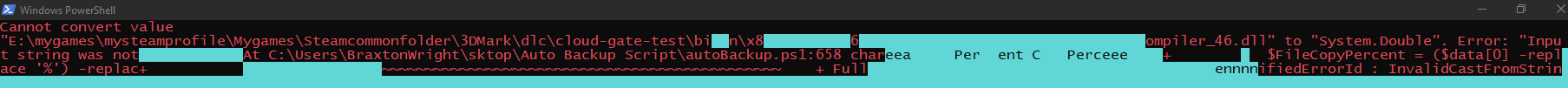
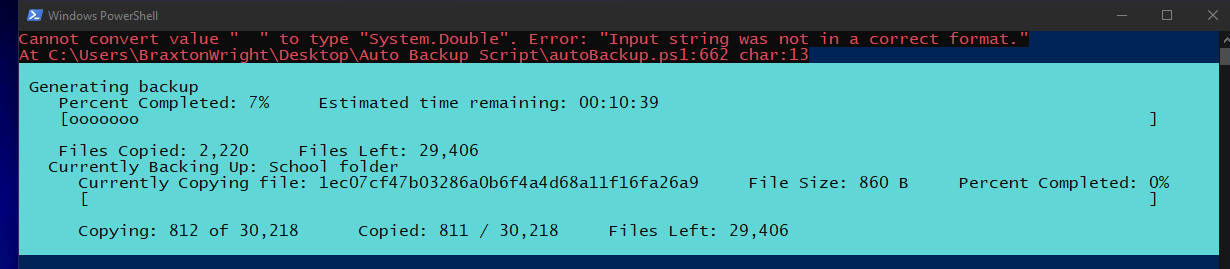
1. Need to see about implementing Start-Threadjob for the counting of the files before the backup progress to speed up the counting of the files. In addition, after you start the jobs, you need to make it so that it automatically starts backing up to make it as fast as possible and the jobs will return the file count after they are done counting, this is because of files like inside a unity project. One project alone contains around 15,000 files. File explore both starts to copy the files while it is counting them and it updates the files remaining portion of the backup.

Also, see about debugging the process and see what is causing it to take so long. This is because when you try to do the job “Unity Projects” it will freeze at the counting of the files and I do not know at what step it is causing the issue. If it is the counting of the files from the source/destination or comparing the objects. Or see if this issue is fixed because of the change of implementation of counting the files (I think this issue is fixed because we are no longer using the Compare-Object cmdlet, we are instead checking each file as we find them).

1. See about finding some logic to better calculate the time remaining for the backup/restore process. It currently does not take into account the size or copy rate of the files, only the number of files copied.
2. See if you can optimize the code to run faster? See if it needs optimized, test by timing how long it takes to copy a set of files using file explorer, a bare bone copy of the robocopy script, and your current iteration of the script.
3. Add a job summary once the backup/restore process has completed. Such as how long it took to complete, how many files were processed, ect. Like how the end of a robocopy, there is that summary table for the job.
4. There is some issue with copying files, it will get a string when it is getting the file size for the function Get-RobocopyProgress. At one time, I got this error in running the script, saying it can’t convert a " " to type "System.Double". However, this issue is every inconsistent when it occurs. I copied ~80,000 files (168 GB total) and I never received an error. I might have to fix by adding a try/catch block here to handle when it doesn’t work? I also receive an error saying “Cannot validate argument on parameter 'PercentComplete'. The 10021875 argument is greater than the maximum allowed range of 100.” See the video inside your videos folder for the video and the txt document containing the errors this run of robocopy pipe returned.
5. Need to modify the logic for the first write-progress parameters because when the first start backing up the files, you get a situation where the “Files Copied“ field will say -1 / # as shown below as I was editing a video of the process running. Probably not worth fixing because, it is only there for a fraction of a second. The moment it starts to copy a file, it is set to 0.  
   
6. See about adding an option to allow the user to choose if they want a progress bar to be shown or not, because it will result in the process being faster I think. Need to use the Measure-Command cmdlet to see if it has any effect.
7. Need to see about adding an option to allow the user to refresh the contents of the jobs so if they added or modified a job, they can simply refresh the contents instead of having to start the script again.
8. See about adding an option to allow users to exclude directories/files as talked about [here](https://pureinfotech.com/exclude-files-folders-robocopy-windows-10/).

Notes:

Network shares are caps insensitive, I.E. “\\WORKSTATION\VM\_SHARED”, “\\workstation\VM\_Shared”, and even this wacky entry “\\WoRkStAtIoN\Vm\_ShArEd” all take you to the same network share.