Créations de dossier partager :

```powershell
$dossier = "C:\share"
$groupe = "Administrateurs"
$nomPartage = "share"

if (-not (Test-Path $dossier)) {
    New-Item -Path $dossier -ItemType Directory
    Write-Host "Dossier $dossier créé."

    if (Get-LocalGroup | Where-Object { $_.Name -eq $groupe }) {
        $partageExiste = Get-SmbShare | Where-Object { $_.Name -eq $nomPartage }

        if ($partageExiste) {
            Write-Host "Le partage '$nomPartage' existe déjà. Aucune action n'a été effectuée."
        } else {
            New-SmbShare -Path $dossier -Name $nomPartage -FullAccess $groupe
            Write-Host "Partage SMB créé avec succès."


            icacls $dossier /grant "${groupe}:(F)"
            Write-Host "Droits d'accès complets accordés au groupe '$groupe'."
        }
    } else {
        Write-Host "Le groupe '$groupe' n'existe pas sur ce système."
    }
} else {
    Write-Host "Le dossier $dossier existe déjà."
}
```

Liste de l’ensemble des utilisateurs

```powershell
$users = Get-LocalUser
$date = Get-Date
$nbr_user = $users.Count
$fichier = "users.txt"

if ((Test-Path $fichier)) {
    Remove-Item $fichier
}

```

```
10  | $contenu = "Date d'éxécution du scrip : $date / "
11  | $contenu += "Nombre d'utilisateurs : $nbr_user / "
12  | $contenu += "Liste des utilisateurs : "
13  |
14  | foreach ($user in $users) {
15  |     $contenu += "$user / "
16  | }
17  |
18  | $contenu | Out-File -FilePath $fichier
19  | Write-Host "Le fichier '$fichier' a été créé mise a jour"
```

Menu d’administation

```
1   | do {
2   |     Write-Host "[1] Afficher le répertoire courant"
3   |     Write-Host "[2] Lister les fichiers du répertoire courant"
4   |     Write-Host "[3] Afficher le nom d'utilisateur courant"
5   |     Write-Host "[4] Afficher le nom de la machine"
6   |     Write-Host "[5] Afficher la liste des interfaces réseau et
leur adresses IP"
7   |     Write-Host "[6] Créer un rapport"
8   |     Write-Host "[7] Quittez"
9   |
10  |     $choix = Read-Host("Faites votre choix ")
11  |
12  |     switch ($choix) {
13  |         1 {
14  |             Write-Host "`nRépertoire courant :`n" $pwd `n
15  |         }
16  |         2 {
17  |             Write-Host "`nFichier du répertoire courant :`n"
18  |             ls
19  |             Write-Host "`n"
20  |         }
21  |         3 {
22  |             Write-Host "`nUtilisateur courant :"
23  |             $env:USERNAME
24  |             Write-Host "`n"
25  |         }
26  |         4 {
27  |             write-Host "`nNom de la machine :"
28  |             $env:COMPUTERNAME
29  |             Write-Host "`n"
30  |         }
31  |         5 {
32  |             Write-Host "`n"
33  |             foreach ($interface in Get-NetIPAddress) {
34  |                 Write-Host "Interface:
$($interface.InterfaceAlias) - IP: $($interface.IPAddress)"
```

```
  35  |             }
  36  |             Write-Host "`n"
  37  |         }
  38  |         6 {
  39  |             # Définir le dossier de destination basé sur le
répertoire du script
  40  |             $rapportFolder = Join-Path -Path $PSScriptRoot -
ChildPath "rapports"
  41  |
  42  |             # Créer le dossier s'il n'existe pas
  43  |             if (-not (Test-Path -Path $rapportFolder)) {
  44  |                 New-Item -ItemType Directory -Path $rapportFolder
  45  |             }
  46  |
  47  |             # Interface et IP
  48  |             $interfaceFichier = "$rapportFolder\Interface_IP.txt"
  49  |             $interfaces = ""
  50  |             foreach ($interface in Get-NetIPAddress) {
  51  |                 $interfaces += "$($interface.InterfaceAlias) -
IP: $($interface.IPAddress)`n"
  52  |             }
  53  |
  54  |             Creer-Fichier -filePath $interfaceFichier -titre
"Interface et IP" -contenu $interfaces
  55  |
  56  |             # Processeur
  57  |             $processorFichier = "$rapportFolder\Processeur.txt"
  58  |             $processor = (Get-WmiObject -Class
Win32_Processor).Name
  59  |             Creer-Fichier -filePath $processorFichier -titre
"Processeur" -contenu $processor
  60  |
  61  |             # RAM
  62  |             $ramFichier = "$rapportFolder\RAM.txt"
  63  |             $ram = Get-WmiObject -Class Win32_PhysicalMemory |
Select-Object Manufacturer, PartNumber, Capacity, DeviceLocato
      | r | Format-List | Out-String
  64  |             Creer-Fichier -filePath $ramFichier -titre "RAM" -
contenu $ram
  65  |
  66  |             # Disque dur
  67  |             $disqueFichier = "$rapportFolder\DisqueDur.txt"
  68  |             $disque = (Get-WmiObject -Class
Win32_DiskDrive).Model
  69  |             Creer-Fichier -filePath $disqueFichier -titre "Disque
Dur" -contenu $disque
  70  |
  71  |             # Carte mère
  72  |             $motherboardFichier = "$rapportFolder\CarteMere.txt"
  73  |             $motherboard = (Get-WmiObject -Class
```

```
Win32_BaseBoard).Product
  74    |             Creer-Fichier -filePath $motherboardFichier -titre
"Carte Mère" -contenu $motherboard
  75    |
  76    |             # Carte Graphique
  77    |             $gpuFichier = "$rapportFolder\CarteGraphique.txt"
  78    |             $gpu = (Get-WmiObject -Class
Win32_VideoController).Caption
  79    |             Creer-Fichier -filePath $gpuFichier -titre "Carte
Graphique" -contenu $gpu
  80    |
  81    |             # Périphérique réseau
  82    |             $netFichier =
"$rapportFolder\PéripheriquesRéseau.txt"
  83    |             $carteReseau = ""
  84    |             foreach ($networkAdapter in Get-WmiObject -Class
Win32_NetworkAdapter) {
  85    |                 $networkAdapters += "$($networkAdapter.Name)`n"
  86    |             }
  87    |             Creer-Fichier -filePath $netFichier -titre
"Périphériques Réseau" -contenu $carteReseau
  88    |
  89    |             # Utilisateurs
  90    |             $userFichier = "$rapportFolder\Utilisateurs.txt"
  91    |             $users = Get-LocalUser | Select-Object Name, Enabled
| Format-Table | Out-String
  92    |             Creer-Fichier -filePath $userFichier -titre
"Utilisateurs" -contenu $users
  93    |
  94    |             Write-Host "Les rapports ont été générés dans le
dossier : $rapportFolder`n"
  95    |         }
  96    |     }
  97    | } while ((-not ($choix -ceq 7)) -or ($choix -ceq "Quittez"))
  98    |
  99    | function Creer-Fichier {
 100    |     param (
 101    |         [string]$filePath,
 102    |         [string]$titre,
 103    |         [string]$contenu
 104    |     )
 105    |
 106    |     $titreFormatte = "$titre :`n`n"
 107    |     $contenuComplet = $titreFormatte + $contenu
 108    |
 109    |     $contenuComplet | Out-File -FilePath $filePath -Force
 110    |
 111    |     Write-Host "Fichier créé ou remplacé avec succès : $filePath"
 112    | }
```