CET 1014 – PowerShell – Project 1 of 2

Updated February 12, 2019 – Note this date and re-download this document periodically.

Overview

Due Date

See the course Moodle page for the due date.

A ten mark deduction will be the penalty for any late submission; this is the same as losing 20%. Five more marks will be deducted each day until the assignment is handed in.

Unique Submissions

The project is not a group assignment. Your script must not contain code that another individual created. You should not copy from other students, the Internet, or any other source – this will not help you learn. Use what you see to help you learn, understand, and create something unique.

Students are encouraged to share knowledge and help guide others. However, the purpose of the exercise is to help you understand the material and how the code works.

If your submission is very similar to that of another student, a sample I provided, or something found online; your marks will reflect that. If you have questions or are unsure about my meaning, please speak to me during class, well before the due date.

Comments and Formatting

Make sure your PowerShell script includes appropriate comments and formatting, use the # or <# #> to explain the various parts of your code, why they are needed, and what they do. Use the format document feature of Visual Studio Code.

Your Understanding

A large part of the evaluation is you verbally explaining to me how your PowerShell script works. If you cannot explain to me why you included each piece of code and what it does, your mark will reflect that.

The goal of the assignment is for you to gain an understanding of the material.

If you need help to build your confidence and understanding about the material, ask questions in class and seek a tutor. Tutoring information is at the top of the course Moodle page.

Creating Your Script

Overview

You are to create a menu driven PowerShell Script using Visual Studio Code that will solve each of the problems listed below.

Round all file sizes and percentages to 2 decimal places.

Testing

Your script needs to be able to run from any computer, and you must only use PowerShell to solve the problems.

Test your script before handing it in by copying it and running it on other computers.

Submission

You will hand in one PowerShell Script File titled **Project_1_ StudentNumber_Full_Name.ps1**

For example, I would hand in **Project_1_A00044040_Nathan_Abourbih.ps1**

Problems

Overview

Today got off to a rough start because your car broke down, and you had to walk uphill all the way to work in a snowstorm, and of course, it was the one day of the year that it is -30 degrees outside. You would have called in sick, but your manager, Douglas Reynholm called you first and pleaded for you to come in because you are the best PowerShell guru in the organization.

You arrived at work and are greeted by four of your collogues all huddled around your desk. As soon as you approach your co-workers, they all start talking at once.

“One at a time, one at a time, …. QUIET!” you yell.

As soon as everyone calms down, you realize that each of your four collogues has a problem that you can solve with PowerShell.

You write down each of the four colleagues names and listen to their problem, here are your notes:

Roy Trenneman

Roy manages the backup system, and he says that the backup last night did not copy all of the files and folders it was supposed to.

Somehow the NTFS permissions on some files and folders were changed. He explains that the Administrators group should have full control over all files and folders.

He takes your mouse and shows you an example of a changed folder by right-clicking it and selecting Properties, choosing the Security tab, and while pointing to the screen, he shows you that the Administrators group is listed.

He needs a list of all of the folders and files that do not grant the Administrators group Full Control.

Please create a script that will:

- Ask the user to type the path to a folder; this will be the folder that they want to be inspected.
    - *Example: C:\Users*
- Recursively determine which of the files and folders inside that path do not grant the Administrators group Full Control.
- Save your findings as one professional looking file in the current directory.
    - Format the information as a table with headings, columns, and rows.
    - File format is up to you.
    - The files should have an extension that is appropriate for the file's format.
- You should include the following information in your output:
    - Full path to the file or folder.
    - Date and time the file or folder was last accessed.
    - Date and time the file or folder was last changed.

Optional for Bonus Marks:
For each file and folder you found, include a list of which users and groups do have Full Control access to that file or folder and combine all the information into a single table.

Jen Barber

Jen manages all of the file servers, and she says that the research division of the company just hired 30 new people that will all be starting tomorrow.

She needs to create a new shared folder for each of these new people. She explains that she got HR to export a list of the people to a text file, but she needs your help to use PowerShell to use that text file to create the shared folders.

You ask about NTFS permissions, and Jen says just to leave them as default because she already has a script that will set the NTFS permissions and she will run it after running your script.

Please create a script that will:

- Ask the user to type the path to a text file; this is the text file with the names in it.
    - *Example: C:\Users\User\Desktop\NewHires.txt*
- Ask the user to type the path to a folder; the new shared folders should be created inside this folder.
    - *Example: C:\Shares*
- For each of the new employees listed in the text file, create a new folder.
    - *Example Full Path: C:\Shares\NewEmployeeName*
- Instruct Windows to share this folder and give the share the same name as the new folder. The share permissions should be Everyone - Full Control.
    - *Example UNC Path: \\ComputerName\NewEmployeeName*
- Display the list of employees on the screen in a professional way that is easy to understand and use.
- You should include the following information in your output:
    - Employee's name.
    - Full path to the new shared folder.
    - UNC path to the new shared folder.

Optional for Bonus Marks:
Combine the information into a single table and use a different colour for each of the pieces of information.

Douglas Reynholm

Douglas, your manager, says that upper management is coming down hard on him to get his budget for computer upgrades submitted. He wants to upgrade everyone to at least 8GB of memory (RAM) and replace everyone's C drive with a solid-state hard drive.

However, he doesn't know how many hard drives or how much memory (RAM) to buy. He has a list of computers to check in a text file to help make things a little easier for you.

Please create a script that will:

- Ask the user to type the path to a text file; this is the text file with the computer names in it.
    - *Example: C:\Users\User\Desktop\Computers.txt*
- For each of the computers listed in the text file, retrieve the size and free space of the C drive and retrieve how much memory (RAM) the computer has.
- Display your findings on the screen in a professional way that is easy to understand and use.
- You should include the following information in your output:
    - Computer name.
    - IP address.
    - C drive size in GB.
    - C drive free space as a percentage.
    - How much memory (RAM) the computer has in GB.

Optional for Bonus Marks:
Combine the information into a single table and use various techniques to make your report very aesthetically pleasing.

Maurice Moss

Maurice manages the file servers with Jen, and he says that one of the servers keeps running out of disk space. He needs to keep an eye on the files that are saved to the servers to make sure that no one is duplicating large files or saving any abnormal files.

The reason this is a problem today because one of the primary servers is out of disk space right now and he is having trouble finding something he can safely delete or move somewhere else.

Please create a script that will:

- Ask the user to type the path to a folder; this will be the folder that they want to be inspected.
    - *Example: C:\Users*
- Recursively examine the files and folders inside that path and produce the following three reports:
    - **Top 25 Newest Files**
      Show the 25 most recent files created.
    - **Top 20 Largest Files**
      Show the 20 largest files.
    - **Top 10 Recently Accessed Files**
      Show the 10 most recently accessed files sorted by file type.
- Save your findings as three professional looking files in the current directory.
    - Format the information as a table with headings, columns, and rows.
    - The files should have a file extension that is appropriate for the file's contents.
    - File format is up to you.
- You should include the following information in your output for each of the reports:
    - Full path to the file.
    - File size in MB.
    - Date and time the file or folder was last accessed.
    - Date and time the file or folder was last changed.

Optional for Bonus Marks:
Include all three reports in one file, include a well-written title for each report, a well-written and descriptive description for each report, and use various techniques to make your report very aesthetically pleasing.

A Final Thought

Because you are a smart IT analyst, you realize that everything you have created could be used by the entire IT department.

You start daydreaming for a moment, and you imagine all of the prestige to be had if you were to distribute your script to everyone in the department. More importantly, you think about all of the baked goods that your collogues might reward you with for saving them so much time.

As a result, you decide to put all of your code into one PowerShell script file (.ps1) and set it up so that when your collogues run it, they will be presented with a menu asking them what they would like to do. Depending on what they choose, the appropriate section of your script will be executed.

Marks

A rubric will be posted to Moodle shortly so you can see precisely how marks will be awarded.

After the projects are submitted, I will be sitting down with each student independently so you can verbally explain your project to me.

Marks will be based on your ability to both verbally explain your project to me as well as its ability to solve the problems listed in the project description.

If you include code in your project that you cannot explain, your marks will reflect that.