

17/03/2025

Powershell

Script mappage lecteurs réseaux



Dorian Montoir
ST SAUVEUR

SOMMAIRE

Table des matières

Script Mappage Lecteurs Réseaux.....	2
Explications du fonctionnement du script.....	6
1. Initialisation des informations de mappage des lecteurs réseau.....	6
2. Fonction Map-NetworkDrive	6
3. Fonction Get-UserGroups	6
4. Récupération du nom de l'utilisateur courant	7
5. Récupération des groupes de l'utilisateur courant	7
6. Mappage des lecteurs réseau en fonction des groupes de l'utilisateur	7
7. Structure de données.....	7
8. Comportement du script.....	7

Script Mappage Lecteurs Réseaux

```
# Liste des utilisateurs/groupes et des lecteurs à mapper
$networkDrives = @(
    @{
        GroupNames = @("directioncompta", "secretairecompta")
        DriveLetter = "W:"
        NetworkPath = "\\SRV-ADDS\secretariat"
    },
    @{
        GroupNames = "directioncompta"
        DriveLetter = "V:"
        NetworkPath = "\\SRV-ADDS\direction"
    },
    @{
        GroupNames = @("directioncompta", "secretairecompta")
        DriveLetter = "Y:"
        NetworkPath = "\\SRV-ADDS\communcompta"
    },
    @{
        GroupNames = @("directioncompta", "etude", "atelier")
        DriveLetter = "S:"
        NetworkPath = "\\SRV-ADDS\archives"
    },
    @{
        GroupNames = @("directioncompta", "etude", "atelier")
        DriveLetter = "T:"
        NetworkPath = "\\SRV-ADDS\communatelier"
    },
    @{
        GroupNames = @("directioncompta", "etude", "atelier", "secretairecompta", "visiteur")
        DriveLetter = "Z:"
    }
)
```

```

    NetworkPath = "\\SRV-ADDS\public"
},
@{
    UserName = "DOE"
    DriveLetter = "U:"
    NetworkPath = "\\SRV-ADDS\JohnDoe"
},
@{
    UserName = "DUPONT"
    DriveLetter = "U:"
    NetworkPath = "\\SRV-ADDS\AntoineDupont"
},
@{
    UserName = "POULAIN"
    DriveLetter = "U:"
    NetworkPath = "\\SRV-ADDS\MelaniePoulain"
},
@{
    UserName = "NOAH"
    DriveLetter = "U:"
    NetworkPath = "\\SRV-ADDS\YannickNoah"
}
)

# Fonction pour vérifier et mapper les lecteurs réseau avec net use (persistant)
function Map-NetworkDrive {
    param(
        [string]$DriveLetter,
        [string]$NetworkPath
    )

    # Vérifier si le lecteur est déjà mappé

```

```

$drive = Get-WmiObject -Class Win32_NetworkConnection | Where-Object { $_.LocalName -eq $DriveLetter
}

if ($drive) {
    Write-Host "$DriveLetter déjà mappé."
} else {
    # Mapper le lecteur réseau de façon persistante avec net use
    net use $DriveLetter $NetworkPath /persistent:yes
    Write-Host "$DriveLetter mappé avec succès vers $NetworkPath."
}
}

# Fonction pour obtenir les groupes de l'utilisateur via "net user /domain" et analyser la sortie
function Get-UserGroups {
    param([string]$UserName)

    # Exécuter la commande "net user" pour récupérer les groupes de l'utilisateur (avec /domain)
    $userGroups = net user $UserName /domain | Select-String -Pattern "Appartient aux groupes |^\s*\*.*"

    # Initialiser un tableau pour stocker les groupes extraits
    $groups = @()

    # Si la commande retourne des groupes, les extraire
    foreach ($line in $userGroups) {
        # Si la ligne contient des groupes (de type globaux ou locaux)
        if ($line -match "Appartient aux groupes (locaux|globaux)\s+(.+)") {
            $groupNames = $matches[2].Trim().Split(",") | ForEach-Object { $_.Trim().TrimStart('*') }
            $groups += $groupNames
        }

        # Si la ligne contient des groupes sur une nouvelle ligne (en raison des astérisques)
        elseif ($line -match "^\s*\* (.+)") {
            $groupNames = $matches[1].Trim()
            $groups += $groupNames
        }
    }
}

```

```

    }
}

# Retourner la liste des groupes extraits
return $groups
}

# Récupérer le nom de l'utilisateur actuel
$user = [System.Security.Principal.WindowsIdentity]::GetCurrent().Name.Split("\")[1]

# Récupérer les groupes de l'utilisateur actuel via "net user /domain"
$userGroups = Get-UserGroups -UserName $user

# Afficher les groupes trouvés pour débogage
Write-Host "Groupes de l'utilisateur $user: $($userGroups -join ', ')"

# Parcourir les configurations de mappage de lecteurs
foreach ($driveConfig in $networkDrives) {
    if ($driveConfig.GroupNames) {
        # Vérifier si l'utilisateur fait partie de l'un des groupes spécifiés
        foreach ($groupName in $driveConfig.GroupNames) {
            if ($userGroups -contains $groupName) {
                Map-NetworkDrive -DriveLetter $driveConfig.DriveLetter -NetworkPath $driveConfig.NetworkPath
                break # Si un groupe correspond, on mappe le lecteur et on sort de la boucle
            }
        }
    }
    elseif ($driveConfig.UserName -eq $user) {
        # Vérifier si l'utilisateur correspond
        Map-NetworkDrive -DriveLetter $driveConfig.DriveLetter -NetworkPath $driveConfig.NetworkPath
    }
}
}

```

Explications du fonctionnement du script

Ce script PowerShell est conçu pour mapper des lecteurs réseau en fonction des groupes d'utilisateurs ou de l'utilisateur lui-même. Il analyse les groupes d'un utilisateur, puis, en fonction des groupes auxquels il appartient ou de son nom d'utilisateur, il mappe les lecteurs réseau appropriés. Voici une analyse détaillée des différentes sections et de leur fonctionnement.

1. Initialisation des informations de mappage des lecteurs réseau

Le script commence par définir un tableau `$networkDrives` qui contient des configurations pour les lecteurs réseau à mapper. Chaque élément de ce tableau est un dictionnaire (ou hash table) avec des clés qui peuvent être `GroupNames` (groupes d'utilisateurs auxquels un lecteur doit être mappé), `DriveLetter` (lettre du lecteur), et `NetworkPath` (chemin du lecteur réseau).

Voici quelques exemples d'éléments de ce tableau :

- Un lecteur `W` : est mappé pour les groupes `directioncompta` et `secretairecompta` vers `\\SRV-ADDS\secretariat`.
- Un lecteur `U` : est mappé spécifiquement pour l'utilisateur `DOE` vers `\\SRV-ADDS\JohnDoe`.

2. Fonction `Map-NetworkDrive`

La fonction `Map-NetworkDrive` est responsable du mappage effectif des lecteurs réseau. Elle prend deux paramètres :

- **`$DriveLetter`** : la lettre du lecteur à mapper (par exemple, `W` :)
- **`$NetworkPath`** : le chemin du lecteur réseau (par exemple, `\\SRV-ADDS\secretariat`)

Voici ce que la fonction fait :

- **Vérification du mappage existant** : Utilise `Get-WmiObject -Class Win32_NetworkConnection` pour vérifier si le lecteur réseau est déjà mappé.
- **Mappage du lecteur** : Si le lecteur n'est pas déjà mappé, elle exécute la commande `net use $DriveLetter $NetworkPath /persistent:yes` pour mapper le lecteur réseau de manière persistante.
- **Message de confirmation** : Affiche un message indiquant si le lecteur a été mappé avec succès ou s'il était déjà mappé.

3. Fonction `Get-UserGroups`

Cette fonction est utilisée pour récupérer la liste des groupes auxquels appartient un utilisateur spécifique en exécutant la commande `net user $UserName /domain`. Elle analyse ensuite la sortie de cette commande pour extraire les groupes auxquels l'utilisateur appartient.

La commande `net user /domain` retourne des informations détaillées sur l'utilisateur, y compris les groupes dont il fait partie. La fonction cherche spécifiquement des lignes qui contiennent des informations sur les groupes, en utilisant les expressions régulières suivantes :

- "**Appartient aux groupes (locaux|globaux)**" : Pour extraire les groupes locaux et globaux.
- "**^\\s***" : Pour capturer les groupes supplémentaires listés après l'astérisque.

4. Récupération du nom de l'utilisateur courant

Le script récupère le nom de l'utilisateur actuellement connecté en utilisant

`[System.Security.Principal.WindowsIdentity]::GetCurrent().Name`. Il extrait la partie du nom d'utilisateur (en excluant le domaine) en utilisant la méthode `Split("\")`.

5. Récupération des groupes de l'utilisateur courant

Le nom de l'utilisateur récupéré est passé à la fonction `Get-UserGroups`, qui retourne la liste des groupes auxquels l'utilisateur appartient. Cette liste est ensuite affichée pour débogage via `Write-Host`.

6. Mappage des lecteurs réseau en fonction des groupes de l'utilisateur

Le script parcourt ensuite chaque entrée du tableau `$networkDrives`. Pour chaque configuration de lecteur réseau :

- Si **GroupNames** est défini (c'est-à-dire que le mappage du lecteur est basé sur des groupes d'utilisateurs), il vérifie si l'utilisateur appartient à l'un des groupes de cette entrée.
 - Si l'utilisateur appartient à un des groupes, la fonction `Map-NetworkDrive` est appelée pour mapper le lecteur réseau.
- Si **UserName** est spécifié (c'est-à-dire que le mappage est basé sur le nom de l'utilisateur), il vérifie si l'utilisateur correspond à celui de l'entrée et, le cas échéant, mappe le lecteur réseau.

7. Structure de données

L'utilisation de la structure de données sous forme de tableau d'objets avec des hash tables permet de bien organiser les mappages des lecteurs et les critères associés (groupes ou utilisateur). Cela permet une gestion flexible et évolutive des mappages, où l'on peut ajouter, modifier ou supprimer facilement des lecteurs et des groupes d'utilisateurs associés.

8. Comportement du script

1. Le script commence par récupérer l'utilisateur et ses groupes.
2. Ensuite, il parcourt les configurations de lecteurs définies dans `$networkDrives` pour déterminer quels lecteurs doivent être mappés en fonction des groupes d'utilisateurs ou de l'utilisateur individuel.
3. Si un lecteur doit être mappé, la fonction `Map-NetworkDrive` est appelée pour effectuer le mappage réseau.