

"Impossible de charger le profil utilisateur"

Puisque vous êtes administrateur local, vous pouvez utiliser l'Éditeur de stratégie de groupe locale (`gpedit.msc`) pour lancer ce script automatiquement au démarrage du système, avant la connexion de l'utilisateur.

1. Préparation du script :

- Copiez le code ci-dessous dans un fichier texte.
- Enregistrez le sous `C:\Scripts\Repair-UserProfiles.ps1`.
- Créez le dossier `C:\Scripts` si nécessaire.

2. Configuration de la tâche (via `gpedit.msc`) :

- Ouvrez `gpedit.msc` (exécutable en tant qu'administrateur).
- Allez dans : Configuration ordinateur > Modèles d'administration > Système > Connexion.
- Cherchez la stratégie : "Exécuter le script de connexion au démarrage de l'ordinateur" (ou "Run logon script" selon la version de Windows).
- Activez la stratégie et indiquez le chemin du script (ex: `C:\Scripts\Repair-UserProfiles.ps1`).
- *Note* : Pour que PowerShell s'exécute, assurez-vous que la politique d'exécution des scripts sur la machine permet l'exécution locale (généralement `RemoteSigned`). Si nécessaire, ajoutez `powershell.exe -ExecutionPolicy Bypass -File "C:\Scripts\Repair-UserProfiles.ps1"` dans les paramètres de la stratégie.

Le code en question :

```
# Script de réparation automatique des profils utilisateur corrompus
# Exécuté avec les droits Administrateur local au démarrage

$regPath = "HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\ProfileList"
$profiles = Get-ChildItem -Path $regPath

foreach ($profile in $profiles) {
    $sid = $profile.Name.Split('\')[2]
    $profileKeys = Get-ItemProperty -Path $profile.PSPath

    # Vérification des conditions d'erreur
    # 1. Présence de la valeur "ProfileImagePath"
    # 2. Présence de la valeur "State" (si 1, profil corrompu)
```

```

# 3. Présence d'une entrée .bak (profil temporaire ou en attente)

if ($profileKeys.ProfileImagePath -and ($profileKeys.State -eq 1 -or
$profileKeys.State -eq 2)) {
    Write-Host "Profil détecté comme corrompu : $sid"

    # Cas 1 : Le profil a une extension .bak (le vrai profil est là, mais pointé
vers .bak)
    if (Test-Path "$regPath\$sid.bak") {
        $bakProfile = Get-ItemProperty -Path "$regPath\$sid.bak"

        # On copie les données du .bak vers le SID principal
        Copy-Item "$regPath\$sid.bak" -Destination "$regPath\$sid" -Force

        # On supprime la valeur State du profil principal pour le rendre valide
        Remove-ItemProperty -Path "$regPath\$sid" -Name "State" -
ErrorAction SilentlyContinue
        Remove-ItemProperty -Path "$regPath\$sid" -Name "RefCount" -
ErrorAction SilentlyContinue

        # On supprime l'entrée .bak (elle n'est plus nécessaire)
        Remove-Item -Path "$regPath\$sid.bak" -Recurse -Force

        Write-Host "Réparation effectuée : Migration de $sid.bak vers $sid."
    }
    # Cas 2 : Le profil est marqué corrompu mais n'a pas de .bak (cas plus
rare)
    elseif ($profileKeys.State -eq 1) {
        # On tente de "nettoyer" les drapeaux d'état sans supprimer le profil
        Remove-ItemProperty -Path "$regPath\$sid" -Name "State" -
ErrorAction SilentlyContinue
        Remove-ItemProperty -Path "$regPath\$sid" -Name "RefCount" -
ErrorAction SilentlyContinue
        Write-Host "Réparation effectuée : Nettoyage des drapeaux d'état
pour $sid."
    }
}
}

Write-Host "Vérification terminée."

```

Revision #2

Created 2026-03-24 09:03:27 UTC by Steven

Updated 2026-03-24 09:36:58 UTC by Steven