



Scripts

Table des matières

| | |
|------------------------------------------|----|
| A- Création des UOs..... | 2 |
| B- Création utilisateurs..... | 4 |
| C- Création des groupes..... | 6 |
| D- Création des mots de passes..... | 7 |
| E- Ajout des utilisateurs à la main..... | 8 |
| F - Suppression des utilisateurs..... | 10 |

A- Création des UOs

```

$CSVFile = "C:\Program` Files\WindowsPowerShell\Programmes\Liste_Salariés_GSB_Siège.csv"
$CSVData = Import-Csv -Path $CSVFile -Delimiter ";" -Encoding Default

New-ADOrganizationalUnit "UTILISATEURS" -Path "DC=galaxy-swiss,DC=com"
Write-Output "L'Unité Organisationnelle UTILISATEURS a été créée"
Write-Output ""
Foreach ($UO in $CSVData)
{
    $NomUO = $UO.SERVICE
    $NomUO =
[Text.Encoding]::ASCII.GetString([Text.Encoding]::GetEncoding("Cyrillic").GetBytes($NomUO
))
    $NomUO = $NomUO -replace '[ ]','- '

    if (Get-ADOrganizationalUnit -Filter {Name -eq $NomUO})
    {
    }
    else
    {
        New-ADOrganizationalUnit "$NomUO" -Path "OU=UTILISATEURS,DC=galaxy-swiss,DC=com"
        Write-Output "L'Unité Organisationnelle $NomUO a été créée"
        Write-Output ""
    }
}

New-ADOrganizationalUnit "MATERIEL" -Path "DC=galaxy-swiss,DC=com"
Write-Output "L'Unité Organisationnelle MATERIEL a été créée"
Write-Output ""

New-ADOrganizationalUnit "PC" -Path "OU=MATERIEL,DC=galaxy-swiss,DC=com"
Write-Output "L'Unité Organisationnelle PC a été créée"
Write-Output ""

Foreach ($UO in $CSVData)
{
    $NomUOPC = $UO.SERVICE + "-PC"
    $NomUOPC =
[Text.Encoding]::ASCII.GetString([Text.Encoding]::GetEncoding("Cyrillic").GetBytes($NomUO
PC))
    $NomUOPC = $NomUOPC -replace '[ ]','- '

    if (Get-ADOrganizationalUnit -Filter {Name -eq $NomUOPC})
    {
    }
    else
    {
        New-ADOrganizationalUnit "$NomUOPC" -Path "OU=PC,OU=MATERIEL,DC=galaxy-
swiss,DC=com"
        Write-Output "L'Unité Organisationnelle $NomUOPC a été créée"
        Write-Output ""
    }
}
New-ADOrganizationalUnit "IMPRIMANTES" -Path "OU=Materiel,DC=galaxy-swiss,DC=com"
Foreach ($UO in $CSVData)
{

```

```
$NomUOImprim = $UO.SERVICE + "-Imprim"
$NomUOImprim =
[Text.Encoding]::ASCII.GetString([Text.Encoding]::GetEncoding("Cyrillic").GetBytes($NomUO
Imprim))
$NomUOImprim = $NomUOImprim -replace '[ ]','-'

if (Get-ADOrganizationalUnit -Filter {Name -eq $NomUOImprim})
{
}
else
{
    New-ADOrganizationalUnit "$NomUOImprim" -Path
"OU=IMPRIMANTES,OU=MATERIEL,DC=galaxy-swiss,DC=com"
    Write-Output "L'Unité Organisationnelle $NomUOImprim a été créée"
    Write-Output ""
}
}

New-ADOrganizationalUnit "GROUPES" -Path "DC=galaxy-swiss,DC=com"
Write-Output "L'Unité Organisationnelle GROUPE a été créée"
Write-Output ""
```

B- Création utilisateurs

```

$CSVFile = Read-Host "Merci de Rentrer le chemin avec le fichier csv d'utilisateurs à
créer " # "C:\Program Files\WindowsPowerShell\Programmes\Liste_Salariés_GSB_Siège.csv"
$CSVData = Import-Csv -Path $CSVFile -Delimiter ";" -Encoding Default

Foreach ($Utilisateur in $CSVData)
{
    $UtilisateurPrenom = $Utilisateur.PRENOM
    $UtilisateurNom = $Utilisateur.NOM
    $UtilisateurLogin = $UtilisateurNom + $UtilisateurPrenom.Substring(0,3)
    $UtilisateurLogin = $UtilisateurLogin.ToLower()
    $UtilisateurLogin =
[Text.Encoding]::ASCII.GetString([Text.Encoding]::GetEncoding("Cyrillic").GetBytes($Utili
sateurLogin))
    $UtilisateurLogin = $UtilisateurLogin -replace '[ ]',' '
    $UtilisateurCpt = "GALAXY-SWISS\" + $UtilisateurLogin
    $UtilisateurEmail = "$UtilisateurLogin@galaxy-swiss.com"
    $UtilisateurMotDePasse = C:\Program` Files\WindowsPowerShell\Programmes\
Creation_mdp.ps1
    $UtilisateurDateEmbau = "Date d'embauche: " + $Utilisateur.'DATE EMBAUCHE'
    $UtilisateurFonction = $Utilisateur.POSTE
    $UtilisateurService = $Utilisateur.SERVICE
    $UtilisateurAdresse = $Utilisateur.ADRESSE
    $UtilisateurCP = $Utilisateur.CP
    $UtilisateurVille = $Utilisateur.VILLE
    $UtilisateurE = "GSB"
    $UtilisateurPath = "Logon.bat"
    $UtilisateurHD = "U:\PRIVE\"
    $UtilisateurPart = "U:\PRIVE\" + $UtilisateurLogin
    $CheminProfil = "\\ADLAB\Profil$\\" + $UtilisateurLogin
    $GroupeAD = "CN=UtilisateursAD,OU=GROUPES,DC=galaxy-swiss,DC=com"

    # Vérifier la présence de l'utilisateur dans l'AD
    if (Get-ADUser -Filter {SamAccountName -eq $UtilisateurLogin})
    {
        Write-Warning "L'identifiant $UtilisateurLogin existe déjà dans l'AD"
    }
    else
    {
        $CheminUO = $UtilisateurService
        $CheminUO =
[Text.Encoding]::ASCII.GetString([Text.Encoding]::GetEncoding("Cyrillic").GetBytes($Chemi
nUO))
        $CheminUO = $CheminUO -replace '[ ]','-'
        $Chemin = "OU=" + $CheminUO + ",OU=UTILISATEURS,DC=galaxy-swiss,DC=com"

        $GroupeUtil = "CN=" + $CheminUO + ",OU=GROUPES,DC=galaxy-swiss,DC=com"

        New-Item -Name $UtilisateurLogin -Path $UtilisateurHD -ItemType Directory

        New-ADUser -Name "$UtilisateurNom $UtilisateurPrenom" `
            -DisplayName "$UtilisateurNom $UtilisateurPrenom" `
            -GivenName $UtilisateurPrenom `
            -Surname $UtilisateurNom `
            -SamAccountName $UtilisateurLogin `
            -UserPrincipalName "$UtilisateurLogin@galaxy-swiss.com" `
            -EmailAddress $UtilisateurEmail `
            -Description $UtilisateurDateEmbau `

```

```

        -Title $UtilisateurFonction `
        -Department $UtilisateurService `
        -Company $UtilisateurE `
        -StreetAddress $UtilisateurAdresse `
        -PostalCode $UtilisateurCP `
        -City $UtilisateurVille `
        -Path $Chemin `
        -ProfilePath $CheminProfil `
        -ScriptPath $UtilisateurPath `
        -AccountPassword(ConvertTo-SecureString $UtilisateurMotDePasse -
AsPlainText -Force) `
        -ChangePasswordAtLogon $true `
        -Enabled $true

    Add-AdGroupMember -Identity $GroupeUtil `
        -Members $UtilisateurLogin

    Add-AdGroupMember -Identity $GroupeAD `
        -Members $UtilisateurLogin

    #Droits répertoire
    $acl = Get-Acl $UtilisateurPart
    $object = New-Object
System.Security.AccessControl.FileSystemAccessRule("$UtilisateurCpt","Modify","Allow")
    $acl.SetAccessRule($object)
    $acl | Set-Acl $UtilisateurPart

    Write-Output "Création de l'utilisateur : $UtilisateurLogin ($UtilisateurNom
$UtilisateurPrenom)"
    Write-Output "Son mot de passe est : $utilisateurMotDePasse"
    Write-Output "Il est dans le groupe : $UtilisateurService"
    Write "$UtilisateurLogin = $UtilisateurMotDePasse" >> U:\Utilisateurs.txt
}
}

```

C- Création des groupes

```
$CSVFile = "C:\Program` Files\WindowsPowerShell\Programmes\Liste_Salariés_GSB_Siège.csv"
$CSVData = Import-Csv -Path $CSVFile -Delimiter ";" -Encoding Default

Foreach ($Groupe in $CSVData)
{
    $Groupe = $Groupe.SERVICE
    $Groupe = $Groupe
    $Groupe =
[Text.Encoding]::ASCII.GetString([Text.Encoding]::GetEncoding("Cyrillic").GetBytes($Groupe))
    $Groupe = $Groupe -replace '[ ]','-'

# Vérifier la présence de l'utilisateur dans l'AD
    if (Get-ADGroup -Filter {SamAccountName -eq $Groupe})
    {
    }
    else
    {
        New-ADGroup -Name "$Groupe" `
                    -DisplayName "$Groupe" `
                    -GroupCategory Security `
                    -GroupScope Global `
                    -Path "OU=GROUPES,DC=galaxy-swiss,DC=com"

        Write-Output "Création du groupe : $Groupe"
    }
}
}
```

D- Création des mots de passes

```

function New-RandomPassword
{
    [CmdletBinding()]
    Param
    (
        [Parameter(Mandatory=$true)][int]$Length,
        [Parameter(Mandatory=$false)][int]$Uppercase=1,
        [Parameter(Mandatory=$false)][int]$Digits=2,
        [Parameter(Mandatory=$false)][int]$SpecialCharacters=2
    )
    Begin
    {
        $Lowercase = $Length - $SpecialCharacters - $Uppercase - $Digits
        $ArrayLowerCharacters =
@('a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v'
,'w','x','y','z')
        $ArrayUpperCharacters =
@('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'
,'W','X','Y','Z')
        $ArraySpecialCharacters = @('_', '*', '$', '%', '#', '?', '!', '-')
    }
    Process
    {
        [string]$NewPassword = $ArrayLowerCharacters | Get-Random -Count $Lowercase
        $NewPassword += 0..9 | Get-Random -Count $Digits
        $NewPassword += $ArrayUpperCharacters | Get-Random -Count $Uppercase
        $NewPassword += $ArraySpecialCharacters | Get-Random -Count
$SpecialCharacters

        $NewPassword = $NewPassword.Replace(' ', '')

        $characterArray = $NewPassword.ToCharArray()
        $scrambledStringArray = $characterArray | Get-Random -Count
$characterArray.Length
        $NewRandomPassword = -join $scrambledStringArray

    }
    End
    {
        Return $NewRandomPassword
    }
}

New-RandomPassword -Length 8

```

E- Ajout des utilisateurs à la main

```

$UtilisateurPrenom = Read-Host "Merci de Rentrer le Prénom de l'Utilisateur à Créer"
$UtilisateurNom = Read-Host "Merci de Rentrer le Nom de l'Utilisateur à Créer"
$UtilisateurLogin = $UtilisateurPrenom + ($UtilisateurNom).Substring(0,3)
$UtilisateurLogin = $UtilisateurLogin.ToLower()
$UtilisateurLogin =
[Text.Encoding]::ASCII.GetString([Text.Encoding]::GetEncoding("Cyrillic").GetBytes($Utili
sateurLogin))
$UtilisateurLogin = $UtilisateurLogin -replace '[ ]',' '
$UtilisateurEmail = "$UtilisateurLogin@galaxy-swiss.com"
$UtilisateurMotDePasse = C:\Program` Files\WindowsPowerShell\Programmes\Creation_mdp.ps1
$UtilisateurDateE = Read-Host "Merci de Rentrer la Date d'embauche de l'Utilisateur à
Créer"
$UtilisateurDateEmbau = "Date d'embauche : " + $UtilisateurDateE
$UtilisateurFonction = Read-Host "Merci de Rentrer la Fonction/Poste de l'Utilisateur à
Créer"
$UtilisateurService = Read-Host "Merci de Rentrer le Service de l'Utilisateur à Créer"
$UtilisateurAdresse = Read-Host "Merci de Rentrer l'Adresse de l'Utilisateur à Créer"
$UtilisateurCP = Read-Host "Merci de Rentrer le Code Postal de l'Utilisateur à Créer"
$UtilisateurVille = Read-Host "Merci de Rentrer la Ville de l'Utilisateur à Créer"
$UtilisateurE = "GSB"
$UtilisateurPath = "Logon.bat"
$UtilisateurHD = "U:\PRIVE"
$UtilisateurPart = "U:\PRIVE\$UtilisateurLogin"
$CheminProfil = "\\ADLAB\Profil\$\" + $UtilisateurLogin
$GroupeAD = "CN=UtilisateursAD,OU=GROUPES,DC=galaxy-swiss,DC=com"

# Vérifier la présence de l'utilisateur dans l'AD
if (Get-ADUser -Filter {SamAccountName -eq $UtilisateurLogin})
{
    Write-Warning "L'identifiant $UtilisateurLogin existe déjà dans l'AD"
}
else
{
    $CheminUO = $UtilisateurService
    $CheminUO =
[Text.Encoding]::ASCII.GetString([Text.Encoding]::GetEncoding("Cyrillic").GetBytes($Chemi
nUO))
    $CheminUO = $CheminUO -replace '[ ]','- '
    $Chemin = "OU=" + $CheminUO + ",OU=UTILISATEURS,DC=galaxy-swiss,DC=com"

    $GroupeUtil = "CN=" + $CheminUO + ",OU=GROUPES,DC=galaxy-swiss,DC=com"

    New-Item -Name $UtilisateurLogin -Path $UtilisateurHD -ItemType Directory
    $acl = Get-Acl "U:\PRIVE\$UtilisateurLogin"

    New-ADUser -Name "$UtilisateurNom $UtilisateurPrenom" `
        -DisplayName "$UtilisateurNom $UtilisateurPrenom" `
        -GivenName $UtilisateurPrenom `
        -Surname $UtilisateurNom `
        -SamAccountName $UtilisateurLogin `
        -UserPrincipalName "$UtilisateurLogin@galaxy-swiss.com" `
        -EmailAddress $UtilisateurEmail `
        -Description $UtilisateurDateEmbau `
        -Title $UtilisateurFonction `
        -Department $UtilisateurService `
        -Company $UtilisateurE `

```



```
-StreetAddress $UtilisateurAdresse `
-PostalCode $UtilisateurCP `
-City $UtilisateurVille `
-Path $Chemin `
-ProfilePath $CheminProfil `
-ScriptPath $UtilisateurPath `
-AccountPassword(ConvertTo-SecureString $UtilisateurMotDePasse -
AsPlainText -Force) `
-ChangePasswordAtLogon $true `
-Enabled $true

Add-AdGroupMember -Identity $GroupeUtil `
                  -Members $UtilisateurLogin

Add-AdGroupMember -Identity $GroupeAD `
                  -Members $UtilisateurLogin

#Droits répertoire
$acl = Get-Acl $UtilisateurPart
$object = New-Object
System.Security.AccessControl.FileSystemAccessRule("$UtilisateurCpt","Modify","Allow")
$acl.SetAccessRule($object)
$acl | Set-Acl $UtilisateurPart

Write-Output "Création de l'utilisateur : $UtilisateurLogin ($UtilisateurNom
$UtilisateurPrenom)"
Write-Output "Son mot de passe est : $utilisateurMotDePasse"
Write-Output "Il est dans le groupe : $UtilisateurService"
Write "$UtilisateurLogin = $UtilisateurMotDePasse" >> U:\Utilisateurs.txt
}
```

F - Suppression des utilisateurs

```
$UtilisateurLogin = Read-Host "Merci de Rentrer le Login l'Utilisateur à supprimer"  
$UtilisateurRep = "\\ADLAB\PRIVE$" + $UtilisateurLogin
```

```
Remove-ADUser -Identity $UtilisateurLogin  
Remove-Item $UtilisateurRep
```

G – Création des partages

```
New-SmbShare -Name "PRIVE$" -Path "U:\PRIVE"  
New-SmbShare -Name "Profil$" -Path "U:\Profil"  
New-SmbShare -Name "Commun$" -Path "U:\Commun"  
New-SmbShare -Name "Ressources$" -Path "U:\Ressources"
```

```
Grant-SmbShareAccess -Name Profil$ -AccountName "UtilisateursAD" -AccessRight Change -  
Force  
Grant-SmbShareAccess -Name Ressources$ -AccountName "Technique" -AccessRight Change -  
Force  
Grant-SmbShareAccess -Name Commun$ -AccountName "UtilisateursAD" -AccessRight Change -  
Force  
Grant-SmbShareAccess -Name PRIVE$ -AccountName "UtilisateursAD" -AccessRight Change -  
Force
```

```
Grant-SmbShareAccess -Name Ressources$ -AccountName "UtilisateursAD" -AccessRight Read -  
Force
```

```
#Droits sur les dossiers partagés  
$UtilisateurGrp = "GALAXY-SWISS\UtilisateursAD"  
$DossPriv = "U:\PRIVE"  
$DossProf = "U:\Profil"  
  
$acl = Get-Acl $DossPriv  
$object = New-Object  
System.Security.AccessControl.FileSystemAccessRule("$UtilisateurGrp", "Modify", "Allow")  
$acl.SetAccessRule($object)  
$acl | Set-Acl $DossPriv  
  
$acl = Get-Acl $DossProf  
$object = New-Object  
System.Security.AccessControl.FileSystemAccessRule("$UtilisateurGrp", "Modify", "Allow")  
$acl.SetAccessRule($object)  
$acl | Set-Acl $DossProf
```

J – Exécution des scripts

```
Write-Output "----- Que voulez-vous faire? -----"
Write-Output ""
Write-Output ""
Write-Output "1. Créer les UOs, groupes, utilisateurs et partages"
Write-Output ""
Write-Output "2. Créer un utilisateur à la main"
Write-Output ""
Write-Output "3. Créer des utilisateurs à partir d'un fichier csv"
Write-Output ""
Write-Output "4. Supprimer un utilisateur"
Write-Output ""
Write-Output ""

$Choix = Read-Host "Ecrire le numéro correspondant"

if ($Choix = 1)
{
    C:\Program` Files\WindowsPowerShell\Programmes\Creation_UO.ps1

    C:\Program` Files\WindowsPowerShell\Programmes\Creation_groupes.ps1

    C:\Program` Files\WindowsPowerShell\Programmes\Creation_rep_partages.ps1

    C:\Program` Files\WindowsPowerShell\Programmes\Creation_partages.ps1

    C:\Program` Files\WindowsPowerShell\Programmes\Creation_user.ps1
}
elseif ($Choix = 2)
{
    C:\Program` Files\WindowsPowerShell\Programmes\Creation_user_main.ps1
}
elseif ($Choix = 3)
{
    C:\Program` Files\WindowsPowerShell\Programmes\Creation_user.ps1
}
elseif ($Choix = 4)
{
    C:\Program` Files\WindowsPowerShell\Programmes\Suppr_user.ps1
}
```
