

Powershell home task

Calculate hash value of a file of your choice using SHA256 algorithm.

```
Get-FileHash .\path\to\file -Algorithm SHA256
```

Calculate MD5 hash values of all files in current folder with sizes less than 10 KB and save to a file **output.txt** only the hashes themselves (without path, algorithm and other parts of standard output).

```
Get-ChildItem -Exclude output.txt | Where-Object Length -CLT (10000)
| Get-FileHash -Algorithm MD5 | Select-Object -ExpandProperty Hash > output.txt
```

Move **output.txt** to current user desktop folder using environment variable.

Note: not using `$env:USERPROFILE` as it will fail for non-standard desktop locations

```
Move-Item -Path .\output.txt
-Destination ([Environment]::GetFolderPath("Desktop"))
```

Clarify the difference between a service and a process.

Services are processes running in the background without depending or interacting with the user / desktop.

Processes are instances of running executables either launched by the user or their session.

Clarify the differences between **-Contains**, **-Match**, **-Like** filters.

- **-Contains**, staying true to OOP, returns a boolean value whether or not a collection contains the object passed as an argument to the filter
- **-Like** performs a check on a string based on wildcards, returning a boolean value if the checked wildcard matches within the string
- **-Match** also performs a check on a string, however the check is based on regular expression and not wildcard, returning a boolean value

Output a list of services that cannot be stopped by a user and are currently in running state.

```
Get-Service | Where-Object Status -eq Running
| Where-Object -Property CanStop -Like False
```

Save to **errors.txt** last five entries from System log that fulfill the following requirements: their type is "Error", there are words "service" or "services" in their description.

Note: the wildcard will also match the plural version

```
Get-EventLog -LogName System -EntryType Error -Message *service* -Newest 5  
| Out-File -FilePath errors.txt # warning: results are truncated
```

Adding **Select-Object**, remove unnecessary information from the output, leaving only the following properties: message, entry type, event id.

```
Get-EventLog ... | Select-Object -Property EntryType, InstanceID, Message  
| Out-File -FilePath errors.txt # warning: results are truncated
```

Note that Message, which is usually long, is truncated. Try avoiding that.

```
Get-EventLog ... | Select-Object -Property EntryType, InstanceID, Message  
-ExpandProperty Message | Out-File -FilePath errors.txt
```

Output a list of all users in the system. Then output the list of all active users in csv format listing only their names and descriptions.

```
Get-LocalUser # lists all users  
Get-LocalUser | Where-Object Enabled | Select-Object -Property Name, Description  
| Export-Csv -Path .\path\to\file.csv
```

You can check the csv file using

```
Import-Csv .\path\to\file.csv | Format-List
```

Create a new disabled local user account with a name "testuser" without a password, making user the user cannot change the password.

```
New-LocalUser -Name testuser -NoPassword -UserMayNotChangePassword -Disabled
```

Output a list of physical drives.

```
Get-PhysicalDisk
```

Add a hard drive to the system, initialize it with MBR and create a partition of a maximum available size. Assign drive letter S to it. Format the partition using NTFS file system and setting volume label to "testdrive".

Note: below, replace *N* with the appropriate drive number obtained from either *Get-Disk* or *Get-PhysicalDisk*

```
Initialize-Disk -Number N -PartitionStyle MBR -PassThru | New-Partition  
-DriveLetter S -UseMaximumSize | Format-Volume -FileSystem NTFS  
-NewFileSystemLabel testdrive
```

Note: due to the nature of the above pipeline, Windows may display prompts which can safely be ignored

Remove "testdrive" partition.

```
Remove-Partition -DriveLetter S
```

Split the disk into two parts, where the first one is of 100 MB size and the other one occupies the rest of the drive space. Make sure both partitions are active. Assign drive letters X and Y to these two partitions respectively.

Note: since the partitions aren't formatted they won't be usable and Windows may prompt you to format them

```
Get-Disk -Number N | New-Partition -Size 100MB -IsActive -DriveLetter X  
| New-Partition -UseMaximumSize -IsActive -DriveLetter Y
```

Assign the partition with a drive letter X a new drive letter – W.

```
Set-Partition -DriveLetter X -NewDriveLetter W
```

Insert a DVD-ROM into DVD-ROM drive and mount it using *Mount-DiskImage*.

Note: *Mount-DiskImage* mounts a previously created disk image, not making it suitable for physical optical disk drives

```
Mount-DiskImage -ImagePath .\path\to\disk
```

Output a list of logical volumes present in the system listing them in order of remaining free space.

```
Get-Volume | Sort-Object -Property SizeRemaining -Descending
```