# Objectives

* + Locate commands for common administrative tasks.
  + List the available command modules.
  + List commands by verb, noun, and module.
  + Use wildcard characters to search for specific commands.

## Skills Reviewed

* + Issuing PowerShell commands

## New Skills

* + Locate PowerShell commands.
  + List available PowerShell command modules.
  + List PowerShell commands by verb, noun, and module.
  + Use wildcard characters to search for specific commands.

# Initial Conditions

Your virtual machine should be in this state prior to beginning this guided practice:

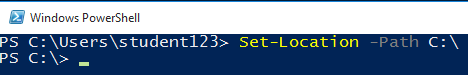
* + Windows 10 VM running in VCASTLE or VMWare Workstation.
  + Windows Server 2012 R2 is running in VCASTLE or VMWare Workstation.

# Final Conditions

No system configuration changes were made during this exercise. The user consumed all the output from the PowerShell command.

# Instructions

1. Log into **CLIENT1**.
2. The console window defaults sometimes result in text that is difficult to read or have features you need disabled. Configure **PowerShell** console as follows:
   * 1. Open **PowerShell** by right clicking **Start**, selecting **Run**, enter **powershell** in the **Open:** dialog box and then clicking **OK**.
     2. **Windows PowerShell** opens. Right click on the title bar and select **Properties**.
     3. On the **Font** tab, select size **16** and **Lucida Console** font. It is acceptable to select another font size if you can read it better.
     4. On the **Options** tab, note the **Command History Buffer size** and verify that the **Quick Edit Mode** and **Insert Mode** check boxes are checked.
     5. On the **Layout** tab, set the **Screen Buffer**, **Windows Size**, and **Window Position** to suit your preferences. You will want to adjust these settings if your output requires you to perform excessive scrolling or if you could not scroll far enough up.
     6. On the **Colors** page, set the text coloring to suit your preferences.
     7. Click **OK** and look at **PowerShell**. Adjust the above items to suit your needs.
3. Use some basic commands.
   * 1. Enter the command **Set-Location -Path C:\** and notice the change in the prompt. The prompt shows the working directory.



* + 1. Enter the **Get-Host** command and note the available data. The host in PowerShell is the console window and command interpreter. In the output, you can see the version of **PowerShell**. At times when commands are not working correctly or are not available you will want to check the version and compare to the required version.
    2. If you recall the buffer history size from the properties dialog, the console will remember the last 50 commands that were typed. You can access these using the up and down arrows. In the **PowerShell** prompt, use the up-arrow key to scroll through the previous commands you typed until you get to the **Set-Location** command. Change the end of the command to be **C:\Windows** instead of **C:\** and press the enter key.
    3. You can search the command history using **CTRL+R** (reverse search) or **CTRL+S** (forward search). The text you entered will be found in the command history in any position. Depress **CTRL+R** and then enter **ho**. You have only entered two characters; the complete command is listed at the prompt. Depress enter to execute the command.



1. Use **Tab** completion.
   * 1. PowerShell allows for tab completion of commands and parameter names. To access this for a command you must type at least the verb for a cmdlet and a dash and then press the tab key. You can start typing the noun before pressing the **Tab** key. Enter **Get-** and then press the **Tab** key until you get to the **Get-ChildItem** command and then depress **Enter**. Hint: **Shift+Tab** scrolls backwards.
     2. Enter **Get-C** and press the **Tab** key through the possible completions. Note how this differs from the previous command. Tab completion is helpful but you should complete as much of the command as you can to find the command as quickly as possible.
     3. Use the command history to access the **Get-ChildItem** and then type a space and the dash character as shown below. Depress the tab key to scroll through the possible parameter names for the **Get-ChildItem** cmdlet. Continue until you get to the **–Recurse** parameter and then press **Enter**. This one parameter had a huge effect on the output of the cmdlet. (**Hint**: If you get tired of watching the output press **CTRL+C**).

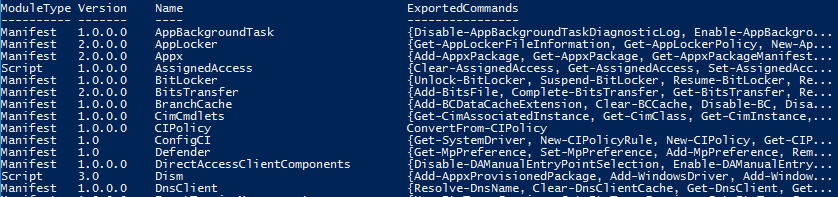


1. Use PowerShell modules.

The format of a cmdlet is ***verb-noun***. This can help you when you are attempting to locate commands in PowerShell. In addition, cmdlets are grouped into modules that are related to a specific functions or applications. This knowledge can help you to narrow down your search for specific commands.

* + 1. Identify available PowerShell modules by entering the command:

**Get-Module -ListAvailable**



* + 1. **DnsClient** is one of the available modules. In order to use command available in this module, you must import the module. Import the module by enter the command **Import-Module –Name DnsClient**

**Note**:In PowerShell 3.0 and later, executing a cmdlet in a module will cause the module to auto-load.

* + 1. The view the commands in a specific module, use the **Get-Command** cmdlet with the **–Module** parameter. List the command available in the **DnsClient** module.



1. Use **Get-Command** cmdlet and the **–Noun** parameter to find commands if you know the noun portion of the cmdlet. Enter **Get-Command –Noun Item** command. This command will show all commands that have **Item** as the noun in the cmdlet. You should see nine cmdlets that have the noun portion as **Item**.
   * 1. What can you infer about the purpose of the **Remove-Item** cmdlet, based solely on its name?
2. Use the **Get-Command** cmdlet and the **–Verb** parameter to find commands that have the verb **Set**. Enter **Get-Command –Verb Set** command.
   * 1. Below is a list of other common verbs and their meanings:
        1. **Clear** - Erase or remove information
        2. **Write** - Write information to the screen
        3. **Copy** - Copy an object
        4. **Set** - Change the value(s) of an object
        5. **Get** - View information about an object
3. Use the **Get-Command** cmdlet and the **–ParameterName** parameter to find cmdlets if you know the parameter name. Enter **Get-Command –ParameterName ComputerName** command. In this case, by looking for commands that have the **ComputerName** parameter we can find commands that can be run against remote machines.
4. Use **Get-Command** cmdlet and wildcards to find cmdlets if you know a partial name.
   * 1. **Get-Command \*Item** lists all cmdlets that end with **Item**. **Get-Command \*item\*** lists all cmdlet that have the characters **item** in the cmdlet name.
     2. Execute both of the commands and look at the difference.

# Document work

1. Capture a snippet showing the output of the **Get-Command \*item\*** command. Save the snippet as **PowerShell\_*studentID*.png**. The snippet should contain:
   * 1. Command.
     2. At least 9 lines of output.

