CNT 4603: System Administration
Spring 2011

Scripting – Windows PowerShell

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 4078-823-2790
http://www.cs.ucf.edu/courses/cnt4603/spr2011

Department of Electrical Engineering and Computer Science
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) commands 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 third-party cmdlets.

Scripting – Windows PowerShell

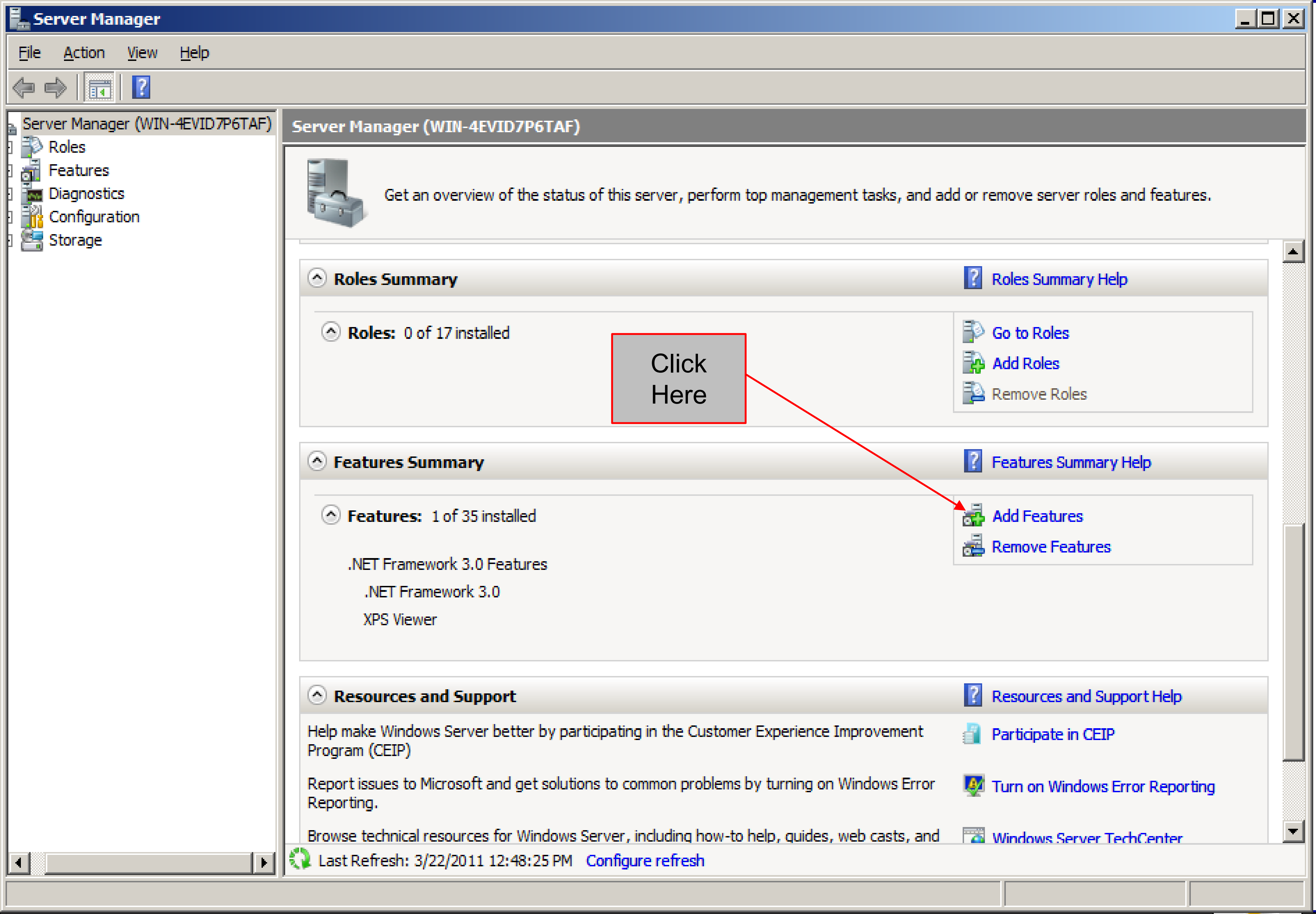
- Windows PowerShell is particularly well suited for situations in which there are multiple servers and it is more efficient to manage them using a consistent set of scripts.
- It is also ideal for managing servers with the Application Server role installed in situations where the applications need to 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.



Server Manager

File Action View Help

Server Manager (WIN-4EVID7P6TAF)
- Roles
- Features
- Diagnostics
- Configuration
- Storage

Server Manager (WIN-4EVID7P6TAF)

Get an overview of the status of this server, perform top management tasks, and add or remove server roles and features.

Roles Summary — Roles Summary Help

Roles: 0 of 17 installed

Go to Roles
Add Roles
Remove Roles

Features Summary — Features Summary Help

Features: 1 of 35 installed

.NET Framework 3.0 Features
.NET Framework 3.0
XPS Viewer

Add Features
Remove Features

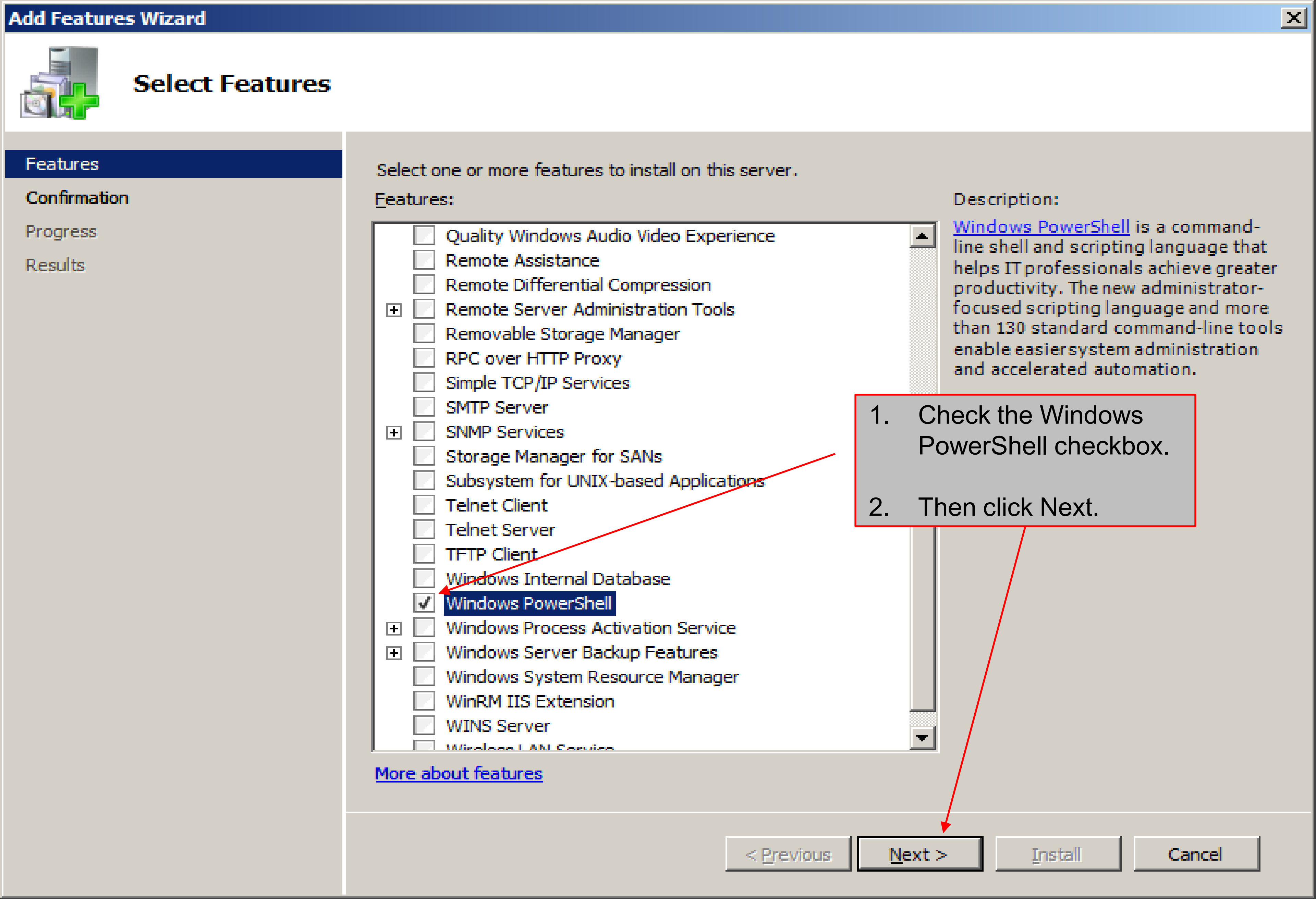
Resources and Support — Resources and Support Help

Help make Windows Server better by participating in the Customer Experience Improvement Program (CEIP) — Participate in CEIP

Report issues to Microsoft and get solutions to common problems by turning on Windows Error Reporting. — Turn on Windows Error Reporting

Browse technical resources for Windows Server, including how-to help, guides, web casts, and — Windows Server TechCenter

Last Refresh: 3/22/2011 12:48:25 PM Configure refresh


Add Features Wizard

Select Features

- Features
- Confirmation
- Progress
- Results

Select one or more features to install on this server.

Features:

- [ ] Quality Windows Audio Video Experience
- [ ] Remote Assistance
- [ ] Remote Differential Compression
- [ ] Remote Server Administration Tools
- [ ] Removable Storage Manager
- [ ] RPC over HTTP Proxy
- [ ] Simple TCP/IP Services
- [ ] SMTP Server
- [ ] SNMP Services
- [ ] Storage Manager for SANs
- [ ] Subsystem for UNIX-based Applications
- [ ] Telnet Client
- [ ] Telnet Server
- [ ] TFTP Client
- [ ] Windows Internal Database
- [✓] Windows PowerShell
- [ ] Windows Process Activation Service
- [ ] Windows Server Backup Features
- [ ] Windows System Resource Manager
- [ ] WinRM IIS Extension
- [ ] WINS Server

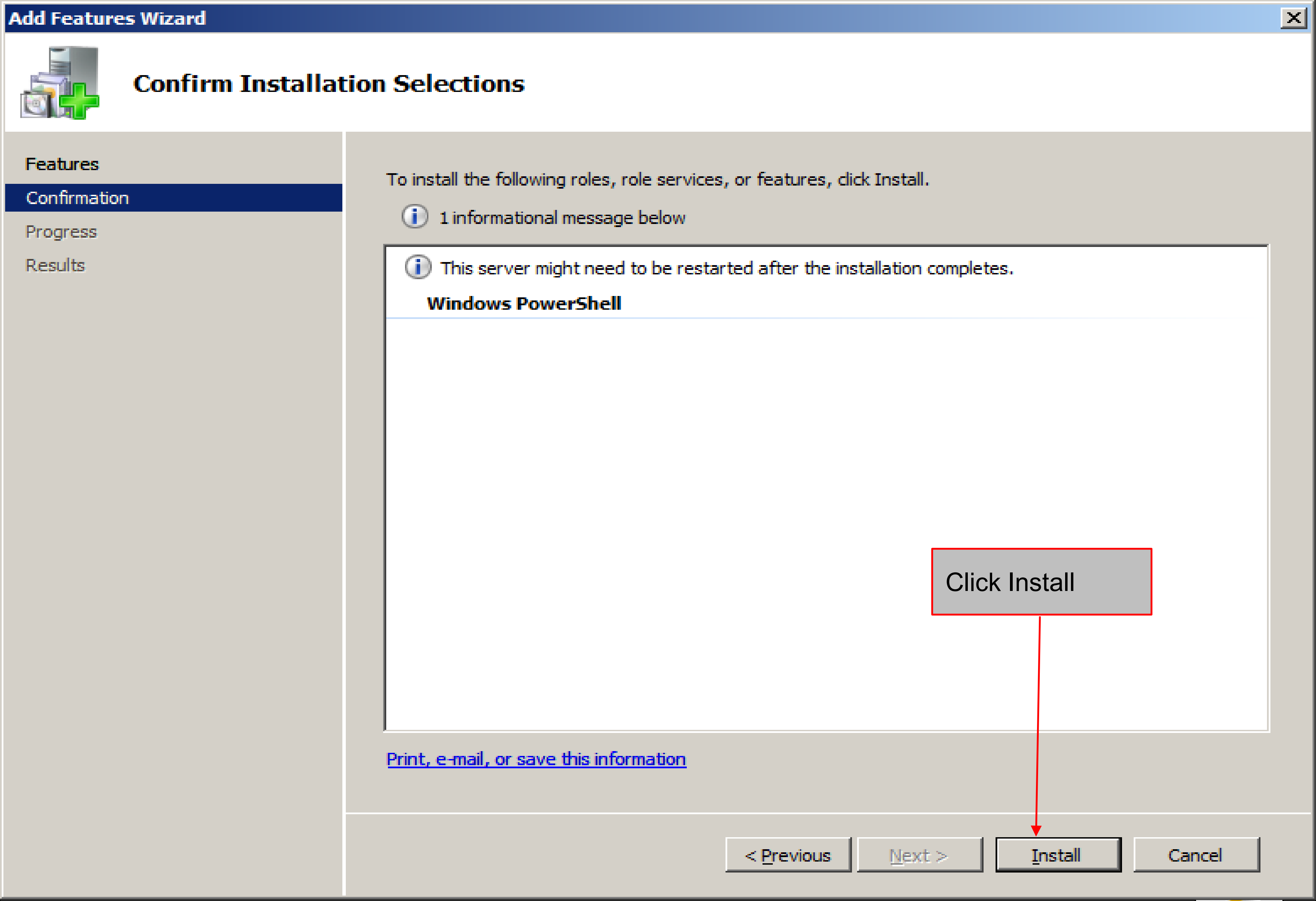
Description:

Windows PowerShell is a command-line shell and scripting language that helps IT professionals achieve greater productivity. The new administrator-focused scripting language and more than 130 standard command-line tools enable easiersystem administration and accelerated automation.

More about features

1. Check the Windows PowerShell checkbox.

2. Then click Next.



Add Features Wizard

Confirm Installation Selections

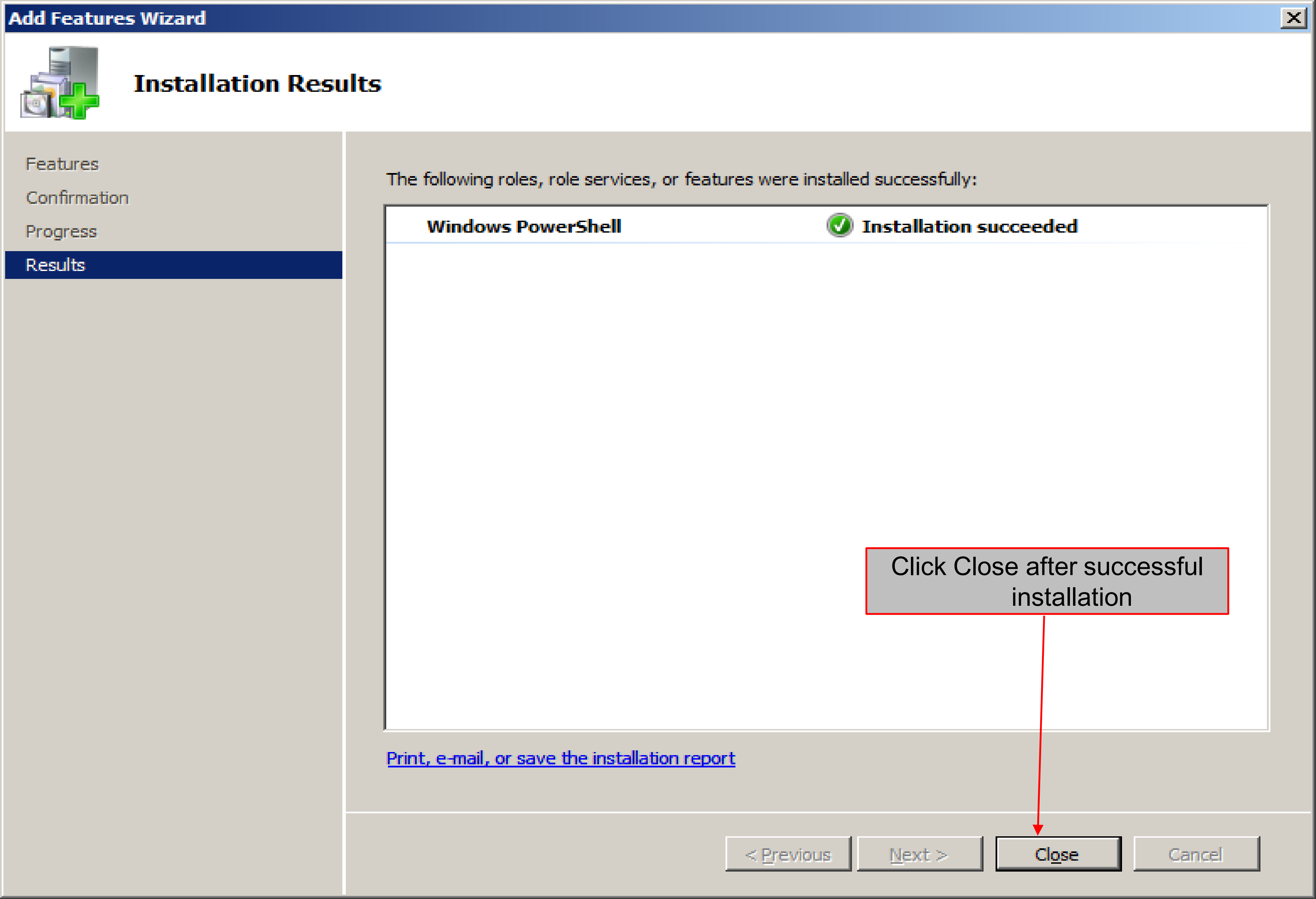
To install the following roles, role services, or features, click Install.

1 informational message below

This server might need to be restarted after the installation completes.

Windows PowerShell

Print, e-mail, or save this information



Add Features Wizard

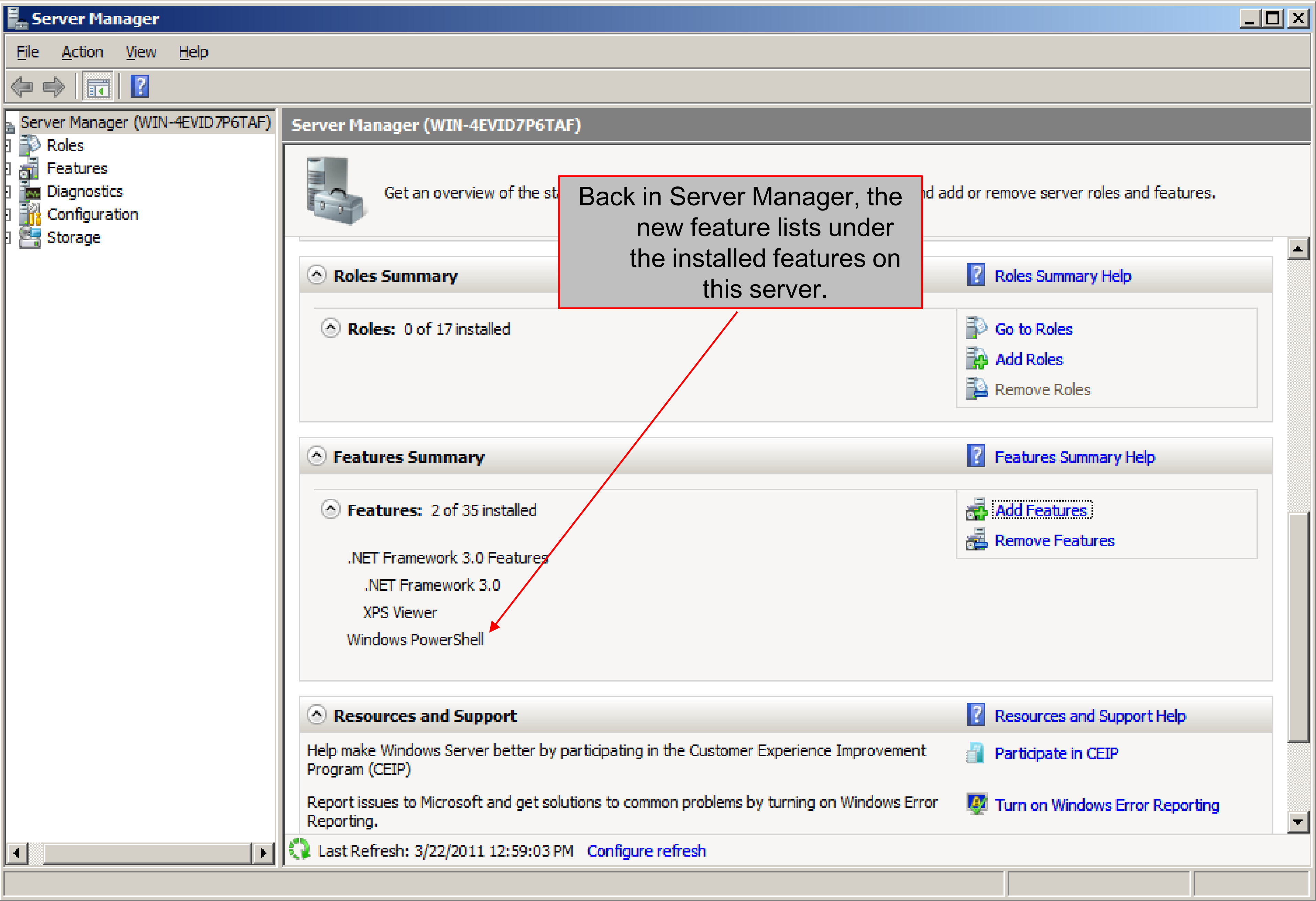
Installation Results

The following roles, role services, or features were installed successfully:

Windows PowerShell	✓ Installation succeeded

Print, e-mail, or save the installation report

Click Close after successful installation



Server Manager

File Action View Help

Server Manager (WIN-4EVID7P6TAF)
- Roles
- Features
- Diagnostics
- Configuration
- Storage

Server Manager (WIN-4EVID7P6TAF)

Get an overview of the st… …d add or remove server roles and features.

Back in Server Manager, the new feature lists under the installed features on this server.

Roles Summary — Roles Summary Help

Roles: 0 of 17 installed

Go to Roles
Add Roles
Remove Roles

Features Summary — Features Summary Help

Features: 2 of 35 installed

.NET Framework 3.0 Features
 .NET Framework 3.0
 XPS Viewer
Windows PowerShell

Add Features
Remove Features

Resources and Support — Resources and Support Help

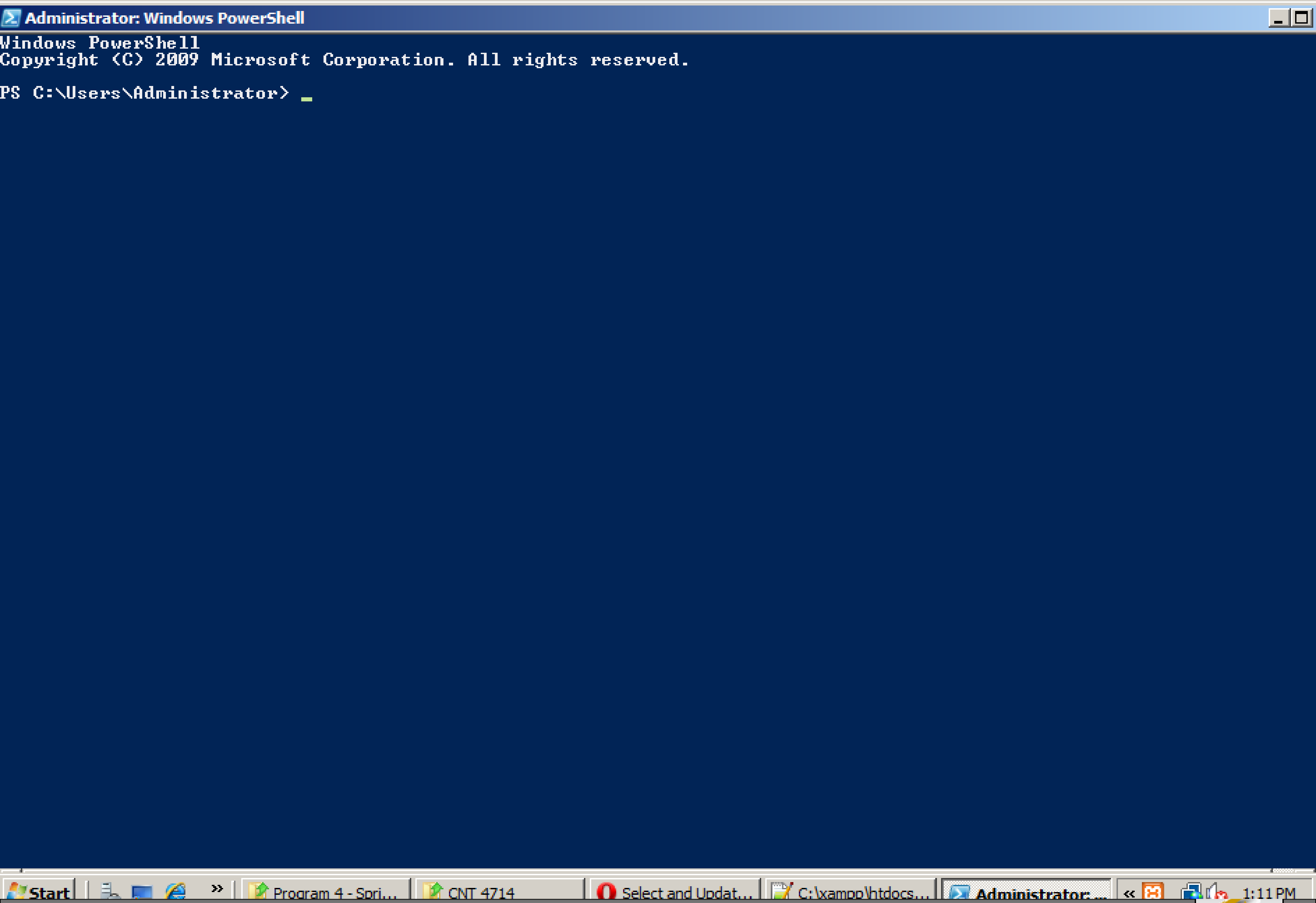
Help make Windows Server better by participating in the Customer Experience Improvement Program (CEIP) — Participate in CEIP

Report issues to Microsoft and get solutions to common problems by turning on Windows Error Reporting. — Turn on Windows Error Reporting

Last Refresh: 3/22/2011 12:59:03 PM Configure refresh

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.
  - 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 the next page.



Administrator: Windows PowerShell

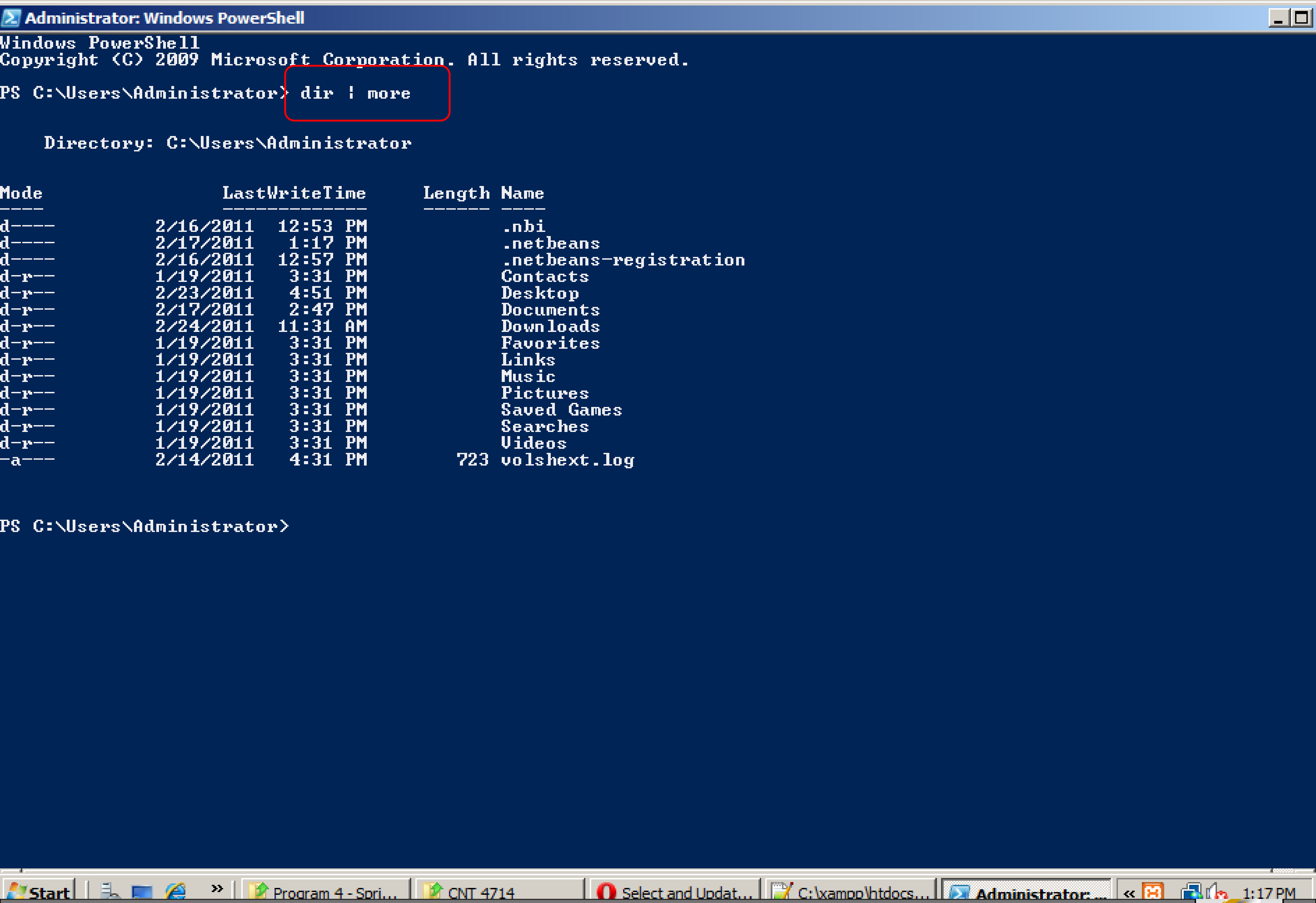
```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

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

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.

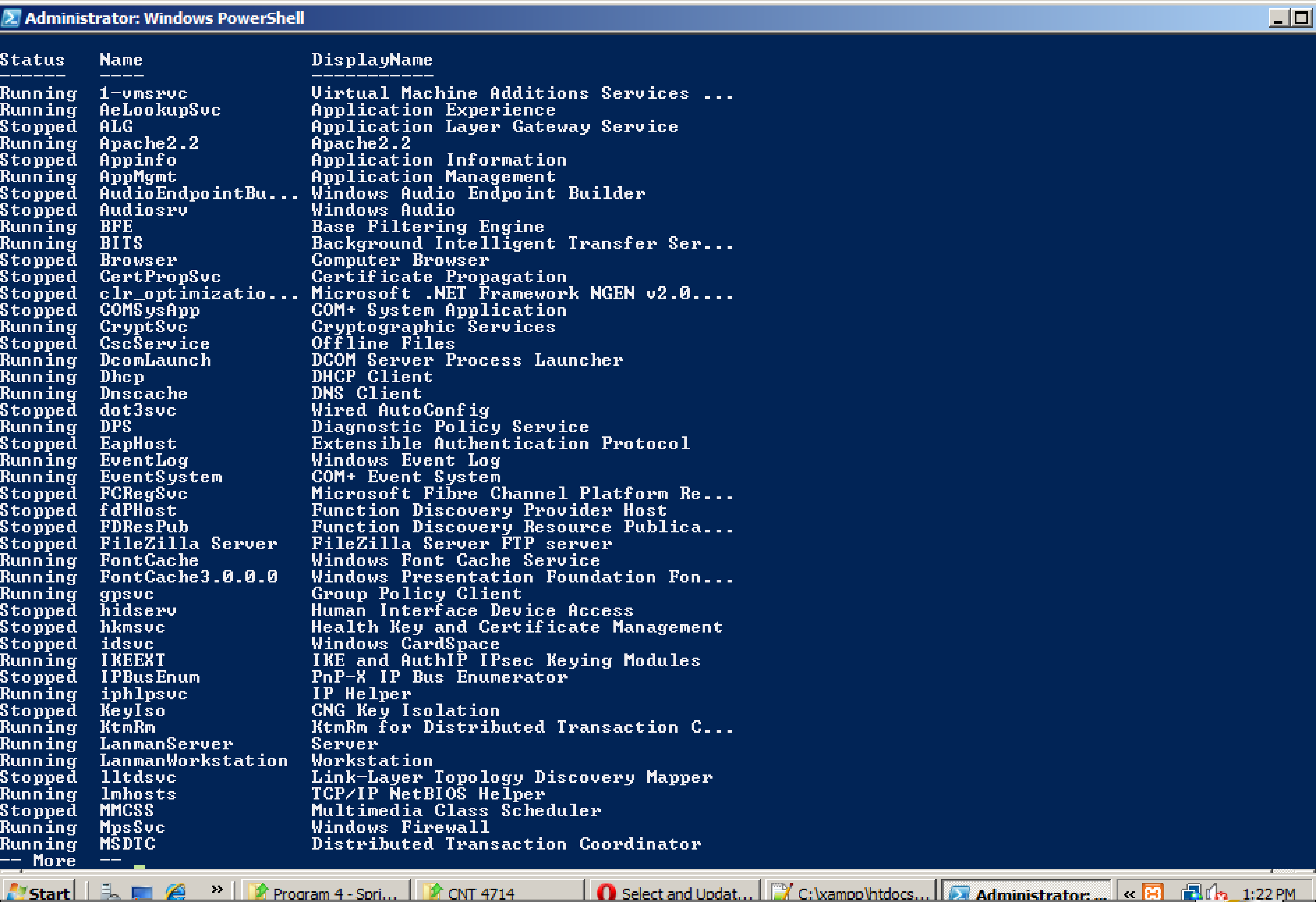


Administrator: Windows PowerShell

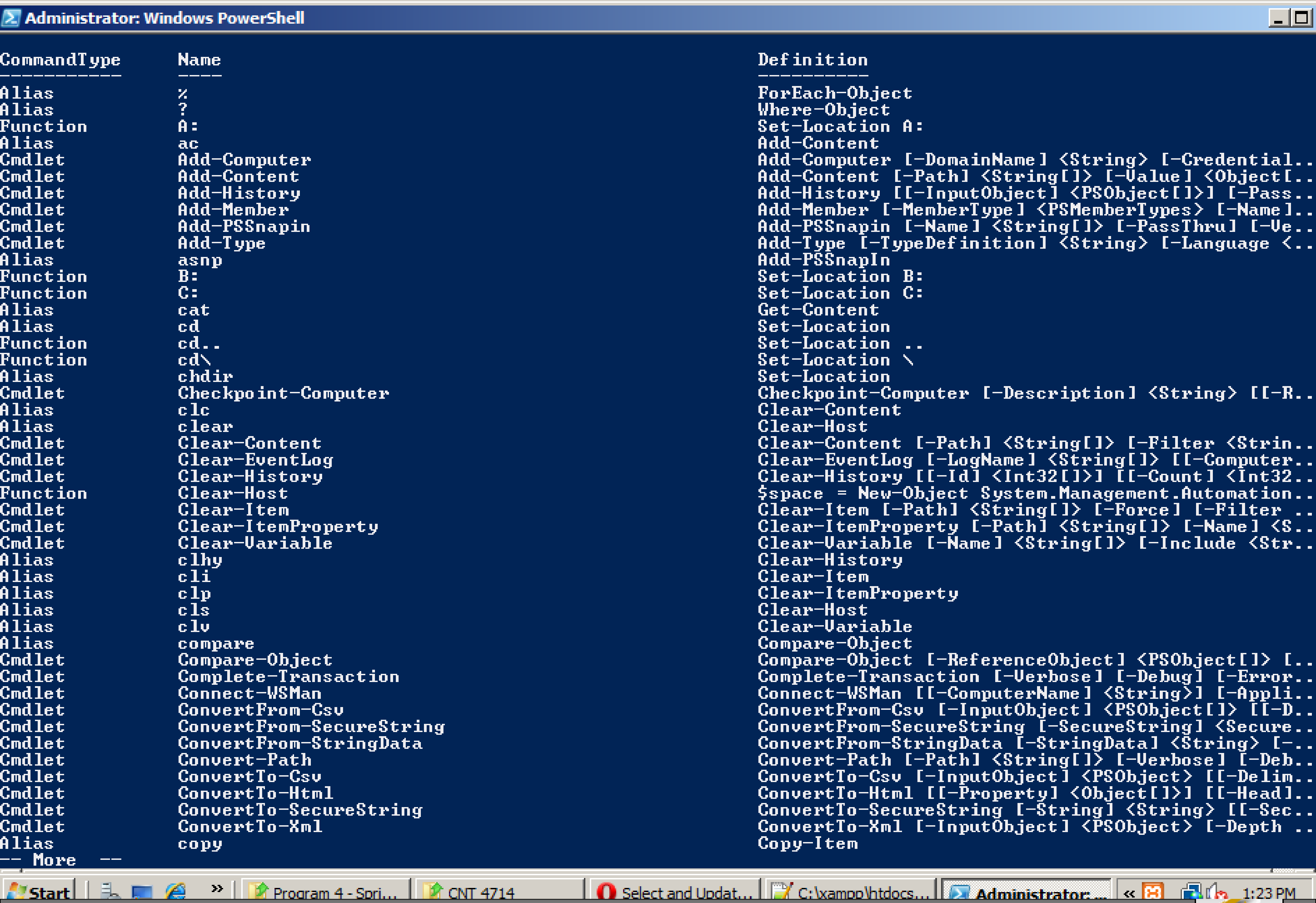
```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> dir | more


    Directory: C:\Users\Administrator


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----         2/16/2011  12:53 PM            .nbi
d----         2/17/2011   1:17 PM            .netbeans
d----         2/16/2011  12:57 PM            .netbeans-registration
d-r--         1/19/2011   3:31 PM            Contacts
d-r--         2/23/2011   4:51 PM            Desktop
d-r--         2/17/2011   2:47 PM            Documents
d-r--         2/24/2011  11:31 AM            Downloads
d-r--         1/19/2011   3:31 PM            Favorites
d-r--         1/19/2011   3:31 PM            Links
d-r--         1/19/2011   3:31 PM            Music
d-r--         1/19/2011   3:31 PM            Pictures
d-r--         1/19/2011   3:31 PM            Saved Games
d-r--         1/19/2011   3:31 PM            Searches
d-r--         1/19/2011   3:31 PM            Videos
-a---         2/14/2011   4:31 PM        723 volshext.log


PS C:\Users\Administrator>
```

Scripting – Windows PowerShell

- To get a listing of the services currently running on your server, enter the command `get-service`, at the command prompt. An example of this is shown on page 17.
- 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 18.


Administrator: Windows PowerShell

```
Status   Name               DisplayName
------   ----               -----------
Running  1-vmsrvc           Virtual Machine Additions Services ...
Running  AeLookupSvc        Application Experience
Stopped  ALG                Application Layer Gateway Service
Running  Apache2.2          Apache2.2
Stopped  Appinfo            Application Information
Running  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  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
Stopped  FDResPub           Function Discovery Resource Publica...
Stopped  FileZilla Server   FileZilla Server FTP server
Running  FontCache          Windows Font Cache Service
Running  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
Stopped  lltdsvc            Link-Layer Topology Discovery Mapper
Running  lmhosts            TCP/IP NetBIOS Helper
Stopped  MMCSS              Multimedia Class Scheduler
Running  MpsSvc             Windows Firewall
Running  MSDTC              Distributed Transaction Coordinator
-- More  --
```



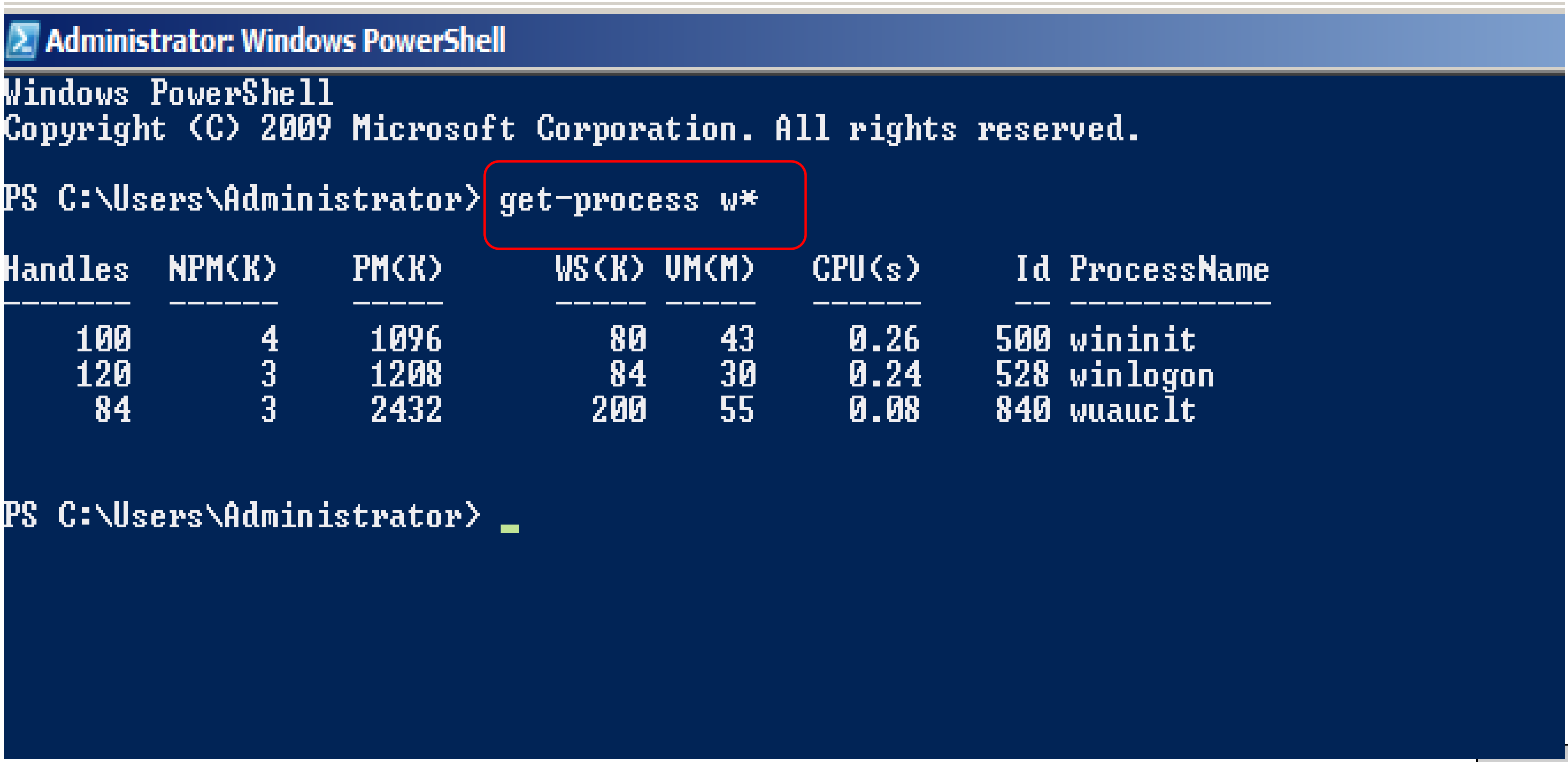
```
CommandType     Name                                                Definition
-----------     ----                                                ----------
Alias           %                                                   ForEach-Object
Alias           ?                                                   Where-Object
Function        A:                                                  Set-Location A:
Alias           ac                                                  Add-Content
Cmdlet          Add-Computer                                        Add-Computer [-DomainName] <String> [-Credential..
Cmdlet          Add-Content                                         Add-Content [-Path] <String[]> [-Value] <Object[..
Cmdlet          Add-History                                         Add-History [[-InputObject] <PSObject[]>] [-Pass..
Cmdlet          Add-Member                                          Add-Member [-MemberType] <PSMemberTypes> [-Name]..
Cmdlet          Add-PSSnapin                                        Add-PSSnapin [-Name] <String[]> [-PassThru] [-Ve..
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> [[-R..
Alias           clc                                                 Clear-Content
Alias           clear                                               Clear-Host
Cmdlet          Clear-Content                                       Clear-Content [-Path] <String[]> [-Filter <Strin..
Cmdlet          Clear-EventLog                                      Clear-EventLog [-LogName] <String[]> [[-Computer..
Cmdlet          Clear-History                                       Clear-History [[-Id] <Int32[]>] [[-Count] <Int32..
Function        Clear-Host                                          $space = New-Object System.Management.Automation..
Cmdlet          Clear-Item                                          Clear-Item [-Path] <String[]> [-Force] [-Filter ..
Cmdlet          Clear-ItemProperty                                  Clear-ItemProperty [-Path] <String[]> [-Name] <S..
Cmdlet          Clear-Variable                                      Clear-Variable [-Name] <String[]> [-Include <Str..
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] [-Error..
Cmdlet          Connect-WSMan                                       Connect-WSMan [[-ComputerName] <String>] [-Appli..
Cmdlet          ConvertFrom-Csv                                     ConvertFrom-Csv [-InputObject] <PSObject[]> [[-D..
Cmdlet          ConvertFrom-SecureString                            ConvertFrom-SecureString [-SecureString] <Secure..
Cmdlet          ConvertFrom-StringData                              ConvertFrom-StringData [-StringData] <String> [-..
Cmdlet          Convert-Path                                        Convert-Path [-Path] <String[]> [-Verbose] [-Deb..
Cmdlet          ConvertTo-Csv                                       ConvertTo-Csv [-InputObject] <PSObject> [[-Delim..
Cmdlet          ConvertTo-Html                                      ConvertTo-Html [[-Property] <Object[]>] [[-Head]..
Cmdlet          ConvertTo-SecureString                              ConvertTo-SecureString [-String] <String> [[-Sec..
Cmdlet          ConvertTo-Xml                                       ConvertTo-Xml [-InputObject] <PSObject> [-Depth ..
Alias           copy                                                Copy-Item
-- More  --
```

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.

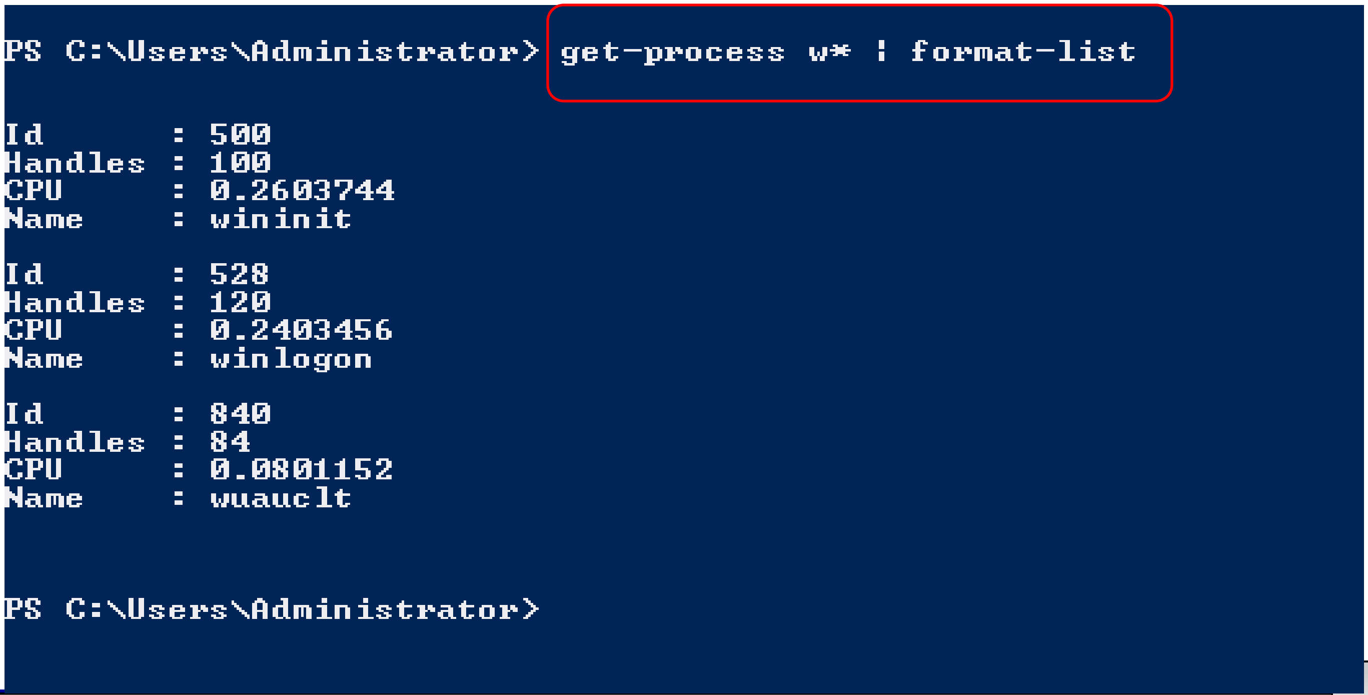


Administrator: Windows PowerShell

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

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

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
    100       4     1096         80    43     0.26    500 wininit
    120       3     1208         84    30     0.24    528 winlogon
     84       3     2432        200    55     0.08    840 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.

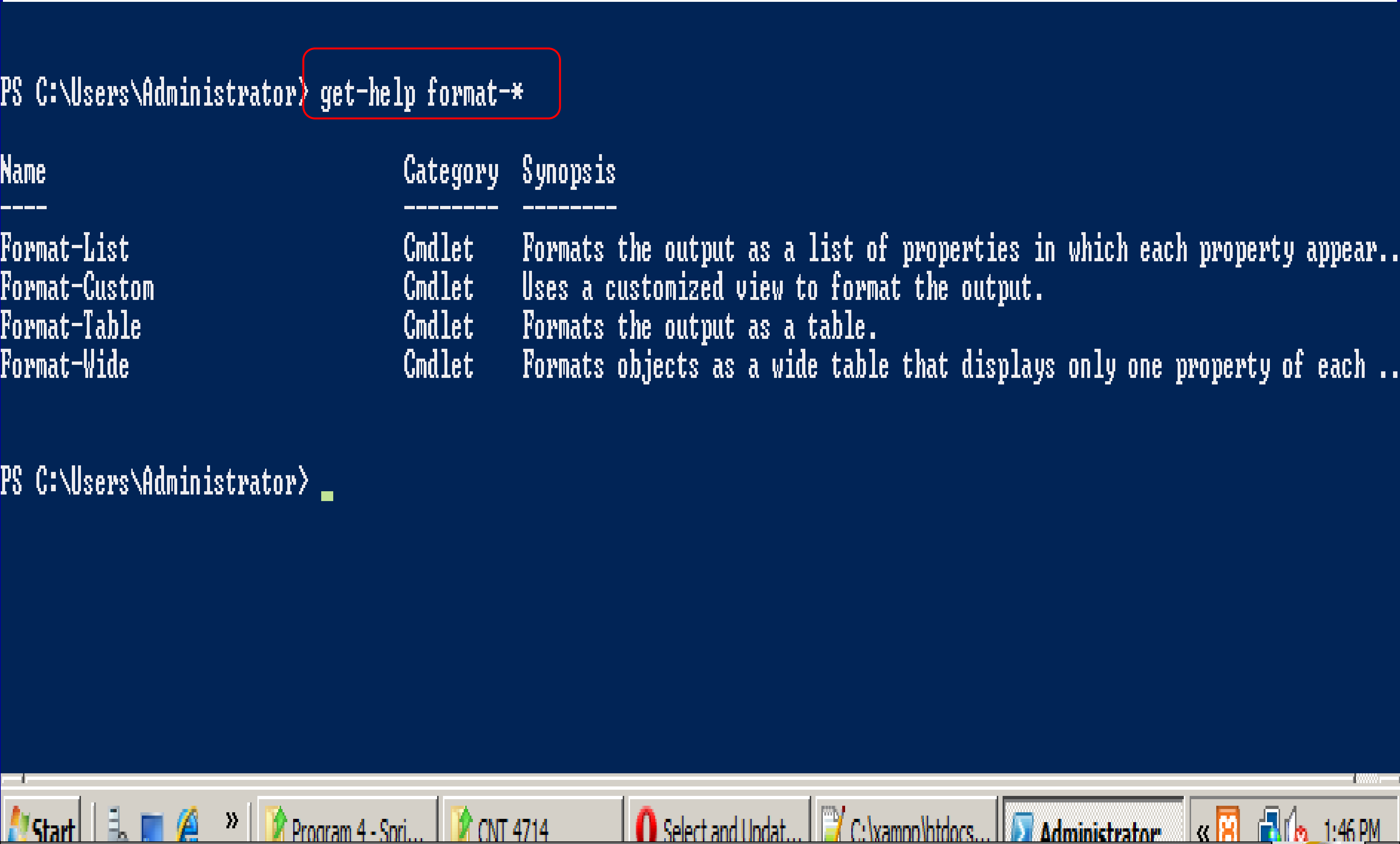


```
PS C:\Users\Administrator> get-process w* | format-list

Id      : 500
Handles : 100
CPU     : 0.2603744
Name    : wininit

Id      : 528
Handles : 120
CPU     : 0.2403456
Name    : winlogon

Id      : 840
Handles : 84
CPU     : 0.0801152
Name    : wuauclt


PS C:\Users\Administrator>
```

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.

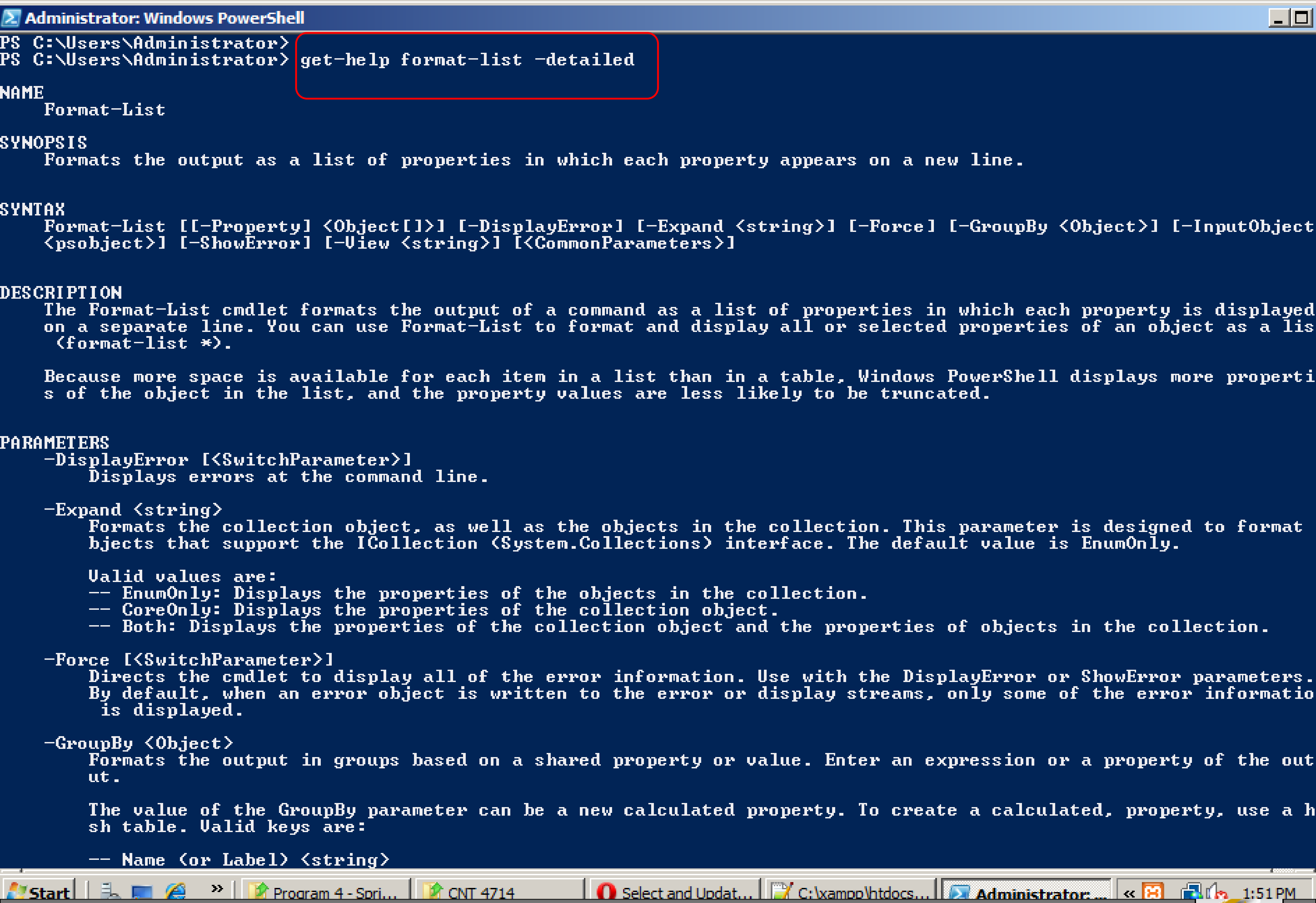
Scripting – Windows PowerShell



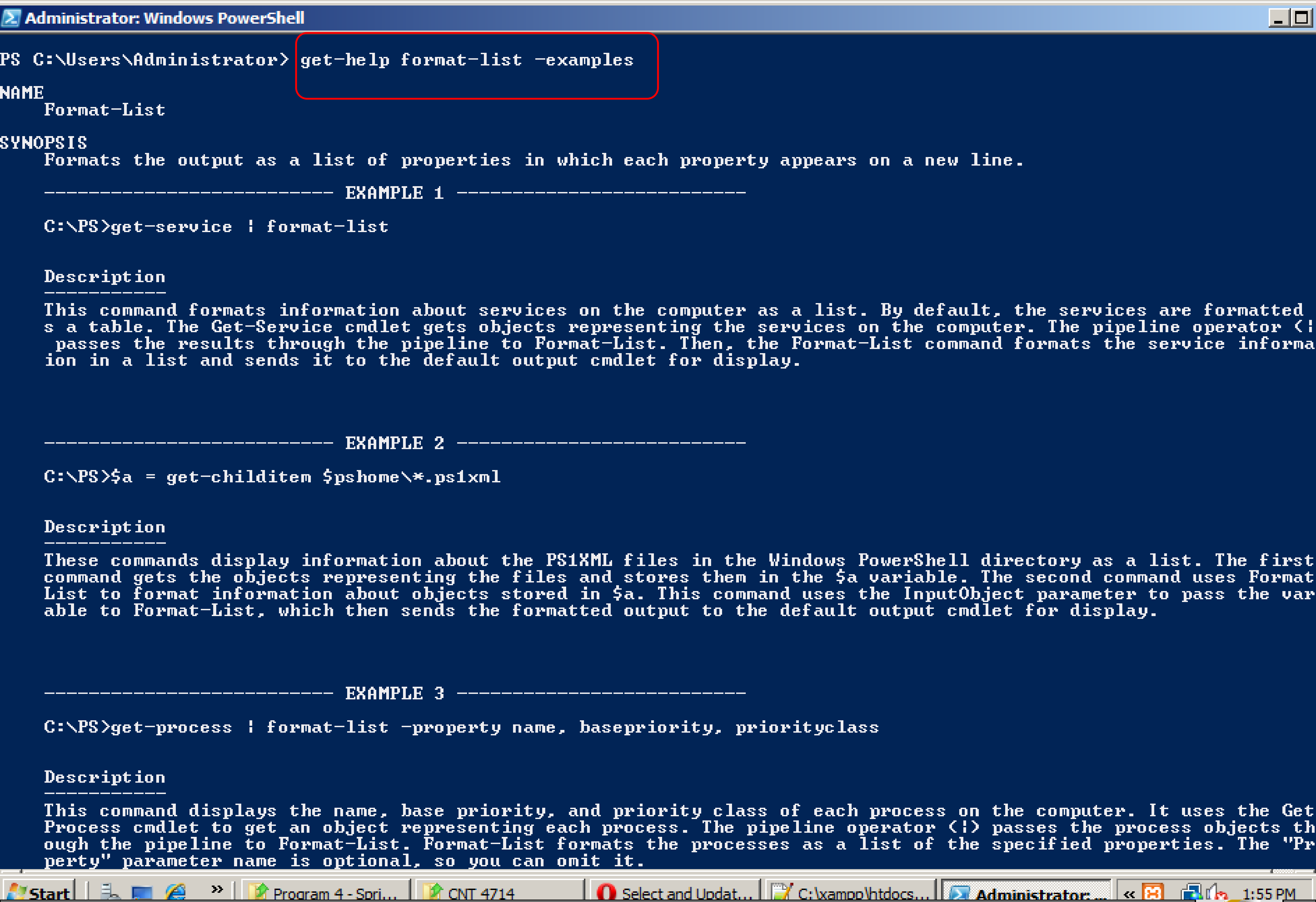
```
PS C:\Users\Administrator> get-help format-*

Name                              Category  Synopsis
----                              --------  --------
Format-List                       Cmdlet    Formats the output as a list of properties in which each property appear..
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 each ..


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

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.



```
PS C:\Users\Administrator>
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>] [-InputObject
    <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 displayed
    on a separate line. You can use Format-List to format and display all or selected properties of an object as a lis
     (format-list *).

    Because more space is available for each item in a list than in a table, Windows PowerShell displays more properti
    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 format
        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 parameters.
        By default, when an error object is written to the error or display streams, only some of the error informatio
         is displayed.

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

        The value of the GroupBy parameter can be a new calculated property. To create a calculated, property, use a h
        sh table. Valid keys are:

        -- Name (or Label) <string>
```

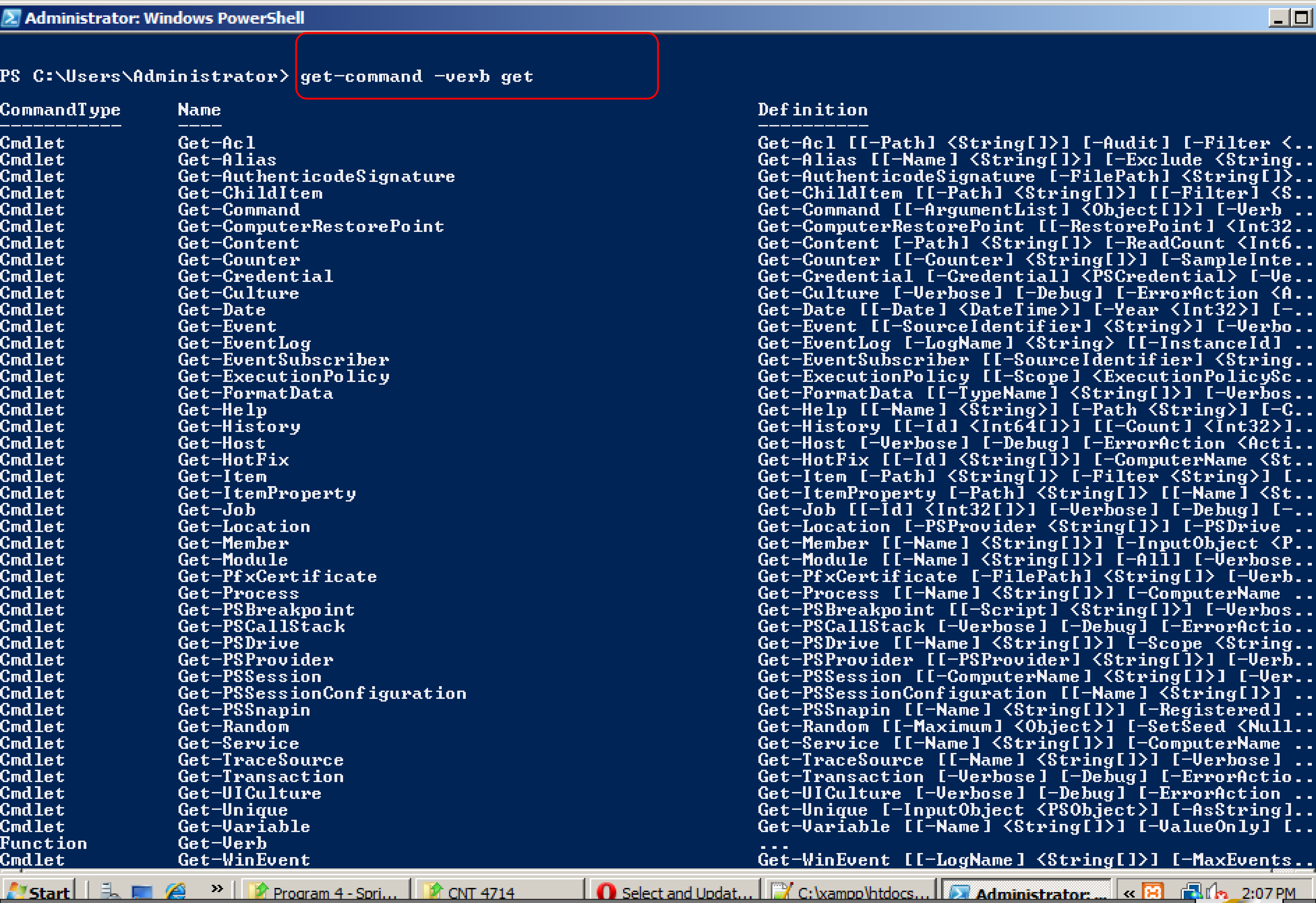


```
Administrator: Windows PowerShell

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 formatted
    s a table. The Get-Service cmdlet gets objects representing the services on the computer. The pipeline operator (|
     passes the results through the pipeline to Format-List. Then, the Format-List command formats the service informa
    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 first
    command gets the objects representing the files and stores them in the $a variable. The second command uses Format
    List to format information about objects stored in $a. This command uses the InputObject parameter to pass the var
    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
    -----------
    This command displays the name, base priority, and priority class of each process on the computer. It uses the Get
    Process cmdlet to get an object representing each process. The pipeline operator (|) passes the process objects th
    ough the pipeline to Format-List. Format-List formats the processes as a list of the specified properties. The "Pr
    perty" parameter name is optional, so you can omit it.
```

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`.

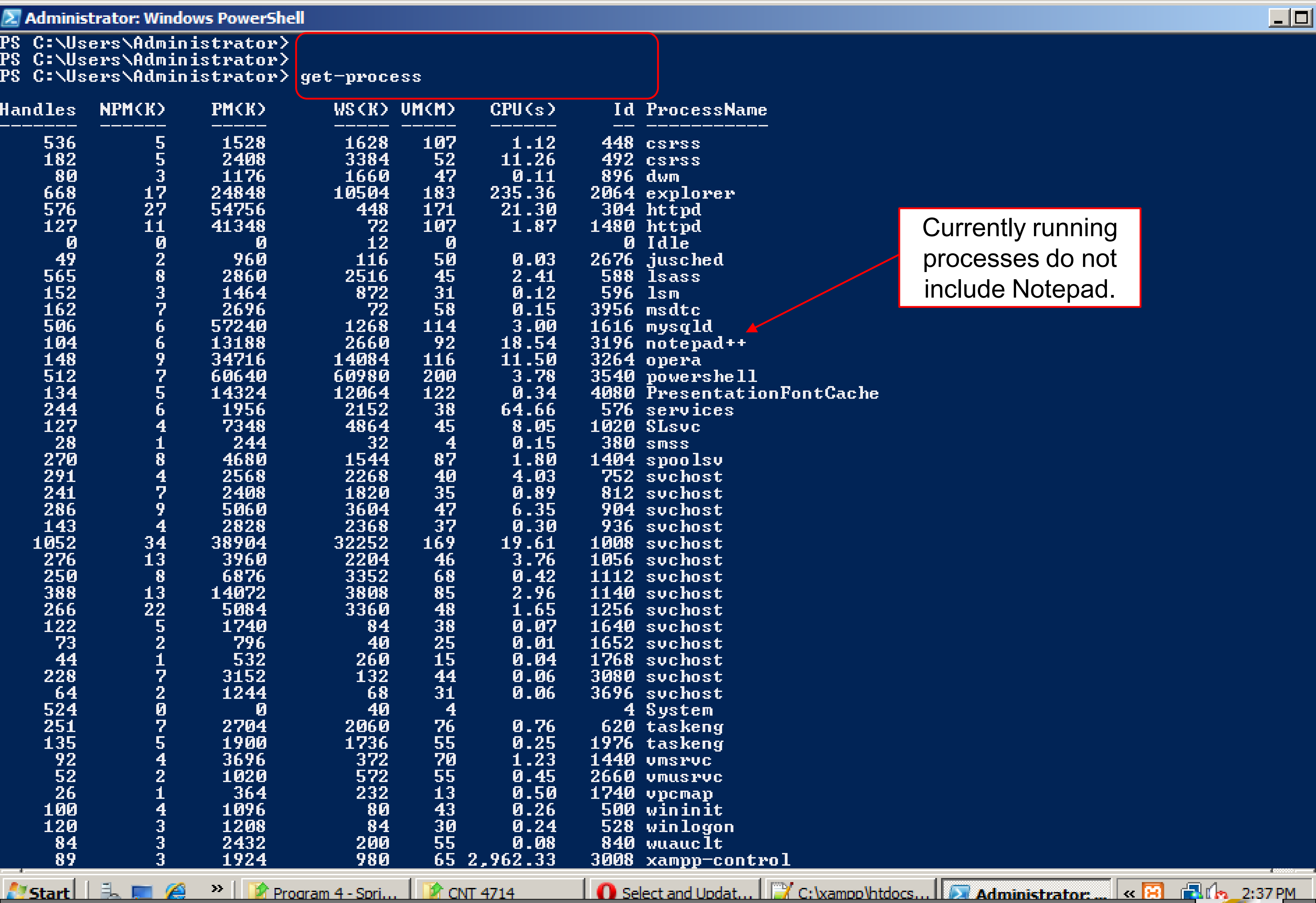
- The output of this command is shown on the next page, but you might want to experiment a bit and try out some other options. For example, try listing all of the commands that use the verbs `add` or `new`.



```
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 <String..
Cmdlet          Get-AuthenticodeSignature         Get-AuthenticodeSignature [-FilePath] <String[]>..
Cmdlet          Get-ChildItem                     Get-ChildItem [[-Path] <String[]>] [[-Filter] <S..
Cmdlet          Get-Command                       Get-Command [[-ArgumentList] <Object[]>] [-Verb ..
Cmdlet          Get-ComputerRestorePoint          Get-ComputerRestorePoint [[-RestorePoint] <Int32..
Cmdlet          Get-Content                       Get-Content [-Path] <String[]> [-ReadCount <Int6..
Cmdlet          Get-Counter                       Get-Counter [[-Counter] <String[]>] [-SampleInte..
Cmdlet          Get-Credential                    Get-Credential [-Credential] <PSCredential> [-Ve..
Cmdlet          Get-Culture                       Get-Culture [-Verbose] [-Debug] [-ErrorAction <A..
Cmdlet          Get-Date                          Get-Date [[-Date] <DateTime>] [-Year <Int32>] [-..
Cmdlet          Get-Event                         Get-Event [[-SourceIdentifier] <String>] [-Verbo..
Cmdlet          Get-EventLog                      Get-EventLog [-LogName] <String> [[-InstanceId] ..
Cmdlet          Get-EventSubscriber               Get-EventSubscriber [[-SourceIdentifier] <String..
Cmdlet          Get-ExecutionPolicy               Get-ExecutionPolicy [[-Scope] <ExecutionPolicySc..
Cmdlet          Get-FormatData                    Get-FormatData [[-TypeName] <String[]>] [-Verbos..
Cmdlet          Get-Help                          Get-Help [[-Name] <String>] [-Path <String>] [-C..
Cmdlet          Get-History                       Get-History [[-Id] <Int64[]>] [[-Count] <Int32>]..
Cmdlet          Get-Host                          Get-Host [-Verbose] [-Debug] [-ErrorAction <Acti..
Cmdlet          Get-HotFix                        Get-HotFix [[-Id] <String[]>] [-ComputerName <St..
Cmdlet          Get-Item                          Get-Item [-Path] <String[]> [-Filter <String>] [..
Cmdlet          Get-ItemProperty                  Get-ItemProperty [-Path] <String[]> [[-Name] <St..
Cmdlet          Get-Job                           Get-Job [[-Id] <Int32[]>] [-Verbose] [-Debug] [-..
Cmdlet          Get-Location                      Get-Location [-PSProvider <String[]>] [-PSDrive ..
Cmdlet          Get-Member                        Get-Member [[-Name] <String[]>] [-InputObject <P..
Cmdlet          Get-Module                        Get-Module [[-Name] <String[]>] [-All] [-Verbose..
Cmdlet          Get-PfxCertificate                Get-PfxCertificate [-FilePath] <String[]> [-Verb..
Cmdlet          Get-Process                       Get-Process [[-Name] <String[]>] [-ComputerName ..
Cmdlet          Get-PSBreakpoint                  Get-PSBreakpoint [[-Script] <String[]>] [-Verbos..
Cmdlet          Get-PSCallStack                   Get-PSCallStack [-Verbose] [-Debug] [-ErrorActio..
Cmdlet          Get-PSDrive                       Get-PSDrive [[-Name] <String[]>] [-Scope <String..
Cmdlet          Get-PSProvider                    Get-PSProvider [[-PSProvider] <String[]>] [-Verb..
Cmdlet          Get-PSSession                     Get-PSSession [[-ComputerName] <String[]>] [-Ver..
Cmdlet          Get-PSSessionConfiguration        Get-PSSessionConfiguration [[-Name] <String[]>] ..
Cmdlet          Get-PSSnapin                      Get-PSSnapin [[-Name] <String[]>] [-Registered] ..
Cmdlet          Get-Random                        Get-Random [[-Maximum] <Object>] [-SetSeed <Null..
Cmdlet          Get-Service                       Get-Service [[-Name] <String[]>] [-ComputerName ..
Cmdlet          Get-TraceSource                   Get-TraceSource [[-Name] <String[]>] [-Verbose] ..
Cmdlet          Get-Transaction                   Get-Transaction [-Verbose] [-Debug] [-ErrorActio..
Cmdlet          Get-UICulture                     Get-UICulture [-Verbose] [-Debug] [-ErrorAction ..
Cmdlet          Get-Unique                        Get-Unique [-InputObject <PSObject>] [-AsString]..
Cmdlet          Get-Variable                      Get-Variable [[-Name] <String[]>] [-ValueOnly] [..
Function        Get-Verb                          ...
Cmdlet          Get-WinEvent                      Get-WinEvent [[-LogName] <String[]>] [-MaxEvents..
```

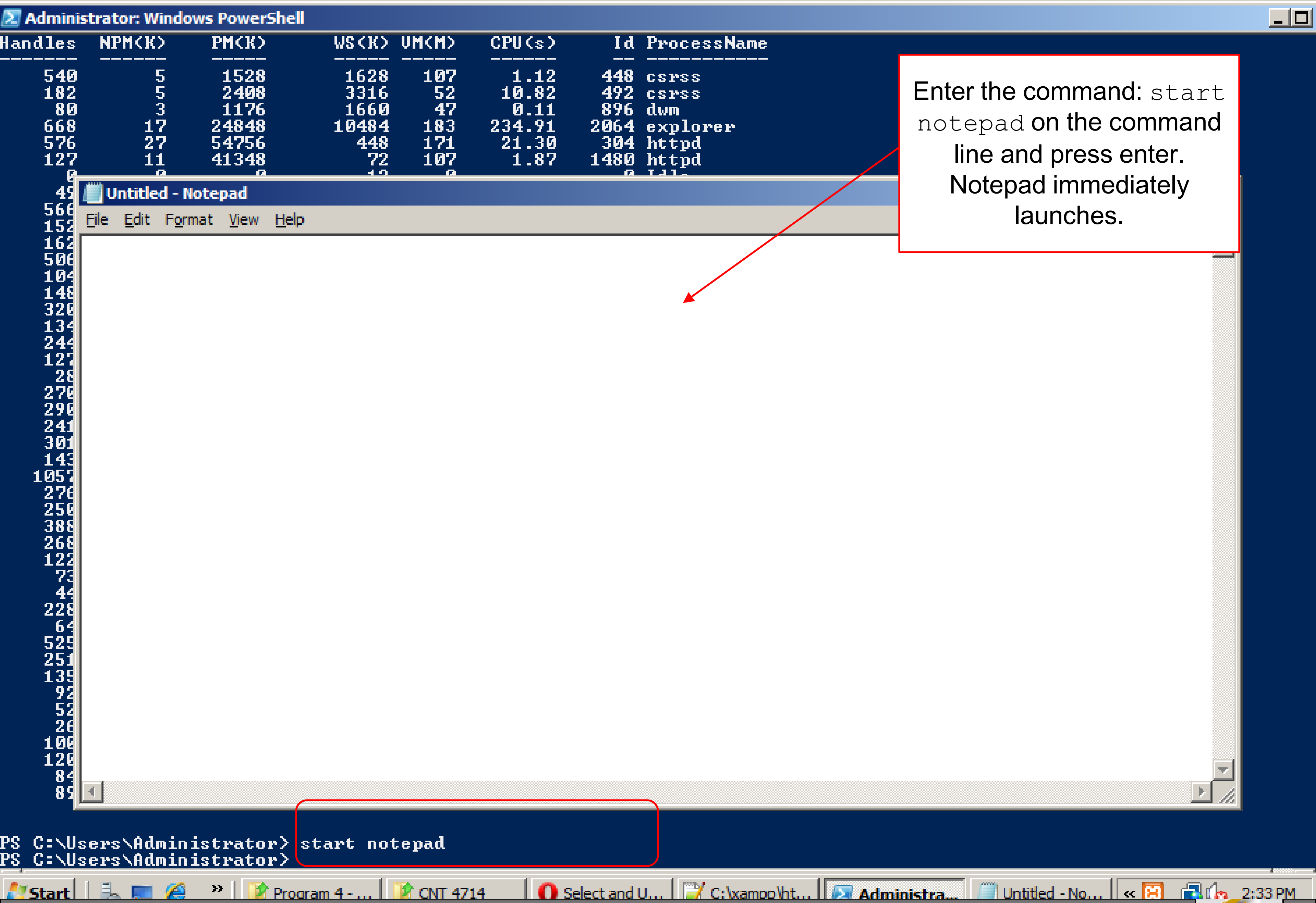
Scripting – Windows PowerShell

- 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).



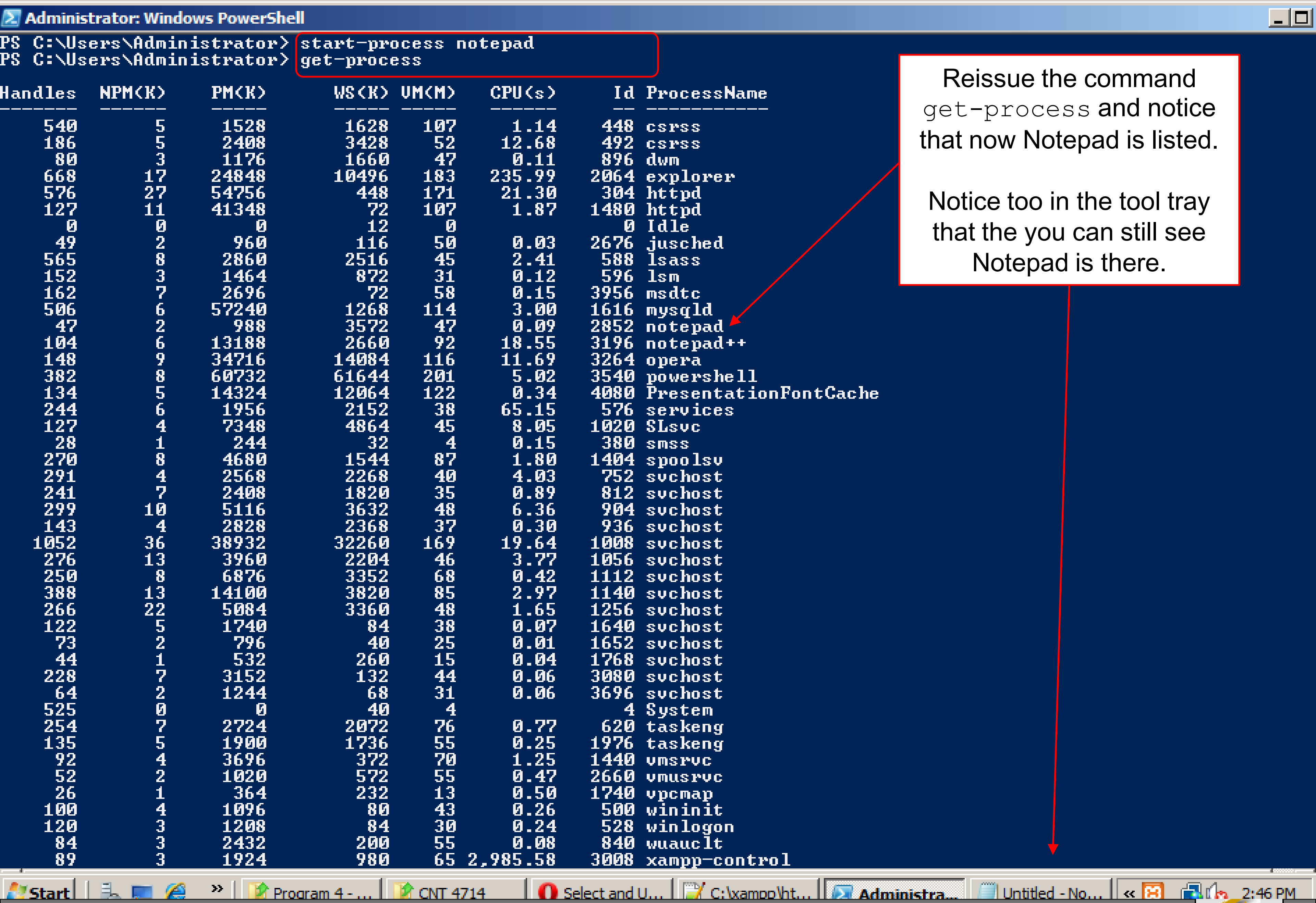
```
PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator> get-process

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
    536       5     1528       1628   107     1.12    448 csrss
    182       5     2408       3384    52    11.26    492 csrss
     80       3     1176       1660    47     0.11    896 dwm
    668      17    24848      10504   183   235.36   2064 explorer
    576      27    54756        448   171    21.30    304 httpd
    127      11    41348         72   107     1.87   1480 httpd
      0       0        0         12     0               0 Idle
     49       2      960        116    50     0.03   2676 jusched
    565       8     2860       2516    45     2.41    588 lsass
    152       3     1464        872    31     0.12    596 lsm
    162       7     2696         72    58     0.15   3956 msdtc
    506       6    57240       1268   114     3.00   1616 mysqld
    104       6    13188       2660    92    18.54   3196 notepad++
    148       9    34716      14084   116    11.50   3264 opera
    512       7    60640      60980   200     3.78   3540 powershell
    134       5    14324      12064   122     0.34   4080 PresentationFontCache
    244       6     1956       2152    38    64.66    576 services
    127       4     7348       4864    45     8.05   1020 SLsvc
     28       1      244         32     4     0.15    380 smss
    270       8     4680       1544    87     1.80   1404 spoolsv
    291       4     2568       2268    40     4.03    752 svchost
    241       7     2408       1820    35     0.89    812 svchost
    286       9     5060       3604    47     6.35    904 svchost
    143       4     2828       2368    37     0.30    936 svchost
   1052      34    38904      32252   169    19.61   1008 svchost
    276      13     3960       2204    46     3.76   1056 svchost
    250       8     6876       3352    68     0.42   1112 svchost
    388      13    14072       3808    85     2.96   1140 svchost
    266      22     5084       3360    48     1.65   1256 svchost
    122       5     1740         84    38     0.07   1640 svchost
     73       2      796         40    25     0.01   1652 svchost
     44       1      532        260    15     0.04   1768 svchost
    228       7     3152        132    44     0.06   3080 svchost
     64       2     1244         68    31     0.06   3696 svchost
    524       0        0         40     4               4 System
    251       7     2704       2060    76     0.76    620 taskeng
    135       5     1900       1736    55     0.25   1976 taskeng
     92       4     3696        372    70     1.23   1440 vmsrvc
     52       2     1020        572    55     0.45   2660 vmusrvc
     26       1      364        232    13     0.50   1740 vpcmap
    100       4     1096         80    43     0.26    500 wininit
    120       3     1208         84    30     0.24    528 winlogon
     84       3     2432        200    55     0.08    840 wuauclt
     89       3     1924        980    65 2,962.33   3008 xampp-control
```

Currently running processes do not include Notepad.



```
Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
    540       5     1528       1628   107     1.12    448 csrss
    182       5     2408       3316    52    10.82    492 csrss
     80       3     1176       1660    47     0.11    896 dwm
    668      17    24848      10484   183   234.91   2064 explorer
    576      27    54756        448   171    21.30    304 httpd
    127      11    41348         72   107     1.87   1480 httpd
      0       0        0         12     0              0 Idle
     49
    566
    152
    162
    506
    104
    148
    320
    134
    244
    127
     28
    270
    290
    241
    301
    143
   1057
    276
    250
    388
    268
    122
     73
     44
    228
     64
    525
    251
    135
     92
     52
     26
    100
    120
     84
     89

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

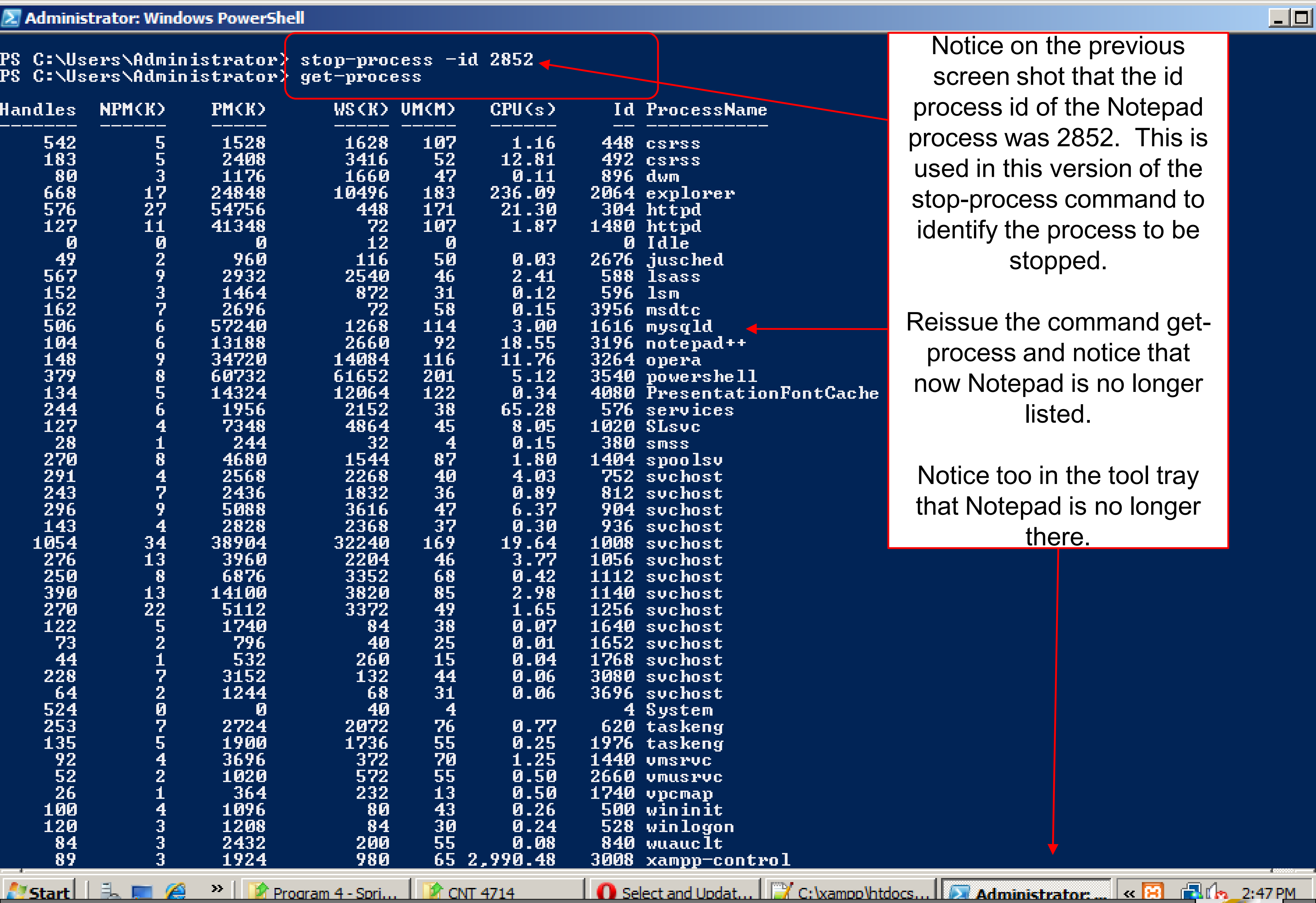
Enter the command: `start notepad` on the command line and press enter. Notepad immediately launches.


Administrator: Windows PowerShell

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

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
    540       5     1528       1628   107     1.14    448 csrss
    186       5     2408       3428    52    12.68    492 csrss
     80       3     1176       1660    47     0.11    896 dwm
    668      17    24848      10496   183   235.99   2064 explorer
    576      27    54756        448   171    21.30    304 httpd
    127      11    41348         72   107     1.87   1480 httpd
      0       0        0         12     0               0 Idle
     49       2      960        116    50     0.03   2676 jusched
    565       8     2860       2516    45     2.41    588 lsass
    152       3     1464        872    31     0.12    596 lsm
    162       7     2696         72    58     0.15   3956 msdtc
    506       6    57240       1268   114     3.00   1616 mysqld
     47       2      988       3572    47     0.09   2852 notepad
    104       6    13188       2660    92    18.55   3196 notepad++
    148       9    34716      14084   116    11.69   3264 opera
    382       8    60732      61644   201     5.02   3540 powershell
    134       5    14324      12064   122     0.34   4080 PresentationFontCache
    244       6     1956       2152    38    65.15    576 services
    127       4     7348       4864    45     8.05   1020 SLsvc
     28       1      244         32     4     0.15    380 smss
    270       8     4680       1544    87     1.80   1404 spoolsv
    291       4     2568       2268    40     4.03    752 svchost
    241       7     2408       1820    35     0.89    812 svchost
    299      10     5116       3632    48     6.36    904 svchost
    143       4     2828       2368    37     0.30    936 svchost
   1052      36    38932      32260   169    19.64   1008 svchost
    276      13     3960       2204    46     3.77   1056 svchost
    250       8     6876       3352    68     0.42   1112 svchost
    388      13    14100       3820    85     2.97   1140 svchost
    266      22     5084       3360    48     1.65   1256 svchost
    122       5     1740         84    38     0.07   1640 svchost
     73       2      796         40    25     0.01   1652 svchost
     44       1      532        260    15     0.04   1768 svchost
    228       7     3152        132    44     0.06   3080 svchost
     64       2     1244         68    31     0.06   3696 svchost
    525       0        0         40     4               4 System
    254       7     2724       2072    76     0.77    620 taskeng
    135       5     1900       1736    55     0.25   1976 taskeng
     92       4     3696        372    70     1.25   1440 vmsrvc
     52       2     1020        572    55     0.47   2660 vmusrvc
     26       1      364        232    13     0.50   1740 vpcmap
    100       4     1096         80    43     0.26    500 wininit
    120       3     1208         84    30     0.24    528 winlogon
     84       3     2432        200    55     0.08    840 wuauclt
     89       3     1924        980    65 2,985.58   3008 xampp-control
```

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.



```
PS C:\Users\Administrator> stop-process -id 2852
PS C:\Users\Administrator> get-process

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
    542       5     1528       1628   107     1.16    448 csrss
    183       5     2408       3416    52    12.81    492 csrss
     80       3     1176       1660    47     0.11    896 dwm
    668      17    24848      10496   183   236.09   2064 explorer
    576      27    54756        448   171    21.30    304 httpd
    127      11    41348         72   107     1.87   1480 httpd
      0       0        0         12     0              0 Idle
     49       2      960        116    50     0.03   2676 jusched
    567       9     2932       2540    46     2.41    588 lsass
    152       3     1464        872    31     0.12    596 lsm
    162       7     2696         72    58     0.15   3956 msdtc
    506       6    57240       1268   114     3.00   1616 mysqld
    104       6    13188       2660    92    18.55   3196 notepad++
    148       9    34720      14084   116    11.76   3264 opera
    379       8    60732      61652   201     5.12   3540 powershell
    134       5    14324      12064   122     0.34   4080 PresentationFontCache
    244       6     1956       2152    38    65.28    576 services
    127       4     7348       4864    45     8.05   1020 SLsvc
     28       1      244         32     4     0.15    380 smss
    270       8     4680       1544    87     1.80   1404 spoolsv
    291       4     2568       2268    40     4.03    752 svchost
    243       7     2436       1832    36     0.89    812 svchost
    296       9     5088       3616    47     6.37    904 svchost
    143       4     2828       2368    37     0.30    936 svchost
   1054      34    38904      32240   169    19.64   1008 svchost
    276      13     3960       2204    46     3.77   1056 svchost
    250       8     6876       3352    68     0.42   1112 svchost
    390      13    14100       3820    85     2.98   1140 svchost
    270      22     5112       3372    49     1.65   1256 svchost
    122       5     1740         84    38     0.07   1640 svchost
     73       2      796         40    25     0.01   1652 svchost
     44       1      532        260    15     0.04   1768 svchost
    228       7     3152        132    44     0.06   3080 svchost
     64       2     1244         68    31     0.06   3696 svchost
    524       0        0         40     4              4 System
    253       7     2724       2072    76     0.77    620 taskeng
    135       5     1900       1736    55     0.25   1976 taskeng
     92       4     3696        372    70     1.25   1440 vmsrvc
     52       2     1020        572    55     0.50   2660 vmusrvc
     26       1      364        232    13     0.50   1740 vpcmap
    100       4     1096         80    43     0.26    500 wininit
    120       3     1208         84    30     0.24    528 winlogon
     84       3     2432        200    55     0.08    840 wuauclt
     89       3     1924        980    65 2,990.48   3008 xampp-control
```

Notice on the previous screen shot that the id process id of the Notepad process was 2852. 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.

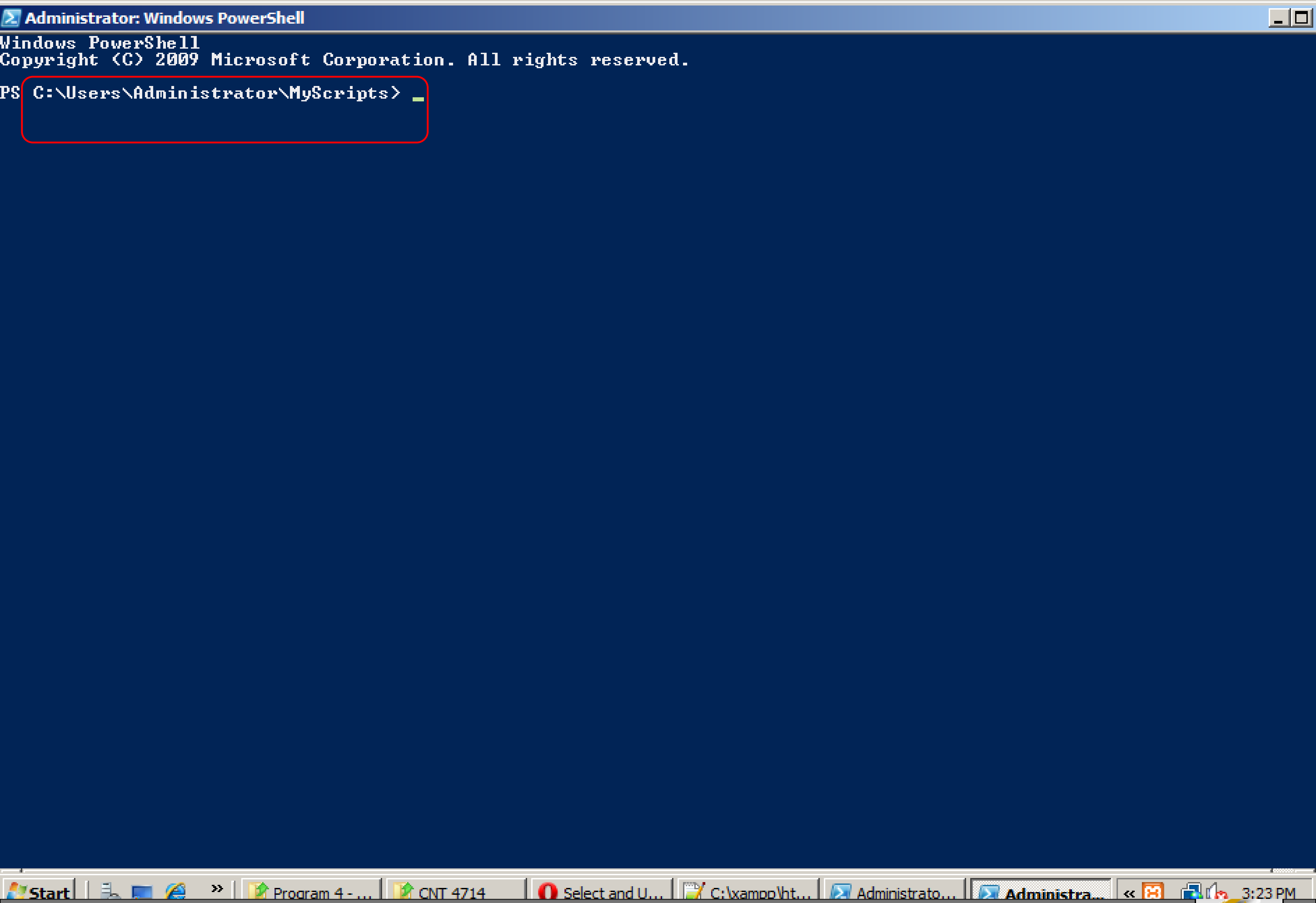
Notice too in the tool tray that Notepad is no longer there.

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 desktop.

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.

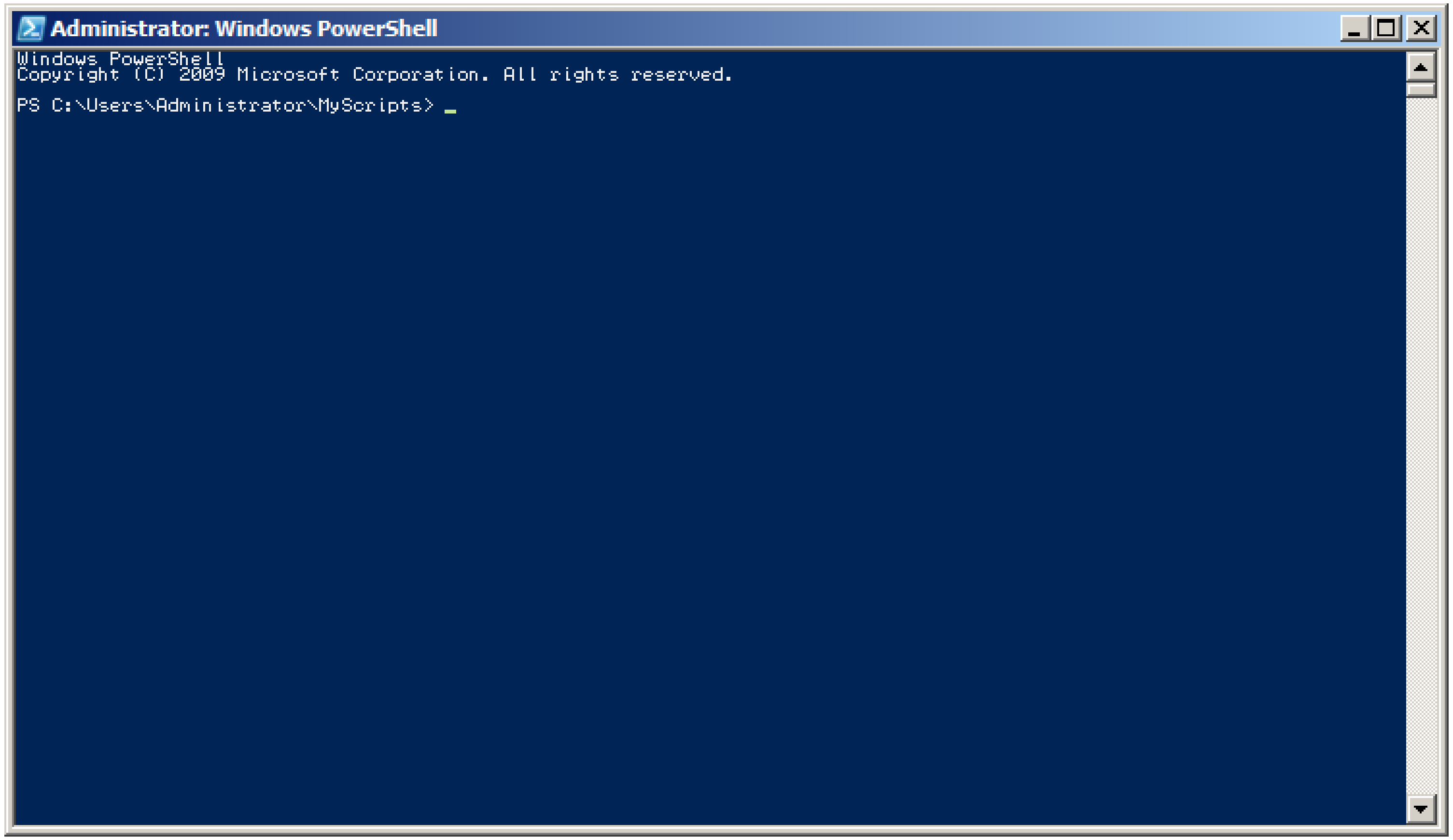


```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

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

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.

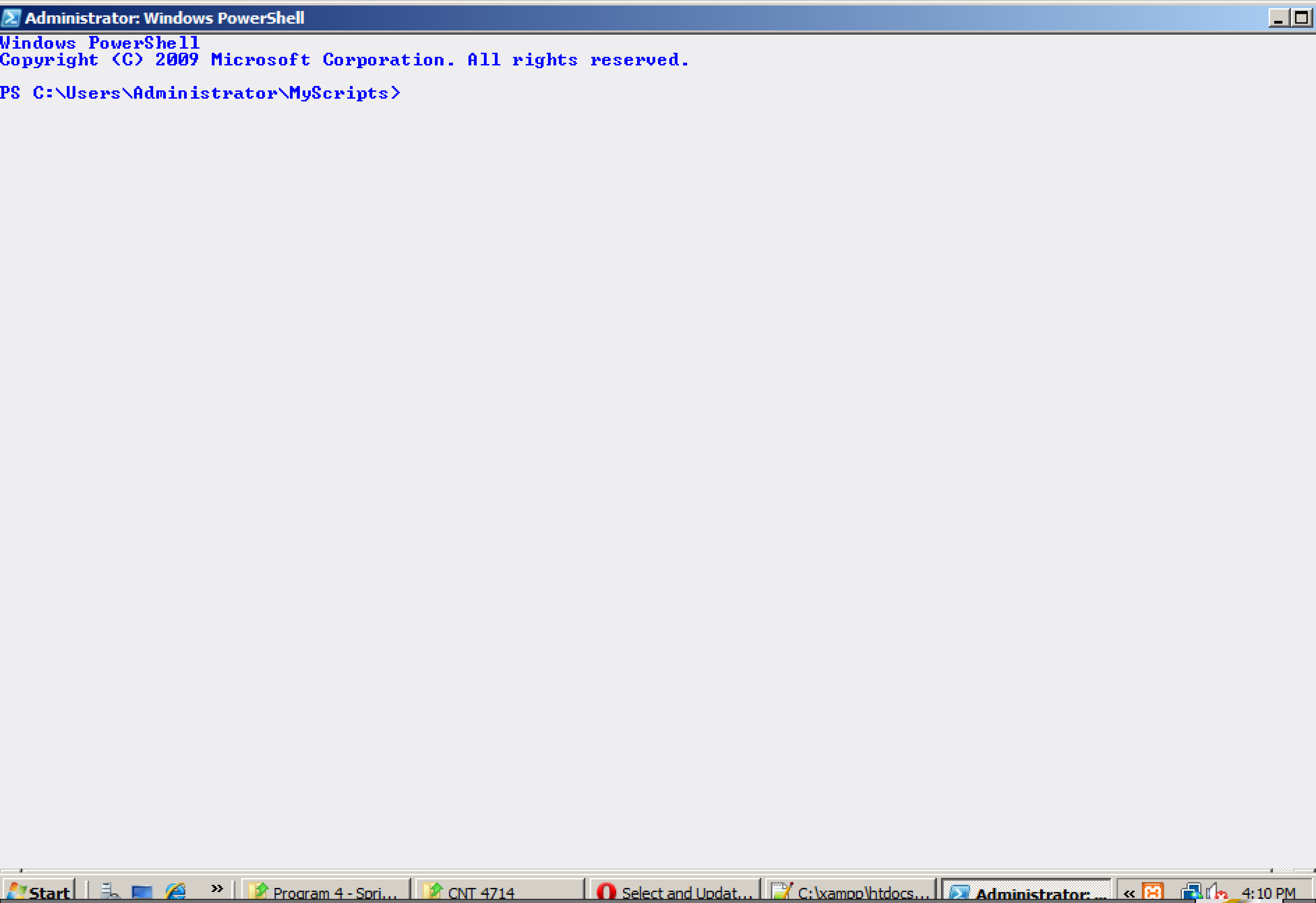
Scripting – Windows PowerShell



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

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

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.



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

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