

Script create group

```
# Liste des groupes à créer

$groupes = @(
    @{Nom = "Mark"},
    @{Nom = "Dev"},
    @{Nom = "Com"},
    @{Nom = "Rh"}
)

# Créer chaque groupe s'il n'existe pas

foreach ($groupe in $groupes) {
    $nomGroupe = $groupe.Nom
    $existe = Get-LocalGroup -Name $nomGroupe -ErrorAction SilentlyContinue

    if ($existe) {
        Write-Host "Le groupe '$nomGroupe' existe déjà. Aucune action nécessaire."
        -ForegroundColor Yellow
    }
    else {
        try {
            # On ne passe pas de description si elle n'est pas définie
            New-LocalGroup -Name $nomGroupe -ErrorAction Stop
            Write-Host "Groupe '$nomGroupe' créé avec succès." -ForegroundColor Green
        }
        catch {
            Write-Host "Erreur lors de la création du groupe '$nomGroupe' : $_"
            -ForegroundColor Red
        }
    }
}

Write-Host "Vérification et création des groupes terminée." -ForegroundColor Cyan
```

Script create user

```
# Liste des utilisateurs à créer (nom uniquement)
$utilisateurs = @(
    @{Nom = "Nprisejackt"},
    @{Nom = "Pborjon"},
    @{Nom = "Bibm"},
    @{Nom = "Rair"}
)

# Mot de passe par défaut pour tous les utilisateurs
$motDePasseParDefaut = "MotDePasse123!" # Remplace par un mot de passe sécurisé

# Créer chaque utilisateur s'il n'existe pas
foreach ($utilisateur in $utilisateurs) {
    $nomUtilisateur = $utilisateur.Nom
    $existe = Get-LocalUser -Name $nomUtilisateur -ErrorAction SilentlyContinue

    if ($existe) {
        Write-Host "L'utilisateur '$nomUtilisateur' existe déjà. Aucune action nécessaire." -ForegroundColor Yellow
    }
    else {
        try {
            # Créer un objet SecureString pour le mot de passe
            $securePassword = ConvertTo-SecureString $motDePasseParDefaut -AsPlainText -Force

            # Créer l'utilisateur (sans description)
            New-LocalUser -Name $nomUtilisateur -Password $securePassword -ErrorAction Stop
            Write-Host "Utilisateur '$nomUtilisateur' créé avec succès." -ForegroundColor Green
        }
        catch {
            Write-Host "Erreur lors de la création de l'utilisateur '$nomUtilisateur' : $_" -ForegroundColor Red
        }
    }
}

Write-Host "Vérification et création des utilisateurs terminée." -ForegroundColor Cyan
```

Script add user to group + Script add group to UO

```
# =====
# Script : Création utilisateurs + groupes + UO
# =====

Import-Module ActiveDirectory

# =====
# Liste des utilisateurs et de leurs groupes
# =====
$utilisateurs = @(
    @{Nom = "Nprisejackt"; Groupe = "Mark"},
    @{Nom = "Pborjon"; Groupe = "Dev"},
    @{Nom = "Bibm"; Groupe = "Com"},
    @{Nom = "Rair"; Groupe = "Rh"}
)

# Mot de passe par défaut (à modifier par un mot de passe sécurisé)
$motDePasseParDefaut = "MotDePasse123!"

# =====
#①Création des groupes
# =====
$groupes = $utilisateurs | ForEach-Object { $_.Groupe } | Sort-Object -Unique

foreach ($nomGroupe in $groupes) {
    $existe = Get-ADGroup -Filter "Name -eq '$nomGroupe'" -ErrorAction SilentlyContinue
    if (-not $existe) {
        try {
            New-ADGroup -Name $nomGroupe -GroupScope Global -GroupCategory Security
            -ErrorAction Stop
            Write-Host "Groupe '$nomGroupe' créé avec succès." -ForegroundColor Green
        } catch {
            Write-Host "Erreur lors de la création du groupe '$nomGroupe' : $_"
            -ForegroundColor Red
        }
    } else {
        Write-Host "Le groupe '$nomGroupe' existe déjà." -ForegroundColor Yellow
    }
}

# =====
#②Création des utilisateurs et ajout aux groupes
# =====
```

Banuls Alexandre

```
foreach ($utilisateur in $utilisateurs) {
    $nomUtilisateur = $utilisateur.Nom
    $nomGroupe = $utilisateur.Groupe
    $existe = Get-ADUser -Filter "SamAccountName -eq '$nomUtilisateur'" -ErrorAction
    SilentlyContinue

    if ($existe) {
        Write-Host "L'utilisateur '$nomUtilisateur' existe déjà." -ForegroundColor Yellow
    } else {
        try {
            $securePassword = ConvertTo-SecureString $motDePasseParDefaut -AsPlainText
            -Force
            New-ADUser -Name $nomUtilisateur -SamAccountName $nomUtilisateur
            -AccountPassword $securePassword -Enabled $true -PasswordNeverExpires $false
            -ChangePasswordAtLogon $true -ErrorAction Stop
            Write-Host "Utilisateur '$nomUtilisateur' créé avec succès." -ForegroundColor Green
        } catch {
            Write-Host "Erreur lors de la création de l'utilisateur '$nomUtilisateur' : $_"
            -ForegroundColor Red
        }
    }

    # Ajout au groupe
    try {
        Add-ADGroupMember -Identity $nomGroupe -Members $nomUtilisateur -ErrorAction
        Stop
        Write-Host "Utilisateur '$nomUtilisateur' ajouté au groupe '$nomGroupe'."
        -ForegroundColor Cyan
    } catch {
        Write-Host "Erreur lors de l'ajout de '$nomUtilisateur' au groupe '$nomGroupe' : $_"
        -ForegroundColor Red
    }
}

# =====
# ③Création des UO et association des groupes
# =====

$groupesVersUO = @{
    "Mark" = "market";
    "Dev" = "devlp";
    "Com" = "communi";
    "Rh" = "ressource"
}

foreach ($groupe in $groupesVersUO.Keys) {
    $uoCible = $groupesVersUO[$groupe]

    # Vérifie si l'UO existe
```

```
$uo = Get-ADOrganizationalUnit -Filter "Name -eq '$uoCible'" -ErrorAction SilentlyContinue
if ($null -eq $uo) {
    try {
        New-ADOrganizationalUnit -Name $uoCible -ErrorAction Stop
        $uo = Get-ADOrganizationalUnit -Filter "Name -eq '$uoCible'" -ErrorAction Stop
        Write-Host "UO '$uoCible' créée avec succès." -ForegroundColor Green
    } catch {
        Write-Host "Impossible de créer l'UO '$uoCible' : $_" -ForegroundColor Red
        continue
    }
}

# Déplace le groupe dans l'UO
try {
    $groupeAD = Get-ADGroup -Identity $groupe -ErrorAction Stop
    Move-ADObject -Identity $groupeAD.DistinguishedName -TargetPath
$uo.DistinguishedName
    Write-Host "Groupe '$groupe' déplacé dans l'UO '$uoCible'." -ForegroundColor Cyan
} catch {
    Write-Host "Erreur lors du déplacement du groupe '$groupe' dans l'UO '$uoCible' : $_"
-ForegroundColor Red
}
}

# =====
# ✅ Fin du script
# =====

Write-Host "Création des utilisateurs, groupes et UO terminée." -ForegroundColor Magenta
```

Script avec CSV final

```
# =====
# Script : Création utilisateurs + groupes + UO depuis CSV
# =====

Import-Module ActiveDirectory

# =====
# Chemin vers le fichier CSV
# =====
$cheminCSV = "C:\Users\Administrateur\Documents\csv.csv"
```

Banuls Alexandre

```
# Mot de passe par défaut (identique à celui du CSV)
$motDePasseParDefaut = "Passe2!"

# =====
#① Lecture manuelle du fichier CSV
# =====
try {
    $lignes = Get-Content -Path $cheminCSV -Encoding UTF8

    # Supprimer la ligne d'en-tête et traiter les données
    $utilisateurs = @()
    for ($i = 1; $i -lt $lignes.Count; $i++) {
        $ligne = $lignes[$i].Trim("") # Supprimer les guillemets
        $colonnes = $ligne -split "`t" # Séparer par tabulation

        if ($colonnes.Count -ge 8) {
            $utilisateur = [PSCustomObject]@{
                Login = $colonnes[0].Trim()
                Prenom = $colonnes[1].Trim()
                Nom = $colonnes[2].Trim()
                Initiale = $colonnes[3].Trim()
                NomComplet = $colonnes[4].Trim()
                MotDePasse = $colonnes[5].Trim()
                Groupe1 = $colonnes[6].Trim()
                Groupe2 = $colonnes[7].Trim()
            }
            $utilisateurs += $utilisateur
        }
    }
}

Write-Host "Fichier CSV chargé avec succès. $($utilisateurs.Count) utilisateurs trouvés."
-ForegroundColor Green

# Afficher le premier utilisateur pour vérification
Write-Host "nPremier utilisateur :" -ForegroundColor Yellow
$utilisateurs[0] | Format-Table -AutoSize

} catch {
    Write-Host "Erreur lors de la lecture du fichier CSV : $_" -ForegroundColor Red
    exit
}

# =====
#② Extraction des groupes uniques
# =====
$groupesUniques = @()
foreach ($user in $utilisateurs) {
    if ($user.Groupe1 -and $user.Groupe1 -ne "") {
```

Banuls Alexandre

```
$groupesUniques += $user.Groupe1
}
if ($user.Groupe2 -and $user.Groupe2 -ne "") {
    $groupesUniques += $user.Groupe2
}
}
$groupes = $groupesUniques | Sort-Object -Unique | Where-Object { $_ -ne "" }

Write-Host "Groupes identifiés : $($groupes -join ', ')" -ForegroundColor Green

# =====
#③Création des groupes
# =====
foreach ($nomGroupe in $groupes) {
    try {
        $existe = Get-ADGroup -Filter "Name -eq '$nomGroupe'" -ErrorAction SilentlyContinue
        if (-not $existe) {
            New-ADGroup -Name $nomGroupe -GroupScope Global -GroupCategory Security
            -ErrorAction Stop
            Write-Host "Groupe '$nomGroupe' créé avec succès." -ForegroundColor Green
        } else {
            Write-Host "Le groupe '$nomGroupe' existe déjà." -ForegroundColor Yellow
        }
    } catch {
        Write-Host "Erreur avec le groupe '$nomGroupe' : $_" -ForegroundColor Red
    }
}

# =====
#④Création des utilisateurs et ajout aux groupes
# =====
foreach ($user in $utilisateurs) {
    $nomUtilisateur = $user.Login
    $prenom = $user.Prenom
    $nom = $user.Nom
    $nomComplet = $user.NomComplet

    if ([string]::IsNullOrEmpty($nomUtilisateur)) {
        Write-Host "Nom d'utilisateur vide pour : $($user.NomComplet)" -ForegroundColor Red
        continue
    }

    Write-Host "Traitement de l'utilisateur : $nomUtilisateur" -ForegroundColor Cyan

    # Liste des groupes de l'utilisateur
    $groupesUtilisateur = @()
    if ($user.Groupe1 -and $user.Groupe1 -ne "") {
        $groupesUtilisateur += $user.Groupe1
```

```
}

if ($user.Groupe2 -and $user.Groupe2 -ne "") {
    $groupesUtilisateur += $user.Groupe2
}

# Vérification si l'utilisateur existe
try {
    $existe = Get-ADUser -Filter "SamAccountName -eq '$nomUtilisateur'" -ErrorAction Stop
} catch {
    $existe = $null
}

if ($existe) {
    Write-Host "L'utilisateur '$nomUtilisateur' existe déjà." -ForegroundColor Yellow
} else {
    try {
        $securePassword = ConvertTo-SecureString $motDePasseParDefaut -AsPlainText -Force
        $userParams = @{
            Name          = $nomComplet
            SamAccountName = $nomUtilisateur
            GivenName     = $prenom
            Surname       = $nom
            DisplayName   = $nomComplet
            AccountPassword = $securePassword
            Enabled       = $true
            PasswordNeverExpires = $false
            ChangePasswordAtLogon = $true
        }
        New-ADUser @userParams -ErrorAction Stop
        Write-Host "Utilisateur '$nomUtilisateur' créé avec succès." -ForegroundColor Green
    } catch {
        Write-Host "Erreur lors de la création de l'utilisateur '$nomUtilisateur' : $_"
    } -ForegroundColor Red
    continue
}

# Ajout aux groupes
foreach ($groupe in $groupesUtilisateur) {
    if ([string]::IsNullOrEmpty($groupe)) { continue }

    try {
        Add-ADGroupMember -Identity $groupe -Members $nomUtilisateur -ErrorAction Stop
        Write-Host "Utilisateur '$nomUtilisateur' ajouté au groupe '$groupe'."
    } -ForegroundColor Cyan
```

Banuls Alexandre

```
    } catch {
        Write-Host "Erreur lors de l'ajout de '$nomUtilisateur' au groupe '$groupe' : $_"
        -ForegroundColor Red
    }
}

# =====
#⑤Création des UO et association des groupes
# =====

$groupesVersUO = @{
    "COMPTA" = "RH"
    "COMMU" = "Redac"
    "COMMERCIAL" = "Com"
    "SR" = "RS"
    "DEV" = "Dev"
}

foreach ($groupe in $groupesVersUO.Keys) {
    # Vérifier si le groupe existe avant de le déplacer
    try {
        $groupeAD = Get-ADGroup -Identity $groupe -ErrorAction Stop

        $uoCible = $groupesVersUO[$groupe]

        # Vérifie si l'UO existe
        $uo = Get-ADOrganizationalUnit -Filter "Name -eq '$uoCible'" -ErrorAction
SilentlyContinue
        if ($null -eq $uo) {
            try {
                New-ADOrganizationalUnit -Name $uoCible -ErrorAction Stop
                $uo = Get-ADOrganizationalUnit -Filter "Name -eq '$uoCible'" -ErrorAction Stop
                Write-Host "UO '$uoCible' créée avec succès." -ForegroundColor Green
            } catch {
                Write-Host "Impossible de créer l'UO '$uoCible' : $_" -ForegroundColor Red
                continue
            }
        }
    }

    # Déplace le groupe dans l'UO
    try {
        Move-ADObject -Identity $groupeAD.DistinguishedName -TargetPath
$uo.DistinguishedName
        Write-Host "Groupe '$groupe' déplacé dans l'UO '$uoCible'." -ForegroundColor Cyan
    } catch {
        Write-Host "Erreur lors du déplacement du groupe '$groupe' dans l'UO '$uoCible' :
$_" -ForegroundColor Red
    }
}
```

Banuls Alexandre

```
 } catch {
    Write-Host "Le groupe '$groupe' n'existe pas, impossible de le déplacer."
    -ForegroundColor Red
}
}

# =====
# ✅ Fin du script
# =====

Write-Host "`nCréation des utilisateurs, groupes et UO terminée." -ForegroundColor Magenta
Write-Host "Nombre d'utilisateurs traités : $($utilisateurs.Count)" -ForegroundColor Magenta
Write-Host "Groupes créés : $($groupes -join ', ')" -ForegroundColor Magenta
Write-Host "UO créées : $($groupesVersUO.Values -join ', ')" -ForegroundColor Magenta
```

```
Fichier CSV chargé avec succès. 36 utilisateurs trouvés.

Premier utilisateur :

Login      Prenom Nom      Initiale NomComplet      MotDePasse Groupe1 Groupe2
-----      -----      -----      -----
CDURANT1 Céline DURANT C.D      Céline DURANT Passe2!      SR

Groupes identifiés : COMMERCIAL, COMMU, COMPTA, DEV, SR
Groupe 'COMMERCIAL' créé avec succès.
Groupe 'COMMU' créé avec succès.
Groupe 'COMPTA' créé avec succès.
Groupe 'DEV' créé avec succès.
Groupe 'SR' créé avec succès.
Traitement de l'utilisateur : CDURANT1
Utilisateur 'CDURANT1' créé avec succès.
Utilisateur 'CDURANT1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : CSEBEC1
Utilisateur 'CSEBEC1' créé avec succès.
Utilisateur 'CSEBEC1' ajouté au groupe 'SR'.
Utilisateur 'CSEBEC1' ajouté au groupe 'DEV'.
Traitement de l'utilisateur : CTERALP1
Utilisateur 'CTERALP1' créé avec succès.
Utilisateur 'CTERALP1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : CTESERE1
Utilisateur 'CTESERE1' créé avec succès.
Utilisateur 'CTESERE1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : CTUJAGU1
Utilisateur 'CTUJAGU1' créé avec succès.
Utilisateur 'CTUJAGU1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : IDUTEC1
Utilisateur 'IDUTEC1' créé avec succès.
Utilisateur 'IDUTEC1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : IIRIBAR1
Utilisateur 'IIRIBAR1' créé avec succès.
Utilisateur 'IIRIBAR1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : IIRIEN1
Utilisateur 'IIRIEN1' créé avec succès.
Utilisateur 'IIRIEN1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : ISYBER1
Utilisateur 'ISYBER1' créé avec succès.
Utilisateur 'ISYBER1' ajouté au groupe 'DEV'.
Traitement de l'utilisateur : ITISSOT1
Utilisateur 'ITISSOT1' créé avec succès.
Utilisateur 'ITISSOT1' ajouté au groupe 'DEV'.
Traitement de l'utilisateur : ITOTON1
Utilisateur 'ITOTON1' créé avec succès.
Utilisateur 'ITOTON1' ajouté au groupe 'DEV'.
Utilisateur 'ITOTON1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : ITUGEL1
Utilisateur 'ITUGEL1' créé avec succès.
Utilisateur 'ITUGEL1' ajouté au groupe 'DEV'.
Utilisateur 'ITUGEL1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : JHULM1
Utilisateur 'JHULM1' créé avec succès.
Utilisateur 'JHULM1' ajouté au groupe 'DEV'.
Traitement de l'utilisateur : JICKX1
Utilisateur 'JICKX1' créé avec succès.
Utilisateur 'JICKX1' ajouté au groupe 'DEV'.
Utilisateur 'JICKX1' ajouté au groupe 'SR'.
Traitement de l'utilisateur : MDUPERT1
```

```
Traitements de l'utilisateur : ALALANN1
Utilisateur 'ALALANN1' créé avec succès.
Utilisateur 'ALALANN1' ajouté au groupe 'COMMERCIAL'.
Traitement de l'utilisateur : BVECCIA1
Utilisateur 'BVECCIA1' créé avec succès.
Utilisateur 'BVECCIA1' ajouté au groupe 'COMMERCIAL'.
Utilisateur 'BVECCIA1' ajouté au groupe 'COMMU'.
Traitement de l'utilisateur : CDARRIC1
Utilisateur 'CDARRIC1' créé avec succès.
Utilisateur 'CDARRIC1' ajouté au groupe 'COMMERCIAL'.
Traitement de l'utilisateur : CMOUMIQ1
Utilisateur 'CMOUMIQ1' créé avec succès.
Utilisateur 'CMOUMIQ1' ajouté au groupe 'COMPTA'.
Traitement de l'utilisateur : DDUMERC1
Utilisateur 'DDUMERC1' créé avec succès.
Utilisateur 'DDUMERC1' ajouté au groupe 'COMPTA'.
Traitement de l'utilisateur : FUHXY1
Utilisateur 'FUHXY1' créé avec succès.
Utilisateur 'FUHXY1' ajouté au groupe 'COMPTA'.
Traitement de l'utilisateur : GLAXALD1
Utilisateur 'GLAXALD1' créé avec succès.
Utilisateur 'GLAXALD1' ajouté au groupe 'COMPTA'.
Traitement de l'utilisateur : ITISSIONE1
Utilisateur 'ITISSIONE1' créé avec succès.
Utilisateur 'ITISSIONE1' ajouté au groupe 'COMPTA'.
Traitement de l'utilisateur : JMATHED01
Utilisateur 'JMATHED01' créé avec succès.
Utilisateur 'JMATHED01' ajouté au groupe 'COMPTA'.
Traitement de l'utilisateur : CDAURIA1
Utilisateur 'CDAURIA1' créé avec succès.
Utilisateur 'CDAURIA1' ajouté au groupe 'COMMU'.
Utilisateur 'CDAURIA1' ajouté au groupe 'COMMERCIAL'.
Traitement de l'utilisateur : CSARRAZ1
Utilisateur 'CSARRAZ1' créé avec succès.
Utilisateur 'CSARRAZ1' ajouté au groupe 'COMMU'.
Utilisateur 'CSARRAZ1' ajouté au groupe 'COMMERCIAL'.
Traitement de l'utilisateur : CTOSON1
Utilisateur 'CTOSON1' créé avec succès.
Utilisateur 'CTOSON1' ajouté au groupe 'COMMU'.
Traitement de l'utilisateur : IDARRAI1
Utilisateur 'IDARRAI1' créé avec succès.
Utilisateur 'IDARRAI1' ajouté au groupe 'COMMU'.
Traitement de l'utilisateur : ITAVENA1
Utilisateur 'ITAVENA1' créé avec succès.
Utilisateur 'ITAVENA1' ajouté au groupe 'COMMU'.
Groupe 'COMPTA' déplacé dans l'UO 'RH'.
Groupe 'COMMU' déplacé dans l'UO 'Redac'.
Groupe 'COMMERCIAL' déplacé dans l'UO 'Com'.
Groupe 'SR' déplacé dans l'UO 'RS'.
Groupe 'DEV' déplacé dans l'UO 'Dev'.

Création des utilisateurs, groupes et UO terminée.
Nombre d'utilisateurs traités : 36
Groupes créés : COMMERCIAL, COMMU, COMPTA, DEV, SR
UO créées : RH, Redac, Com, RS, Dev
```

Listes des Utilisateurs et des UO dans l'AD

Banuls Alexandre

Utilisateurs et ordinateurs Active	Nom	Type	Description
> Requêtes enregistrées			
gsb.local	admin	Utilisateur	
BuiltIn	Administrateur	Utilisateur	Compte d'utilisateur d'a...
Com	André LALANNE	Utilisateur	
Computers	Bruno VECCHIANI	Utilisateur	
Dev	Camille DAURIAUC	Utilisateur	
Domain Controllers	Catherine SEBEC	Utilisateur	
ForeignSecurityPrincipal	Christian DARRICAU	Utilisateur	
Managed Service Account	Christian MOUMIQUE	Utilisateur	
Redac	Christel TUJAGUE	Utilisateur	
RH	Claire TERALPA	Utilisateur	
RS	Claude SARRAZIN	Utilisateur	
Users	Céline DURANT	Utilisateur	
	Clementine TOSON	Utilisateur	
	Constant TESEREU	Utilisateur	
	Daniel DUMERCOQ	Utilisateur	
	DefaultAccount	Utilisateur	Compte utilisateur géré ...
	Francis UHXY	Utilisateur	
	Georges LAXALDE	Utilisateur	
	Iannech IRIBARE	Utilisateur	
	Ibrahim TAVENARD	Utilisateur	
	Ida TUGEL	Utilisateur	
	Ignace DARAILLANS	Utilisateur	
	Ignace TISSOT	Utilisateur	
	Igor DUTEC	Utilisateur	
	Invés TOTON	Utilisateur	
	Invité	Utilisateur	Compte d'utilisateur inv...
	Isabelle SYBER	Utilisateur	
	Isabelle TISSIER	Utilisateur	
	Isidore IRIEN	Utilisateur	
	Jack HULM	Utilisateur	
	Jean ICKX	Utilisateur	
	Jean Louis MATHEO	Utilisateur	
	Marc DUPERT	Utilisateur	
	Marcel ISARTE	Utilisateur	
	Martin TENIE	Utilisateur	
	Martine SOISSON	Utilisateur	
	Matthieu TERASON	Utilisateur	
	Patricia TENIE	Utilisateur	
	Patrick TIGNOUS	Utilisateur	
	Pierre DUDON	Utilisateur	

On voit bien que lorsque que j'ouvre l'UO RH , il ya bien son groupe ainsi que c'est utilisateur dans le groupe

Banuls Alexandre

The screenshot shows the Windows Active Directory Users and Computers management console. On the left, a tree view displays the organizational structure under 'Utilisateurs et ordinateurs Active'. A node 'gsb.local' is expanded, showing subfolders like 'Builtin', 'Com', 'Computers', 'Dev', 'Domain Controllers', 'ForeignSecurityPrincipal', 'Managed Service Account', 'Redac', 'RH', 'RS', and 'Users'. In the center, a table lists a group named 'COMPTA'. The table has columns 'Nom', 'Type', and 'Description'. The 'Nom' column shows 'COMPTA', 'Type' shows 'Groupe de séc...', and 'Description' is empty. A context menu is open over the 'COMPTA' row, with the 'Propriétés...' option selected. A detailed properties dialog box titled 'Propriétés de : COMPTA' is displayed. The 'Membres' tab is selected, showing a list of users: Christian MO..., Daniel DUME..., Francis UHXY, Georges LAX..., Isabelle TISSI..., and Jean Louis M.... Each entry includes a small user icon and the full name followed by the path 'gsb.local/Users'. At the bottom of the dialog are 'Ajouter...' and 'Supprimer' buttons, and standard 'OK', 'Annuler', and 'Appliquer' buttons.