3048 jusched
    588       9     3060     4620    44    2.23   636 lsass
    159       3     1596     2152    29    0.22   644 lsm
    162       7     2812     2472    60    0.28  2740 msdtc
    516       6    50612    25136   101    0.52  3284 mysqld
     84       5     7080    11424    65    0.50  2220 notepad++
    397       8    49956    52864   186    3.03  3300 powershell
    243       7     2388     3808    37   13.55   624 services
     96       3     5756     3176    40    1.47  1040 SLsvc
     28       1      256      484     4    0.22   424 smss
    309      10     7492     5464   100    2.39  1540 spoolsv
    298       4     2696     3372    38    7.94   828 svchost
    258       7     2584     3264    34    0.73   888 svchost
    299      10     6236     5684    50   21.42   964 svchost
    148       4     2876     3392    35    0.30  1008 svchost
   1051      36    58252    49996   200   37.66  1028 svchost
    572      17     6620     5872    60    1.06  1084 svchost
    255       8     7320     4884    66    0.73  1148 svchost
    425      14    14788     6840    88    4.20  1172 svchost
    266      22     6100     5216    47    1.06  1336 svchost
    125       5     1860     1352    36    0.14  1792 svchost
     73       2      832      944    23    0.05  1804 svchost
     81       3     1016     1388    23    0.20  1936 svchost
    228       7     3220     1684    44    0.11  3096 svchost
     64       2     1284     1508    25    0.11  3720 svchost
    512       0        0      728     4          4 System
    137       5     1868     2156    53    0.25  1448 taskeng
    254       7     2816     4632    74    0.70  2200 taskeng
     47       2     1028      752    45    0.03  3068 Tomcat7.0.22w
     66       2     1252     2480    49    0.22  3084 Tomcat7.0.25w
    111       4     2272     3792    63   22.81  2704 TPAutoConnect
```

Notice on the previous screen shot that the id process id of the Notepad process was 3680. This is used in this version of the stop-process command to identify the process to be stopped.

Reissue the command get-process and notice that now Notepad is no longer listed.

Notice too in the tool tray that Notepad is no longer there.

Start   Administrator: Windo...   new 1 - Notepad++   3:15 PM

To direct input to this virtual machine, press Ctrl+G.   vmware

Scripting – Windows PowerShell

- You can also do a fair amount of customization of the PowerShell interface.

- A common system administrator technique is to place scripts in a folder on a server that is frequently backed up. Thus, you might want PowerShell to open up in this default directory.

- To illustrate doing this, let's create a subdirectory in the C:\Users\Administrators folder named MyScripts. Then we'll configure PowerShell to open in this folder.

- To make some of these repetitive steps easier to accomplish, I also created a short-cut to PowerShell and put it on the start menu.

Scripting – Windows PowerShell

- To set-up the default folder for PowerShell to open in, right click the short-cut to PowerShell and select Properties.

- Locate the ShortCut tab on the Properties dialog box and in the Start in: text box enter the path to the new directory "C:\Users\Administrator\MyScripts", then click OK.

- Restart PowerShell and you should now see the new default directory loaded.

Scripting – Windows PowerShell

- You can also change the text size and the screen foreground and background colors and many other features including hot-keys and so on in PowerShell.

- The next part simply shows you how to reset the text size and the screen colors to customize your PowerShell environment.

- Again going through the desktop shortcut to PowerShell, right click on the short cut and select Properties. Locate the Font tab on the Properties dialog box and reset the Window size to 8x8 (the default is 8x12), then click OK.

- Restart PowerShell and you should now see the new default screen size and font size for the window.

Scripting – Windows PowerShell

- To change the screen colors for PowerShell, repeat the process but select the Colors tab.

- Again going through the desktop shortcut to PowerShell, right click on the short cut and select Properties. Locate the Colors tab on the Properties dialog box and reset the colors to your liking, then click OK.

- Restart PowerShell and you should now see the new colors appear.
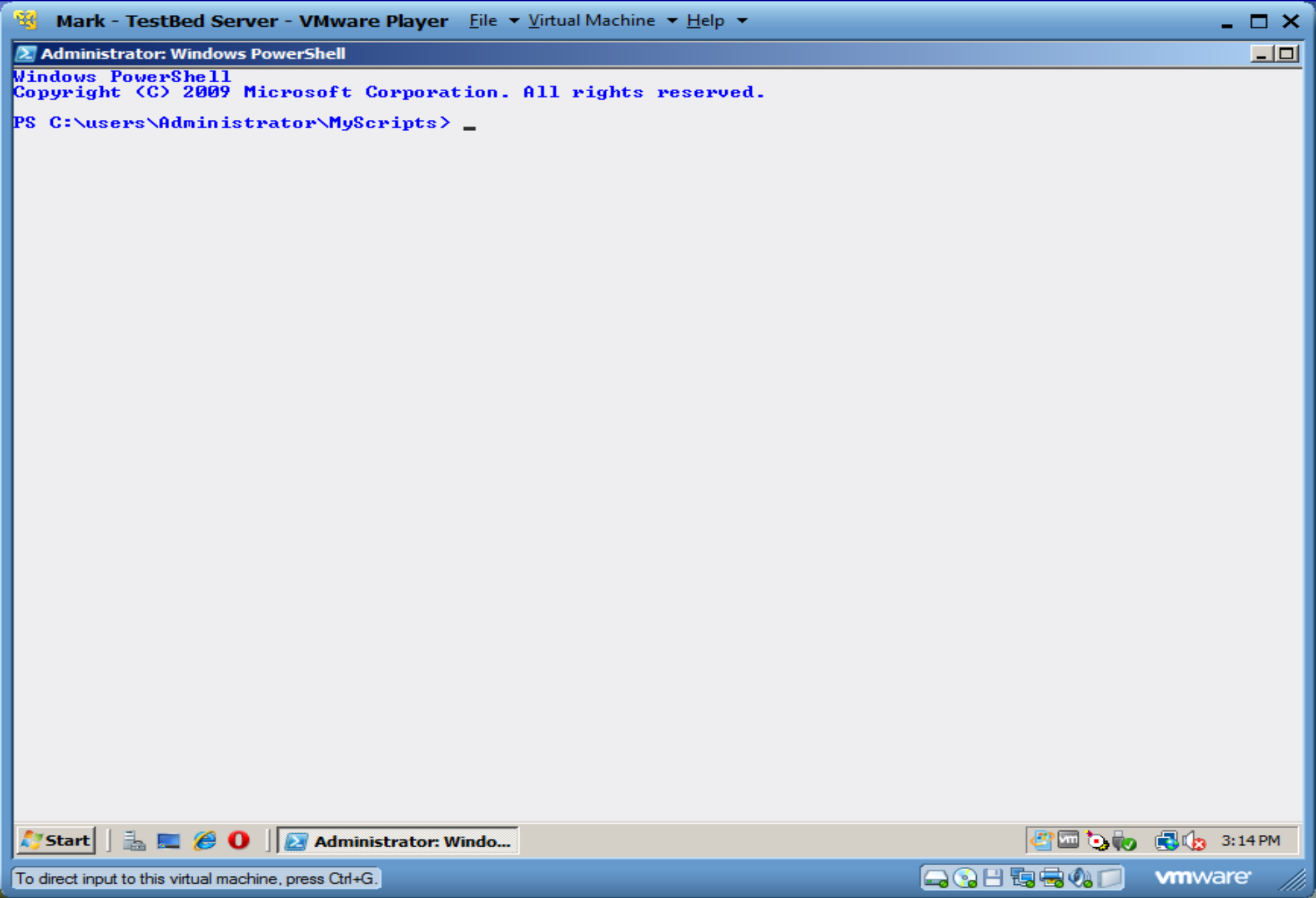
Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\users\Administrator\MyScripts> _
```

Start   Administrator: Windo...   3:14 PM

To direct input to this virtual machine, press Ctrl+G.