Module 1: Getting Started

1. First, run Update-Help and ensure it completes without errors. That will get a copy of the help on your local computer. This requires an Internet connection, and requires that the shell be running under elevated privileges (which means it must say “Administrator” in the shell’s window title bar).

|  |
| --- |
| Update-Help  or if you run it more than once in a single day:  Update-Help –force |

1. Can you find any cmdlets capable of converting other cmdlets’ output into HTML?

|  |
| --- |
| Help html  Or you could try with Get-Command  get-command -noun html |
| Yes, we can convert the output of the other cmdlets into HTML using the following command:   * ConvertTo-HTML |

1. Are there any cmdlets that can redirect output into a file, or to a printer?

|  |
| --- |
| get-command -noun file, printer  Yes, there are cmdlets that can be used to re-direct the output into a file or to a printer. The cmdlets are as follows:   1. Out-File 2. Out-Printer 3. Unblock-File |

1. How many cmdlets are available for working with processes? (Hint: remember that cmdlets all use a singular noun.)

|  |
| --- |
| Get-command –noun process  Or  Help \*Process  There are 5 cmdlets available for working with the processes. They are:   1. Debug-Process 2. Get-Process 3. Start-Process 4. Stop-Process 5. Wait-Process |

1. What cmdlet might you use to write to an event log?

|  |
| --- |
| get-command -verb write -noun eventlog  or if you weren’t sure about the noun, use a wildcard  help \*log   * We use the “Write-Eventlog” cmdlet to write to an event log. |

1. You’ve learned that aliases are nicknames for cmdlets; what cmdlets are available to create, modify, export, or import aliases?

|  |
| --- |
| Help \*alias  Or  get-command -noun alias  There are 5 cmdlets that are available to create, modify, export and import the alias. They are:   * Export-alias * Import-alias * Get-alias * New-alias * Set-alias |

1. Is there a way to keep a transcript of everything you type in the shell, and save that transcript to a text file?

|  |
| --- |
| Help transcript  Yes, we can keep a transcript of everything we type in the shell and save the transcript to a text file.  We have two cmdlets in order to perform this task:   * Start-Transcript <location where the text file needs to be stored>   “ By running the above cmdlet the transcript starts and whatever we type and the resultant output is saved to a text file in the desired location”   * Stop-Transcript   “This cmdlet is used to stop the transcript and save the text file in the location specified” |

1. It can take a long time to retrieve all of the entries from the Security event log. How can you get just the 100 most recent entries?

|  |
| --- |
| help Get-EventLog -Parameter Newest  We can get the event logs of the 100 most recent entries by using the following cmdlet:   * Get-EventLog –Newest 100   After the above command is executed we need to enter the LogName for which we need the information. Ex. System, Security etc. |

1. Is there a way to retrieve a list of the services that are installed on a remote computer?

|  |
| --- |
| help Get-Service -Parameter computernamepower  We can retrieve the list of services that are installed on a remote computer using the below cmdlet:   * Get-service –ComputerName <hostname of the remote computer> |

1. Is there a way to see what processes are running on a remote computer?

|  |
| --- |
| Help Get-Process –Parameter computername  Yes, we can see what processes are running on a remote computer by using the below cmdlet:   * Get-process –ComputerName <hostname of the remote computer> |

1. Examine the help file for the Out-File cmdlet. The files created by this cmdlet default to a width of how many characters? Is there a parameter that would enable you to change that width?

|  |
| --- |
| Help Out-File –full  Or  Help Out-File –Parameter Width  Should show you 80 characters as the default for the PowerShell console. You would use this parameter to change it as well.  Yes, the width is 80 characters as the default for the PowerShell Console. We can change the width of the file that is created by using the “-Width” parameter, as follows:   * Ex:- get-process | Out-File –Width 50   This will change the width of the output file that will be created to 50. |

1. By default, Out-File will overwrite any existing file that has the same filename as what you specify. Is there a parameter that would prevent the cmdlet from overwriting an existing file?

|  |
| --- |
| If you run: Help Out-File –full and look at parameters you should see –NoClobber.  Yes, we can restrict the Out-File from overwriting the existing file with the same file name as what we specify using the parameter “NoClobber”  Ex:- Get-Process | Out-File –NoClobber <file path where it needs to be stored>  The above cmdlet throws an exception if there is already an existing file of the same name. |

1. How could you see a list of all aliases defined in PowerShell?

|  |
| --- |
| Get-alias  We can get the list of all the aliases defined in PowerShell by using the below cmdlet:  Get-Alias |

1. Using both an alias and abbreviated parameter names, what is the shortest command line you could type to retrieve a list of running processes from a computer named Server1?

|  |
| --- |
| ps –c server1  There are two alias for the cmdlet “get-process”. They are mentioned as below:   * gps * ps   The abbreviation for the parameter “–ComputerName” is “–c “ or “-cn”  We can use either of the alias in order to get the list of the processes of a computer named Server1 by using the below cmdlet:   * gps –c server1 * ps – c server1 |

1. How many cmdlets are available that can deal with generic objects? (Hint: remember to use a singular noun like “object” rather than a plural one like “objects”).

|  |
| --- |
| get-command -noun object  There are 9 cmdlets which are available to deal with the generic objects. They are:   * Where-Object * Sort-Object * Compare-Object * For-Each Object * Group-Object * Tee-Object * Measure-Object * New-Object * Select-Object |

1. This chapter briefly mentioned arrays. What help topic could tell you more about them?

|  |
| --- |
| help about arrays  or if you weren’t sure, use wildcards  help \*array\*  We will be getting a detailed explanation of the arrays by using the below cmdlet:  “Get-Help array”  The above cmdlet works only after updating the help using the cmdlet “Update-Help” |

1. The Help command can also search the contents of a help file. Are there any topics that might explain any breaking changes between PowerShell v1 and PowerShell v2?

|  |
| --- |
| Help "breaking change"  Or we might try searching  Help about\*powershell\* |

1. Display a list of running processes.  
     
   We can display the running processes of the system by using the cmdlet “**Get-Process**”.
2. Display the 100 most recent entries from the Application event log (don’t use Get-WinEvent for this – We’ve shown you another command that will do this task).  
     
   We can get the 100 most recent entries from the Application event log using the cmdlet

“**get -eventlog –logname Application –newest 100**”

1. Display a list of all commands that are of the “cmdlet” type (this is tricky – we’ve shown you Get-Command, but you’re going to have to read the help to find out how to narrow down the list as we’ve asked).  
     
   We can get the list of all commands that are of the type – “cmdlet” by using the below cmdlet:

**“Get-command –Command Type cmdlet“**

1. Display a list of all aliases.  
   We can display the list of all aliases using the command “**Get-Alias**”
2. Make a new alias, so that you can run that to get a directory listing.  
     
   We usually use the Get-ChildItem to get a directory listing. In order to set a new-alias “list” which can be used in the place of New-ChildItem we use the following cmdlet.

**“New-Alias -Name list -Value Get-ChildItem”**

1. Display a list of services that begin with the letter “M.” Again, read the help for the necessary command – and don’t forget that “\*” is a near-universal wildcard in PowerShell.  
     
   We can able to get the names of all the services that run on a machine using the cmdlet

**“Get-Service -Name m\*”**

1. Display a list of all Windows Firewall rules. You’ll need to use Help or Get-Command to discover the necessary cmdlet!  
     
   We will be able to display all the Windows Firewall rules using the below cmdlet:

**“Get-NetFirewallRule”**

1. Display a list only of inbound Windows Firewall rules. Same cmdlet as above, but you’ll need to read its help to discover the necessary parameter and its allowable values  
     
   To display the list of only Inbound Firewall rules we can use the below cmdlet

**“Get-NetFirewallRule -Direction Inbound”**