

# Gérer un AD avec du powershell



Cette petite fiche concerne la gestion d'un AD avec Powershell.

## Vérifier un fichier .CSV

[snippet.powershell](#)

```
Import-Csv C:\Users\Administrateur\Desktop\Scripts\utilisateurs.csv ";"  
| Format-Table
```

## Créer des OU et des sous-OU

### Exemple de fichier .CSV

[Téléchargeable ici.](#)

### Le script de création des OU :

[snippet.powershell](#)

```
Import-Module activedirectory  
  
$ADOU = Import-csv 'C:\Users\Administrateur\Desktop\Scripts\ou.csv'  
  
foreach ($ou in $ADou) {  
  
    $name = $ou.name  
    $path = $ou.path  
  
    New-ADOrganizationalUnit `   
    -Name $name `   
    -path $path `
```

```
write-Host "OU créée." -ForegroundColor Cyan  
}
```

## Powershell sur les utilisateurs

### Lister les utilisateurs existant dans une OU

[snippet.powershell](#)

```
Get-ADUser -Filter * -SearchBase  
"OU=Utilisateurs,OU=Megaproduction,DC=dom,DC=megaprod,DC=lan" `\  
-Properties Name,GivenName,Surname,EmailAddress,Title | `\  
Select-Object Name,GivenName,Surname,EmailAddress,Title |  
Ft
```

Name	GivenName	Surname	EmailAddress	Title
Emily Oregans	Emily	Oregans	e.oregans@megaproduction	
Alan Parsons	Alan	Parsons	a.parsons@megaproduction	
Rick Davies	Rick	Davies	r.davies@megaproduction	
Roger Hodgson	Roger	Hodgson	r.hodgson@megaproduction	
David Bowie	David	Bowie	d.bowie@megaproduction	
Stevie Nicks	Stevie	Nicks	s.nicks@megaproduction	
Bernard Edwards	Bernard	Edwards	b.edwards@megaproduction	
Marc Cerrone	Marc	Cerrone	m.cerrone@megaproduction	
David Paich	David	Paich	d.paich@megaproduction	
Christopher Cross	Christopher	Cross	c.cross@megaproduction	

### Création d'utilisateurs à partir d'un .CSV

#### Exemple de fichier .CSV

[Téléchargeable ici.](#)

#### Le script de création utilisateurs :

[snippet.powershell](#)

```
Import-Module ActiveDirectory  
  
$ADUsers = Import-Csv  
C:\Users\Administrateur\Desktop\Scripts\utilisateurs.csv  
$UPN = "dom.megaprod.lan"
```

```
foreach ($User in $ADUsers) {

    $username = $User.username
    $password = $User.password
    $firstname = $User.firstname
    $lastname = $User.lastname
    $initials = $User.initials
    $OU = $User.ou
    $email = $User.email
    $streetaddress = $User.streetaddress
    $city = $User.city
    $zipcode = $User.zipcode
    $state = $User.state
    $country = $User.country
    $telephone = $User.telephone
    $jobtitle = $User.jobtitle
    $company = $User.company
    $department = $User.department

    if (Get-ADUser -F { SamAccountName -eq $username }) {
        Write-Warning "Compte utilisateur déjà existant."
    }
    else {
        New-ADUser `
            -SamAccountName $username `
            -UserPrincipalName "$username@$UPN" `
            -Name "$firstname $lastname" `
            -GivenName $firstname `
            -Surname $lastname `
            -Initials $initials `
            -Enabled $True `
            -DisplayName "$lastname, $firstname" `
            -Path $OU `
            -City $city `
            -PostalCode $zipcode `
            -Country $country `
            -Company $company `
            -State $state `
            -StreetAddress $streetaddress `
            -OfficePhone $telephone `
            -EmailAddress $email `
            -Title $jobtitle `
            -Department $department `
            -AccountPassword (ConvertTo-secureString $password -
AsPlainText -Force) -ChangePasswordAtLogon $True
        Write-Host "Utilisateur créé." -ForegroundColor Cyan
    }
}

Read-Host -Prompt "Appuyer sur ENTRER pour terminer."
```

## Création des groupes

### Exemple de fichier .CSV

[Téléchargeable ici.](#)

### Le script de création des groupes :

[snippet.powershell](#)

```
Import-Module ActiveDirectory

$groups = Import-Csv
'C:\Users\Administrateur\Desktop\Scripts\groupes.csv'

foreach ($group in $groups) {

    $groupProps = @{

        Name           = $group.name
        Path           = $group.path
        GroupScope     = $group.scope
        GroupCategory = $group.category
        Description    = $group.description

    }

    New-ADGroup @groupProps
    Write-Host "Groupe créé." -ForegroundColor Cyan

}

Read-Host -Prompt "Appuyer sur ENTRER pour terminer."
```

## Mettre le groupe local dans le groupe global

### Exemple de fichier .CSV

[Téléchargeable ici.](#)

## Le script d'ajout de groupe local dans un groupe global :

[snippet.powershell](#)

```
Import-Module ActiveDirectory

$list = Import-Csv "C:\Users\Administrateur\Desktop\Scripts\gl_gg.csv"

foreach ( $Group in $List ) {
    foreach ( $MemberOf in $Group.memberof -split ", " ) {
        Add-ADGroupMember -Identity $MemberOf -Members $Group.group
    }
    write-Host "Groupe $Group ajouté dans le groupe global $MemberOf." -ForegroundColor Cyan
}
```

## Ajouter un utilisateur dans un groupe

### Exemple de fichier .CSV

[Téléchargeable ici.](#)

## Le script d'ajout d'utilisateurs dans des groupes :

[snippet.powershell](#)

```
Import-Module ActiveDirectory

$list = Import-Csv
"C:\Users\Administrateur\Desktop\Scripts\utilisateurs_groups.csv"

foreach ($User in $List) {

    $UserSam = $User.SamAccountName
    $Groups = $User.Group

    $ADUser = Get-ADUser -Filter "SamAccountName -eq '$UserSam'" |
Select-Object SamAccountName
    $ADGroups = Get-ADGroup -Filter * | Select-Object
DistinguishedName, SamAccountName

    if ($ADUser -eq $null) {
        Write-Host "$UserSam utilisateur non existant." -
ForegroundColor Red
    }
}
```

```
        Continue
    }

    if ($Groups -eq $null) {
        Write-Host "$UserSam aucun groupe spécifié pour cet
utilisateur." -ForegroundColor Yellow
        Continue
    }

    $ExistingGroups = Get-ADPrincipalGroupMembership $UserSam | Select-
Object DistinguishedName, SamAccountName

    foreach ($Group in $Groups.Split(';')) {

        if ($ADGroups.SamAccountName -notcontains $Group) {
            Write-Host "$Le groupe n'existe pas." -ForegroundColor Red
            Continue
        }

        if ($ExistingGroups.SamAccountName -eq $Group) {
            Write-Host "$UserSam existe déjà dans le groupe $Group" -
ForegroundColor Yellow
        }
        else {

            Add-ADGroupMember -Identity $Group -Members $UserSam
            Write-Host "L'utilisateur $UserSam a été ajouté dans le
groupe $Group" -ForegroundColor Green
        }
    }
}
```

## Création de la structure des dossiers

*Fonctionnalité :*

- Création de la structure des dossiers - Suppression de l'héritage - Suppression des utilisateurs indésirables - Ajout des groupes et définitions des permissions à partir d'un .csv

### Exemple de fichier .CSV

[Téléchargeable ici.](#)

## Le script d'ajout d'utilisateurs dans des groupes :

snippet.powershell

```
Set-Location "C:\"  
write-Host "Créations de la structure des dossiers..." -ForegroundColor  
Cyan  
  
$Folders = Import-Csv  
"C:\Users\Administrateur\Desktop\Scripts\folder_structure.csv"  
  
ForEach ($Folder in $Folders) {  
  
    if (Test-Path -Path $Folder.Name) {  
        write-Host "Le dossier existe déjà !" -ForegroundColor Red  
    } else {  
        New-Item $Folder.Name -type directory  
        write-Host "Création du dossier $Folder" -ForegroundColor Green  
    }  
  
    echo "-----"  
    echo 'Dossier :' $Folder.Name  
  
    write-Host "Supression de l'heritage" -ForegroundColor Cyan  
  
    $acl = Get-ACL -Path $Folder.Name  
    $acl.SetAccessRuleProtection($True, $True)  
    Set-Acl -Path $Folder.Name -AclObject $acl  
    write-Host "OK!" -ForegroundColor Green  
  
    write-Host "Supression des utilisateurs non-désirés" -  
ForegroundColor Cyan  
  
    $acl = Get-ACL -Path $Folder.Name  
    icacls C:\entreprise /remove 'CREATEUR PROPRIETAIRE' /t  
    $usersid = New-Object System.Security.Principal.Ntaccount  
    ("BUILTIN\Utilisateurs")  
    $acl.PurgeAccessRules($usersid)  
    $acl | Set-Acl -Path $Folder.Name  
    write-Host "OK!" -ForegroundColor Green  
  
    write-Host "Ajouter les ACL sur les dossiers" -ForegroundColor Cyan  
  
    echo $Folder.Group  
    echo $Folder.ACL  
    $acl = Get-Acl -Path $Folder.Name  
    $AccessRule = New-Object  
System.Security.AccessControl.FileSystemAccessRule($Folder.Group,$Folde  
r.ACL,"Allow")
```

```
$acl.SetAccessRule($AccessRule)
$acl | Set-Acl -Path $Folder.Name
write-Host "OK!" -ForegroundColor Green
}
```

## Mes sources

1. <https://www.it-connect.fr/chapitres/creer-des-utilisateurs-dans-lad-a-partir-dun-csv/>
2. <https://www.alitajran.com/add-users-to-multiple-groups-powershell/>
3. <https://community.spiceworks.com/topic/1286638-insert-global-security-group-into-domain-local-groups-with-csv>
4. <https://blog.netwrix.fr/2018/12/12/comment-gerer-les-listes-de-controle-daces-acl-au-systeme-de-fichiers-avec-les-scripts-powershell/>
5. [https://petri.com/how-to-use-powershell-to-manage-folder-permissions/#Modifying\\_files\\_and\\_folder\\_permissions\\_with\\_Get-Acl\\_and\\_Set-Acl](https://petri.com/how-to-use-powershell-to-manage-folder-permissions/#Modifying_files_and_folder_permissions_with_Get-Acl_and_Set-Acl)
6. <https://stackoverflow.com/questions/36103821/how-to-remove-acls-on-remote-disk>

From:  
<https://wiki.stoneset.fr/> - **StoneSet - Documentations**

Permanent link:  
<https://wiki.stoneset.fr/doku.php?id=wiki:windows:scripting:adpowershell&rev=1669021244>

Last update: **2022/11/21 10:00**

