**Initiation aux scripts PowerShell**

(Rédaction Scripts)

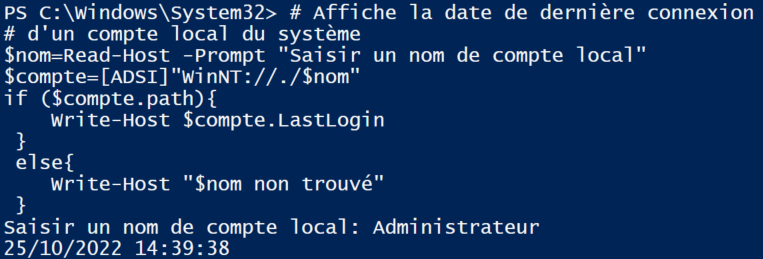
Ce TP et fait initialement pour Win 7 (système utilisé Win 11)

Environnement utilisé Powershell ISE (Integrated Scripting Environment)

**Ex 1: Accès aux comptes locaux du système**

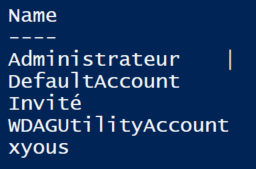
| # Affiche la date de dernière connexion # d'un compte local du système $nom=Read-Host -Prompt "Saisir un nom de compte local" $compte=[ADSI]"WinNT://./$nom" if ($compte.path){  Write-Host $compte.LastLogin  }  else{  Write-Host "$nom non trouvé"  } |
| --- |





Afficher la liste des utilisateurs locaux

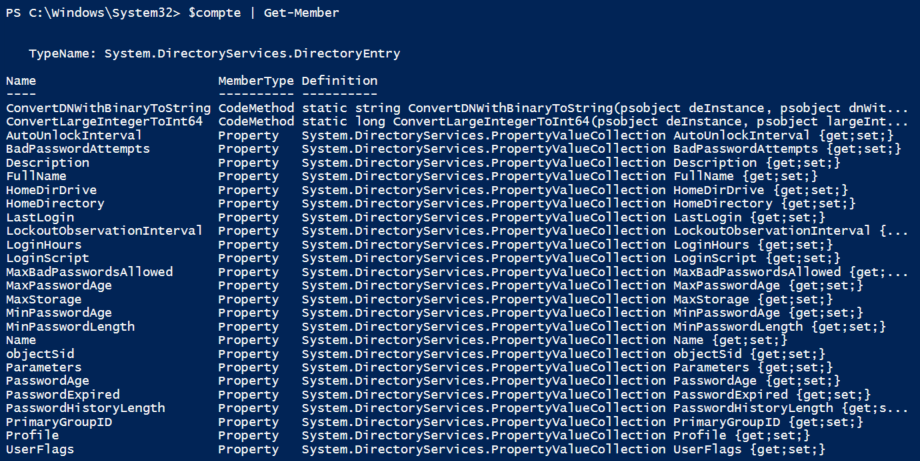
| Get-WmiObject Win32\_UserAccount | Select-Object Name |
| --- |

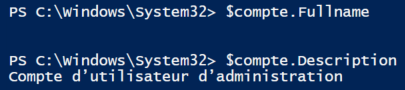


Permet d’accèder à la base de données locales des utilisateurs

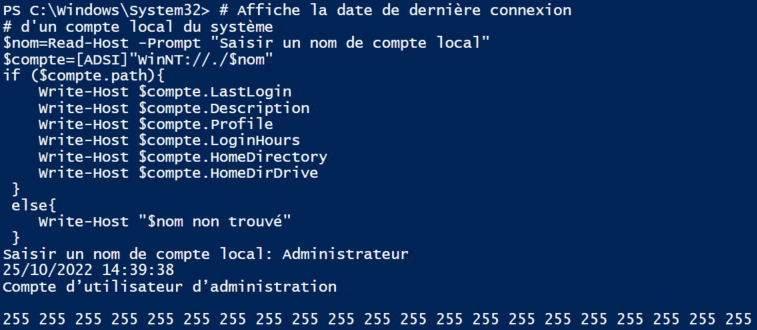
**[ADSI]"WinNT://.**

Les usernames sont case-sensitive sensibles





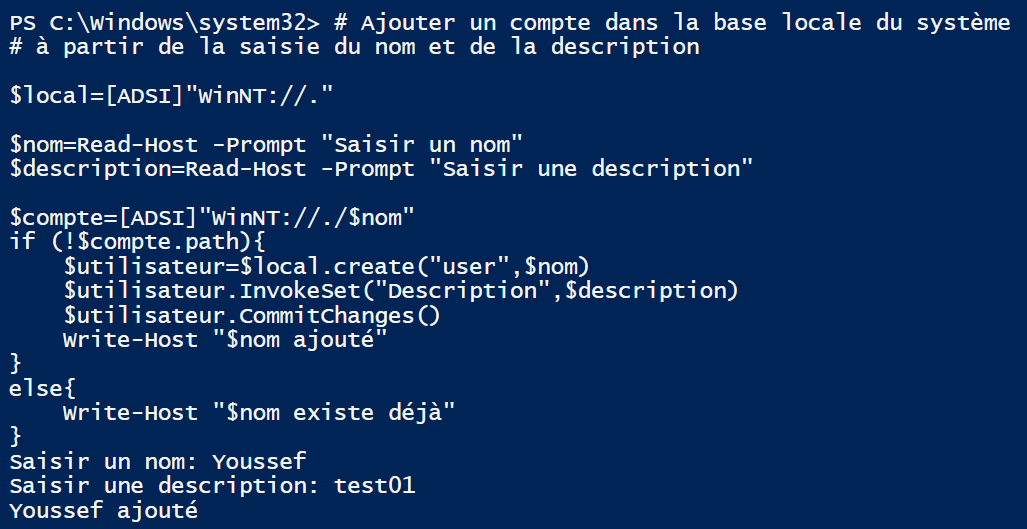
| **# Affiche la date de dernière connexion # d'un compte local du système $nom=Read-Host -Prompt "Saisir un nom de compte local" $compte=[ADSI]"WinNT://./$nom" if ($compte.path){  Write-Host $compte.LastLogin  Write-Host $compte.Description  Write-Host $compte.Profile  Write-Host $compte.LoginHours  Write-Host $compte.HomeDirectory  Write-Host $compte.HomeDirDrive  }  else{  Write-Host "$nom non trouvé"  }** |
| --- |

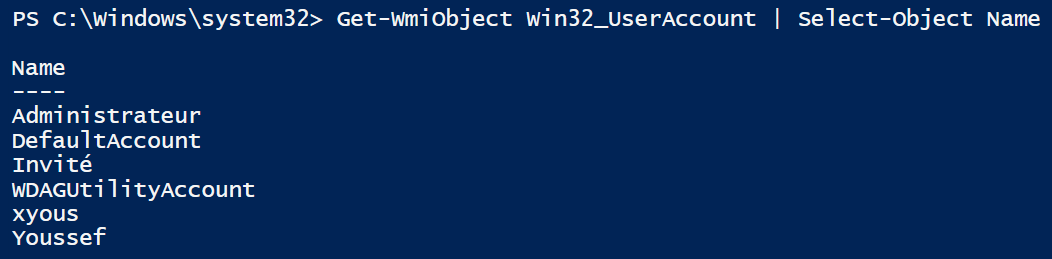
****

Plusieurs propriétés ne retournent rien.

**Ex 2: Ajout d’un compte local du système**

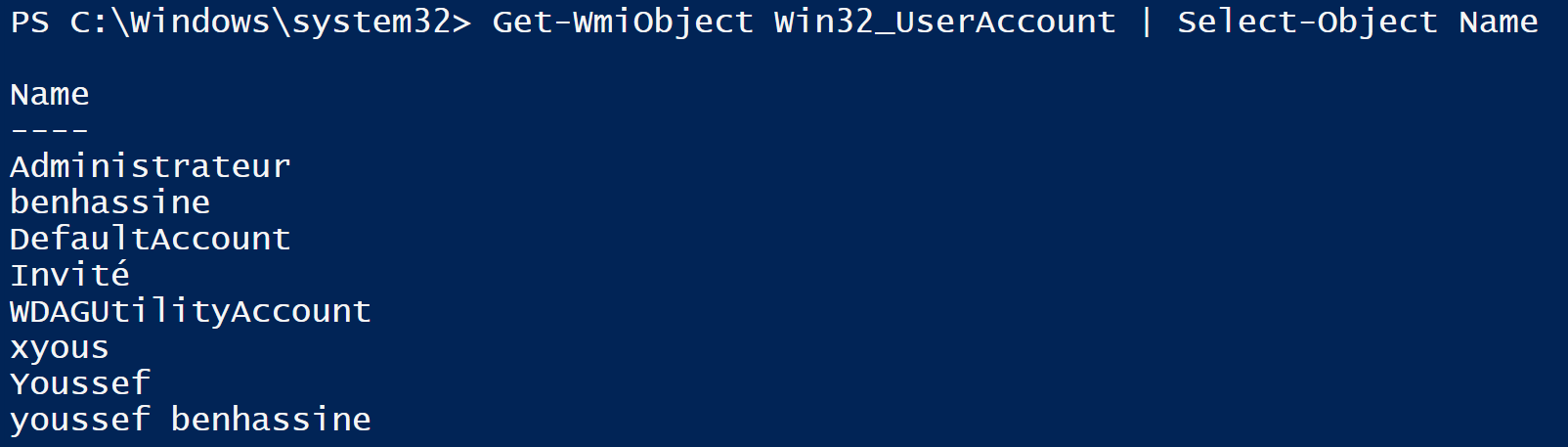
| **# Ajouter un compte dans la base locale du système # à partir de la saisie du nom et de la description  $local=[ADSI]"WinNT://."  $nom=Read-Host -Prompt "Saisir un nom" $description=Read-Host -Prompt "Saisir une description"  $compte=[ADSI]"WinNT://./$nom" if (!$compte.path){  $utilisateur=$local.create("user",$nom)  $utilisateur.InvokeSet("Description",$description)  $utilisateur.CommitChanges()  Write-Host "$nom ajouté" } else{  Write-Host "$nom existe déjà" }** |
| --- |

****

****

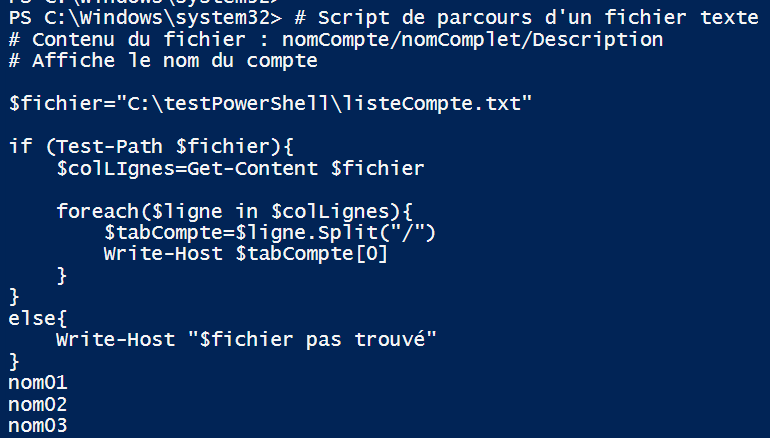
| # Méthode numéro 1  $local=[ADSI]"WinNT://."   $prenom=Read-Host -Prompt "Saisir un prénom" $nom=Read-Host -Prompt "Saisir un nom"  $nom="$prenom $nom" $description=Read-Host -Prompt "Saisir une description"  $compte=[ADSI]"WinNT://./$nom" if (!$compte.path){  $utilisateur=$local.create("user",$nom)  $utilisateur.InvokeSet("Description",$description)  $utilisateur.CommitChanges()  Write-Host "$nom ajouté" } else{  Write-Host "$nom existe déjà" } |
| --- |

| # Méthode numéro 2  $local=[ADSI]"WinNT://."  $nomComplet=Read-Host -Prompt "Saisir nom complet"  $prenom= ($nomComplet -split " ")[0] $nom= ($nomComplet -split " ")[1]  $nomComplet="$prenom $nom" $description=Read-Host -Prompt "Saisir une description"  $compte=[ADSI]"WinNT://./$nomComplet" if (!$compte.path){  $utilisateur=$local.create("user",$nomComplet)  $utilisateur.InvokeSet("Description",$description)  $utilisateur.CommitChanges()  Write-Host "$nomComplet ajouté" } else{  Write-Host "$nomComplet existe déjà" } |
| --- |

****

**Ex 3: Parcours d’un fichier texte contenant les informations des comptes utilisateurs**

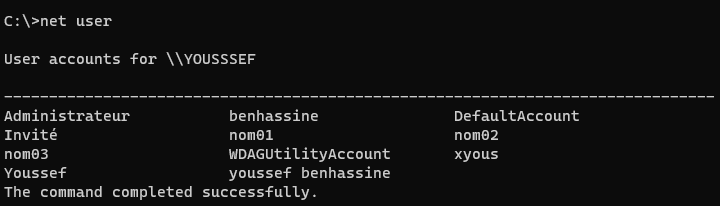
| # Script de parcours d'un fichier texte # Contenu du fichier : nomCompte/nomComplet/Description # Affiche le nom du compte  $fichier="C:\testPowerShell\listeCompte.txt"  if (Test-Path $fichier){  $colLIgnes=Get-Content $fichier    foreach($ligne in $colLignes){  $tabCompte=$ligne.Split("/")  Write-Host $tabCompte[0]  } } else{  Write-Host "$fichier pas trouvé" } |
| --- |



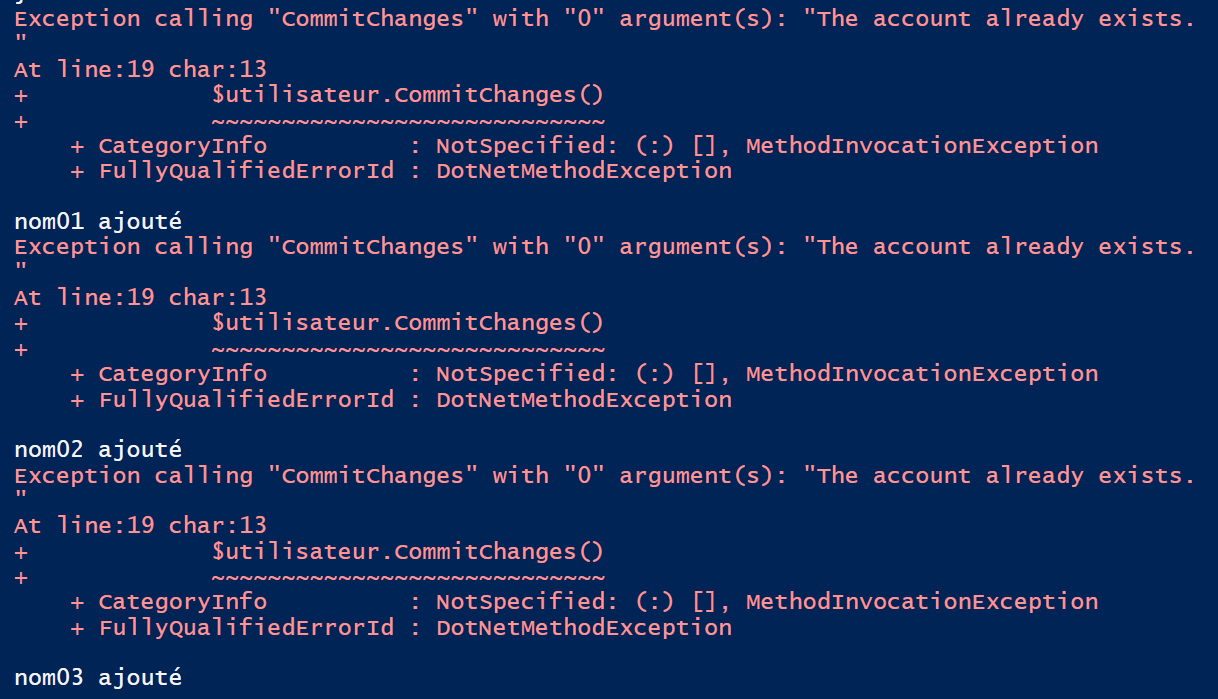
| # Script de parcours d'un fichier texte # Contenu du fichier : nomCompte/nomComplet/Description # Affiche le nom du compte, le nom complet et la description  $fichier = "C:\testPowerShell\listeCompte.txt"  if (Test-Path $fichier){  $colLignes = Get-Content $fichier    foreach ($ligne in $colLignes){  $tabCompte = $ligne.Split("/")  $username = $tabCompte[0]  $fullName = $tabCompte[1]  $description = $tabCompte[2]    Write-Host "Nom du compte : $username"  Write-Host "Nom complet : $fullName"  Write-Host "Description : $description"  Write-Host "-------------------"  } } else {  Write-Host "$fichier pas trouvé" } |
| --- |

Ce script peut créer

| # Ajoute un compte dans la base locale du système # à partir de la saisie du nom et de la description  $local=[ADSI]"WinNT://." $fichier = "C:\testPowerShell\listeCompte.txt" $compte=[ADSI]"WinNT://./$nom"  if (Test-Path $fichier){  $colLignes = Get-Content $fichier    foreach ($ligne in $colLignes){  $tabCompte = $ligne.Split("/")  $username = $tabCompte[0]  $description = $tabCompte[2]   if (!$compte.path){  $utilisateur=$local.create("user",$username)  $utilisateur.InvokeSet("Description",$description)  $utilisateur.CommitChanges()  Write-Host "$username ajouté"  }  else{  Write-Host "$nom existe déjà"  }  } } else {  Write-Host "$fichier pas trouvé" } |
| --- |

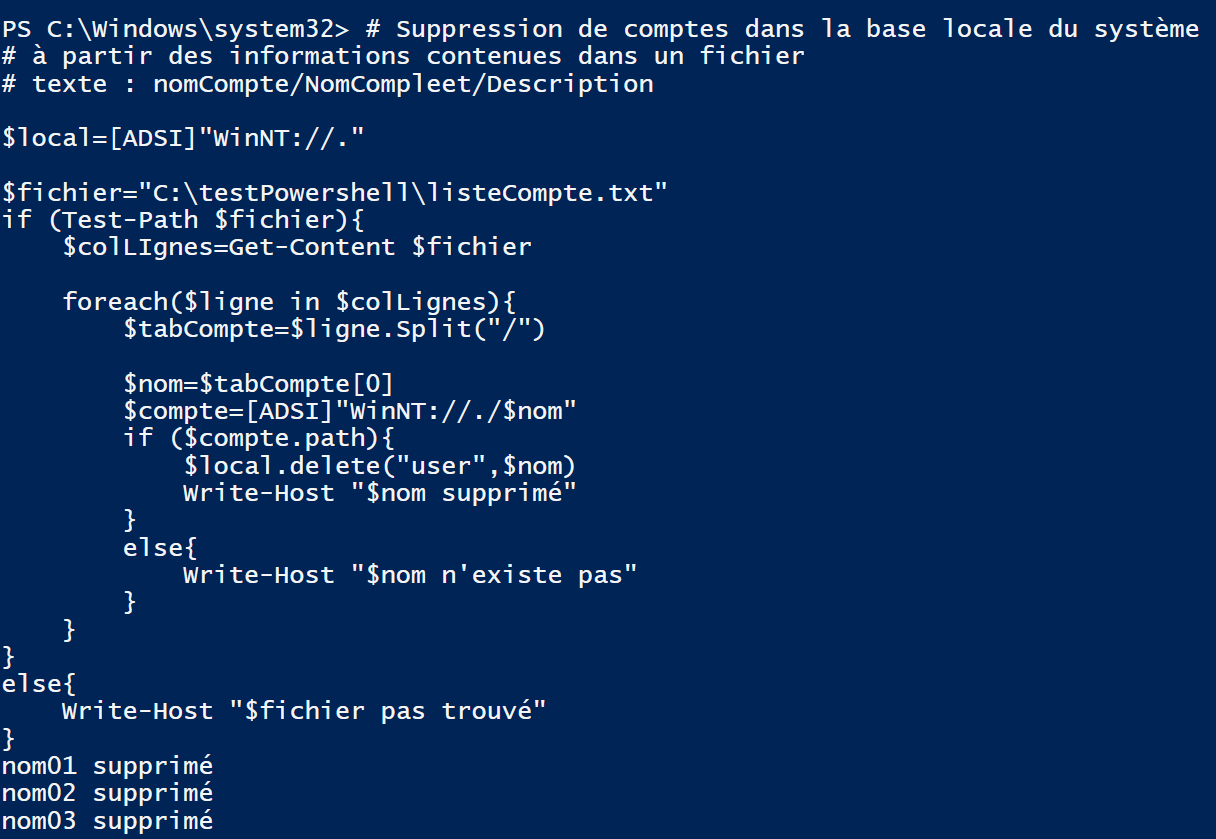
****

Dans le cas où on essaie de rajouter les utilisateurs qui ont les même prénoms le code remet des messages d’erreurs

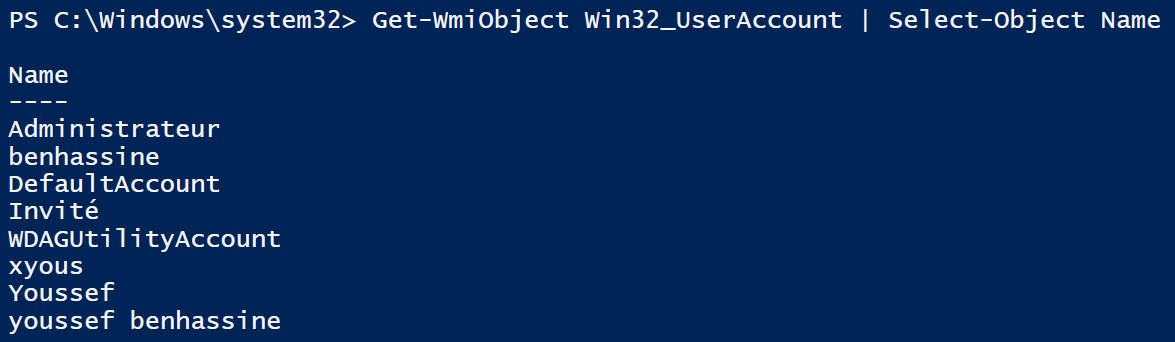
****

**Ex 4: Suppressions de comptes utilisateurs**

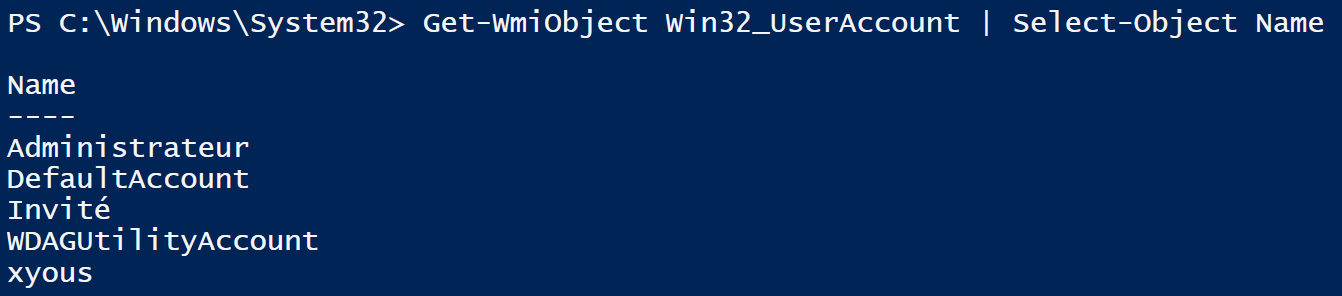
| # Suppression de comptes dans la base locale du système # à partir des informations contenues dans un fichier # texte : nomCompte/NomCompleet/Description  $local=[ADSI]"WinNT://."  $fichier="C:\testPowershell\listeCompte.txt" if (Test-Path $fichier){  $colLIgnes=Get-Content $fichier   foreach($ligne in $colLignes){  $tabCompte=$ligne.Split("/")    $nom=$tabCompte[0]   $compte=[ADSI]"WinNT://./$nom"  if ($compte.path){  $local.delete("user",$nom)  Write-Host "$nom supprimé"  }  else{  Write-Host "$nom n'existe pas"  }  } } else{  Write-Host "$fichier pas trouvé" } |
| --- |



La liste des utilisateurs restants:



| # Suppression de comptes dans la base locale du système # à partir des informations contenues dans un fichier # texte : nomCompte/NomCompleet/Description  $local=[ADSI]"WinNT://."  $fichier="C:\testPowershell\listeCompte.txt" if (Test-Path $fichier){  $colLignes=Get-Content $fichier   foreach($ligne in $colLignes){  $tabCompte=$ligne.Split("/")    $description = $tabCompte[2]       $comptes = $local.Children | Where-Object { $\_.SchemaClassName -eq 'User' }    foreach ($compte in $comptes) {    if ($compte.Description -eq $description) {  $nom = $compte.Name  $local.delete("user",$nom)  Write-Host "$nom supprimé"  }  }  } } else{  Write-Host "$fichier pas trouvé" } |
| --- |



Les comptes ont bien étés supprimés