Advanced Scripting   
Shell Navigation

Last Updated: 12/23/2022 10:28 AM Version 1  
Document Prepared for: CIT361 Student

# Name Daniel Harris ID 235868292

# Instructions

Save a copy of this document. Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

Navigating the shell is key to becoming proficient with PowerShell.

# Requirements

* PowerShell
* Class Sample Files (you will get them in the setup)
* Internet Access

# Setup

You will need some files for this exercise. To get them enter the following commands:  
**Invoke-WebRequest -Uri http://lf.cit.byui.edu/psfiles.zip -OutFile ~/psfiles.zip  
Expand-Archive ~/psfiles.zip -DestinationPath ~**

Those commands will put the files in your user folder (that is, your “home” directory), which is where they should be for the rest of this exercise. The directory structure of the files you downloaded looks like this:

psfiles  
├───data  
├───files  
│ ├───logs  
│ └───pictures  
└───utilities

# Task 1—The Current Directory

The current directory is the directory that the shell is currently “in”. The directory structure is like a tree. The current directory is the branch you are standing on. If you specify a resource by filename without any path information on the shell will look for the file in your current directory.

## Steps

1. By default, the PowerShell prompt displays the current directory. When I start my PowerShell, the prompt looks like this:  
   PS C:\Users\craig\OneDrive\Scripts>  
   This tells me that my current directory is C:\Users\craig\OneDrive\Scripts. Anytime I specify a filename or filesystem command it will happen in this directory unless I specify otherwise. Chances are you are not in the same directory as I am. You can always get your current directory with the get-Location command. Enter the command  
   Get-Location
   1. What is your current path?Z:\
2. To see the aliases for Get-Location, enter the command:  
   Get-Alias -Definition Get-Location
   1. What are the aliases for Get-Location? gl -> Get-Location pwd -> Get-Location
3. What other location cmdlets are there? Enter the command  
   Get-Command \*-Location
   1. List the Location commands: Get-Location Pop-Location Push-Location Set-Location
4. You can use Push-Location to save your current location. Enter the command  
   Push-Location

# Task 2—Changing Directories

Often you will want to change the current directory to another more convenient place.

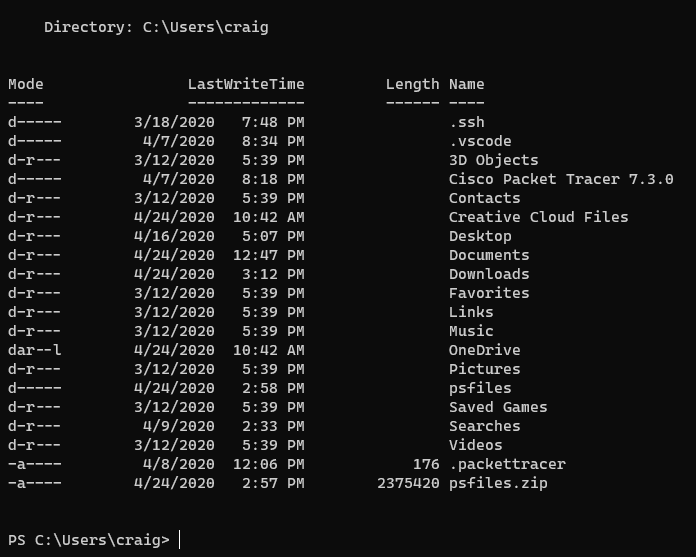
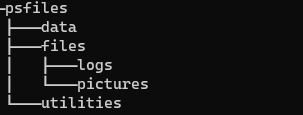
## Steps

1. Each user has a “home” directory. It can be anywhere on a system, but the shell will know where your home directory is. The ~ is used as an accelerator for your home directory. The Set-Location command is used to change your current directory. Change to your home directory with the following command:  
   Set-Location ~  
   What is your home directory? C:\Users\Harris
2. What aliases are defined for Set-Location? cd -> Set-Location -> Set-Location sl -> Set-Location
3. Use the cd alias to change to your root directory. Enter the command   
   cd /
4. Remember a bit ago we used **Push-Location** to save our directory location. You can use **Pop-Location** to change back to that directory. Enter the command  
   Pop-Location
   1. What is your current directory? Z:\
   2. Is it where you started? yes
5. Now go back to the home directory, enter:  
   cd ~
6. You should now be back to your home directory.

# Task 3—Directory Listings

Unless you have a perfect memory, you will often need to see what items are in a location. The **Get‑ChildItem** cmdlet does just that.

## Steps

1. To see the items in your current location, enter the command:  
   **Get-ChildItem**
2. I don’t know what is in your home directory; I can guess there are Documents, Downloads, and Pictures directories among others. I know you should have a file named psfiles.zip and a folder named psfiles. These were created in the setup step. My list looks like this:  
   
3. The ***d*** in the mode column indicates the item is a directory, the length is the size of file.
4. **Get-ChildItem** has a lot of options; we will explore a few of them. You will need to use help to learn the rest.
5. To limit the list to just directories use the -Directory switch. Enter  
   Get-ChildItem -Directory
6. To limit the list to just files use the -File switch. Enter the command:  
   **Get-ChildItem -File**
7. I’ll bet you are tired of typing Get-ChildItem. What aliases are defined for Get-ChildItem? dir -> Get-ChildItem gci -> Get-ChildItem ls -> Get-ChildItem
8. You can pass a directory to Get-ChildItem to get a listing for a different directory other than the current directory. Get a directory list of the psfiles folder. Enter the command:  
   **ls psfiles**
   1. What files/directories are in the psfiles folder? Data, files, utilites
9. You can get an item list that follows the directories down using the -recurse switch. Enter:  
   **dir psfiles -recurse**
10. Remember the psfiles folder structure looks like this  
    
11. Get a directory list from the pictures folder using a relative path. Enter:  
    **gci psfiles/files/pictures**
    1. You should see a listing of picture files. What is the first filename? 2ndEmpireHouseNight.jpg
12. You can use wildcards to list only some file types. To list only the jpeg files enter:  
    gci psfiles/files/pictures/\*.jpg
    1. How many files are there? 8
13. Write a command that would only list the files that start with **arpanet**. Ls psfiles \*arpanet
14. Make the pictures folder your current directory.  
    cd psfiles/files/pictures

# Task 4—Working with files and folders.

A common administrative task is to work with files and folder. You can make, rename, delete and move files and folder. The appropriate cmdlets are New-item, Rename-item, Remove-Item, Move-Item. Now you will further organize the pictures by type

## Steps

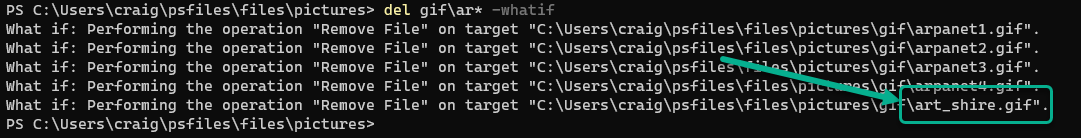
1. You should be in the pictures folder. Create folders for each picture type. First create a folder for jpeg files, name it jpeg. Enter the command  
   New-Item -Name jpeg -ItemType Directory
2. Use the Get-ChildItem (or one of its aliases) to see if the directory was created.
3. Create a new folder for the gif files named gif. This time use the md, or mkdir alias.
4. Now copy all the jpg files to the jpeg directory.   
   Copy-Item -Path \*.jpg -Destination jpeg
5. What are the aliases for the Copy-Item cmdlet? copy -> Copy-Item cp -> Copy-Item cpi -> Copy-Item
6. Verify the files were copied
7. Delete the jpg files from the pictures directory  
   Remove-Item \*.jpg
8. What are the aliases for the Remove-Item cmdlet? del -> Remove-Item -> Remove-Item -> Remove-Item ri -> Remove-Item rm -> Remove-Item rmdir -> Remove-Item
9. This time let’s move the gif files to the gif directory. Enter the command:  
   Move-Item -Path \*.gif -Destination gif
   1. Are the gif files still in the pictures directory? no
   2. Are they in the gif directory? yes
10. What are the aliases for the Move-Item cmdlet? Mv, move, mi
11. Make a directory named gifbackup.  
    md gifbackup
12. Copy the files from the gif folder to gifbackup  
    cp gif\\* gifbackup

# Task 5—Trusting Operations.

PowerShell has a set of common parameters. They are parameters that most cmdlets support. Three of them are -Whatif, -Verbose, and -Confirm.

* -Whatif is used to simulate the command. The command will run and tell you what action it would take but does not actually perform the action.
* -verbose, tells you details about the actions being performed.
* -Confirm requires the user to confirm each operation.

## Steps

1. Now that we have a backup of the gifs we can play with the files a bit. Suppose you wanted to delete all the arpanet images. Craft a statement like this:  
   del gif\ar\* but you are not quite sure it will work correctly. You can add the -whatif switch to the command to see what it would do if it really ran.
   1. Enter the command:  
      del gif\ar\* -Whatif
   2. You should see a list like this  
      
   3. Notice that the last file in the list should not be deleted. Now we have two choices. We can fix the wildcard pattern or we could run the command with the confirm switch and only confirm the files we really want to delete.
2. Lets use the confirm option. Enter the command  
   del gif\ar\* -Confirm
   1. You should be presented with a confirmation prompt for each action. The options are self-explanatory. Answer appropriately for each file until you are done.
   2. Use a directory listing to make sure the command did what you expected.
3. Now restore the deleted files from the backup with the correct wildcard pattern. Enter the command:  
   **cp gifbackup\arp\* gif**  
   Notice how the command didn’t tell us what happened.
   1. Verify the files were really copied
4. Now you will delete the arapanet files from the backup folder, this time using the **-Verbose** switch to see the details about what happened. Enter the command:  
   **del gifbackup/arp\* -Verbose**
   1. You should see detailed information about the process.

# Task 6—Cleanup

We have two cleanup tasks, the first is to get rid of the gifbackup folder and the second is for you to decide where you would like the psfiles.zip file and the psfiles folder.

## Steps

1. First get rid of the backup folder. Enter the command:  
   **del gifbackups**
   1. Notice the warning message, answer appropriately.
2. Now you can move the files we have been using to a place of your choosing. I had you put them in the home directory because I could write explicit instructions that would work for everyone. Now you know how to work with folders, you can put these files wherever you would like on your system. Move or copy them to a location of your choosing. You will need to use the files in the future so you will need to know how to get to them from the shell.
   1. Where did you move your files to? Z:\

# Deliverable

Upload this document with completed answers to i-learn.