

Rapport de Projet

CRÉATION ET CONFIGURATION DE GPO VIA POWERSHELL

Sébastien Brun
CEFF | RUE BAPTISTE-SAVOYE 26

RAPPORT DE PROJET

CREATION ET CONFIGURATION DE GPO VIA POWERSHELL | SÉBASTIEN BRUN

SOMMAIRE

1.	INTRODUCTION	2
1.1.	PRES-REQUIS	2
2.	SCRIPT GPO WALLPAPER	2
3.	RAPPORT GPO	3
4.	BACKUP GPO	4
5.	RSOP.MSC	5
6.	CONCLUSION	5
7.	SOURCE	5

1. INTRODUCTION

- Les GPO (Group Policy ou stratégie de groupe) sont des outils puissants permettant de configurer les paramètres de sécurité, les options des ordinateurs de votre domaine et les utilisateurs dans un environnement Active Directory. Dans cette documentation, nous allons voir comment créer un GPO et le configurer, ensuite nous allons voir comment créer un rapport de nos GPO afin de pouvoir vérifier leur efficacité. Et pour finir, nous allons voir comment créer un backup de nos GPO.

1.1. PRES-REQUIS

- PowerShell 5.1.
- GitHub: <https://github.com/Sebibu0u/PowerShellProjet>

2. SCRIPT GPO WALLPAPER

- Pour commencer nous allons demander si l'utilisateur du script désire créer notre GPO.
- Pour ceci nous allons créer une variable nommée « reponse1 ». Dans cette variable nous allons insérer les cmdlets « Read-Host -Prompt "Texte" » qui permettent de récupérer la réponse de la question « Texte ».

- ```
$reponse1 = Read-Host -Prompt "Voulez-vous attribuer un fond d'écran Lié à une OU spécifique ? (Oui/Non)"
```

- Ensuite nous allons faire une boucle grâce à la cmdlet « if () » afin de pouvoir continuer le script sans créer la GPO en cas de « Non ». If demande des paramètres à entrer dans ça parenthèse. Nous devons vérifier si L'opérateur de comparaison à utiliser est « -like ».

- ```
If ($reponse1 -like 'Oui')  
{ }
```

- Dans les crochets de la boucle, nous commençons par créer une variable afin de stocker le nom de notre GPO.

- ```
$GPOname = 'Wallpaper'
```

- Ensuite, il nous faut utiliser la commande « Set-GpRegistryValue » qui permet de **configurer** (ajouter, modifier ou supprimer) des valeurs de registre pour une stratégie de groupe spécifique. À ne pas confondre avec la cmdlet « Get-GPPrefRegistryValue » qui permet de **récupérer** les valeurs de registre, et non de les configurer. La cmdlet « Set-GpRegistryValue » demande le nom de la GPO à configurer, la clé de registre du paramètre à modifier, le nom de la valeur à modifier, le type et pour finir la valeur à utiliser.
- Pour notre fond d'écran nous allons utiliser notre variable « \$GPOname ». La clé qui correspond à notre fond d'écran est « HKEY\_CURRENT\_USER\Control Panel\Desktop ». Le nom de la valeur à modifier est logiquement « Wallpaper ». Le type à utiliser est « String » et pour finir la valeur correspond au chemin de notre image (le chemin doit être identique sur les machines qui utiliserons notre GPO).

- ```
Set-GpRegistryValue -Name $GPOname -key "HKEY_CURRENT_USER\Control Panel\Desktop" -  
ValueName "Wallpaper" -type string -value "C:\Wallpaper\283763.jpg"
```

- Pour finir la configuration de notre GPO, il nous faut la **lié** à l'OU (organisation unit : groupe d'utilisateurs) à laquelle on veut appliquer la GPO. Ce processus permet d'appliquer des GPO uniquement à

certain utilisateur membre de cette unité d'organisation. Pour cela nous allons utiliser la cmdlt « New-GPLink » à ne pas confondre avec la cmdlt « Set-GPLink » qui permet de **modifier** un lien et non de le créer. « New-GPLink » demande le nom de la GPO (\$GPOName), le Target qui correspond à votre OU (ou=general,dc=seb,dc=com), puis nous allons activer le lien grâce à « -LinkEnabled Yes ».

- `New-GPLink -Name $GPOName -Target "ou=general,dc=seb,dc=com" -LinkEnabled Yes`

- Pour finir nous devons utiliser la cmdlt « Gpupdate /force » afin de forcer l'application de la GPO.

- `Gpupdate /force`

```
#Crée une GPO qui permet de changer de fond d'écran.
$reponse1 = Read-Host -Prompt "Voulez-vous attribuer un font d'écran Lié à une OU spécifique ? (Oui/Non)"
if ($reponse1 -like 'Oui')
{
    $GPOName = 'Wallpaper'
    #Créer la GPO.
    New-GPO -Name $GPOName
    #Configurer la GPO.
    Set-GpRegistryValue -Name $GPOName -key "HKEY_CURRENT_USER\Control Panel\Desktop" -ValueName "Wallpaper" -type string -value "C:\Wallpaper\283763.jpg"
    #Permet de linker vers l'OU, à changer le lien "Target"
    New-GPLink -Name $GPOName -Target "ou=general,dc=seb,dc=com" -LinkEnabled Yes
    Gpupdate /force
}
```

3. RAPPORT GPO

- Pour commencer, nous allons faire en sorte de demander si l'utilisateur désire créer un rapport des différentes GPO. Pour cela, nous allons créer une variable nommé « reponse2 » et dans cette variable, nous allons insérer les cmdlt « Read-Host -Prompt "Texte" ».

- `$rapport = Read-Host -Prompt "Voulez-vous créer un rapport ? (Oui/Non)"`

- Nous allons créer une boucle avec la cmdlt « if () » afin de pouvoir continuer le script dans le cas d'un « Non ». If demande des paramètres à entrer dans ça parenthèse. Nous devons vérifier si L'opérateur de comparaison à utiliser est « -like ».

- `If ($reponse2 -like 'Oui')
{ }`

- Dans les crochets de la boucle, nous commençons par demander le nom de la GPO à laquelle nous désirons créer un rapport et nous allons stocker ce nom dans la variable « nom ». Pour cela, nous allons réutiliser les cmdlt « Read-Host -Prompt "Texte" ».

- `$nom = Read-Host -Prompt "Inserez le nom de votre GPO "`

- Ensuite nous allons demander sous quelle format, le rapport est-il désiré. La réponse devra être stocker dans une variable nommé « Format ».

- `$Format = Read-Host -Prompt "Inserez le format (Html/XML) "`

- Pour pouvoir donner le choix du format à l'utilisateur, nous devons créer une boucle qui prend en compte les deux formats. Commençons avec le format « Html » en sachant que l'opérateur de comparaison à utiliser est « -like ».

- `If ($Format -like 'Html')`
`{`
- Entre les crochets, nous allons utiliser la cmdlet « New Item + chemin -itemType Directory » afin de créer un dossier qui servira à stocker notre rapport.
 - `New-Item "C:\Report" -itemType Directory`
- Ensuite nous allons utiliser la cmdlet « Get-GPOReport » afin de récupérer notre rapport. Get-GPOReport demande plusieurs paramètres comme le nom de votre GPO (\$nom), le type du rapport (\$Format), puis le lieu où nous voulons déposer le rapport (c:\report\Report.html).
 - `Get-GPOReport -Name $nom -ReportType $Format -Path c:\Report\Report.html`
- Maintenant que la boucle « if () » est terminée, nous allons passer à la boucle « else » qui va nous servir à proposer le deuxième format.
 - `Else`
`{`
- Insérez exactement le code vu pour le format html. Le seul changement à faire est sur le nom du rapport (Report.xml).
 - `New-Item "C:\Report" -itemType Directory`
 - `Get-GPOReport -Name $nom -ReportType $Format -Path c:\Report\Report.xml`

```
$rapport = Read-Host -Prompt "Voulez-vous créer un rapport ? (Oui/Non)"
if ($rapport -like 'Oui')
{
    $nom = Read-Host -Prompt "Insérez le nom de votre GPO "
    $Format = Read-Host -Prompt "Insérez le format (Html/Xml) "
    if($Format -like 'Html')
    {
        New-Item "C:\Report" -itemType Directory
        Get-GPOReport -Name $nom -ReportType $Format -Path c:\Report\Report.html
    }
    else
    {
        New-Item "C:\Report" -itemType Directory
        Get-GPOReport -Name $nom -ReportType $Format -Path c:\Report\Report.xml
    }
}
```

4. BACKUP GPO

- Faire en sorte de demander si la Backup est nécessaire et stocker dans la variable « reponse4 »
 - `$reponse4 = Read-Host -Prompt "Voulez-vous faire une backup de vos GPO ? (Oui/Non)"`
- Faire une boucle afin de pouvoir terminer le script sans faire de backup dans le cas d'un « Non ».
 - `If ($reponse4 -like 'Oui')`
`{`

- Entre les crochets, créez un dossier afin d'accueillir la backup. Puis récupéré et copier les données de toutes les GPO avec la cmdlt « Get-GPO -ALL | Backup-GPO -Path C:\GPOBackup\ -Comment "\$(Get-Date)" »
 - `New-Item "C:\GPOBackup" -itemType Directory`
 - `Get-GPO -All | Backup-GPO -Path C:\GPOBackup\ -Comment "$(Get-Date)"`

```
$reponse4 = Read-Host -Prompt "Voulez-vous faire une backup de vos GPO ? (Oui/Non)"

if ($reponse4 -like 'Oui')
{
    New-Item "C:\GPOBackup" -itemType Directory
    Get-GPO -All | Backup-GPO -Path C:\GPOBackup\ -Comment "$(Get-Date)"
}
```

5. RSOP.MSC

- « rsop.msc » ou « Resultant Set of Policy » permet de voir les paramètres de stratégie de groupe qui sont appliqués sur un ordinateur. Pratique pour se renseigner sur la configuration ou pour vérifier si les GPO sont bien appliquées.

6. CONCLUSION

- Pour conclure cette documentation et ce projet, j'ai apprécié effectuer des recherches sur ce sujet qui est un sujet très pratique et utilisé au quotidien en entreprise. Malgré les difficultés, j'ai pu trouver certaines informations qui m'ont permis de réaliser cette documentation technique. Les GPO via PowerShell sont intéressantes dans le sens où elles nous font gagner un temps appréciable dans le cas où on arrive à les mettre en place.

7. SOURCE

- Aide Microsoft.
- Manage group policies with PowerShell.
<https://cloudbrothers.info/en/manage-group-policies-powershell/>