

## GUIDE ETUDIANT

L'objectif de ce workshop est de créer un script que l'on exécutera grâce aux tâches planifiées et qui aura pour but d'écrire dans le fichier *C:\result.log* les noms des fichiers de plus de 5 Mo du répertoire *C:\Users\Public* (et ses sous-répertoires). Si le script n'a pas été lancé depuis plus de 5 jours, un message l'indiquera au début du fichier, la date de dernière exécution étant enregistrée dans la base de registres avec le nombre de fichiers trouvés. Dans le journal des événements Application, on pourra suivre la date et l'heure de début et de fin d'exécution du script. A chaque nouvelle exécution, le fichier *result.log* existant déjà est conservé sous le nom *result.log.bak*.

## GUIDE ETUDIANT

### ÉTAPE 1

Rédiger un script permettant d'écrire la chaîne de caractères « toto » dans le fichier *C:\result.log*. Utiliser les variables *rep* pour stocker la chaîne de caractères et *log* pour le nom du fichier. Plus tard, la variable *rep* stockera le nom du répertoire que l'on souhaite parcourir.

Indices :

- Un nom de variable commence par \$ ;
- La commande pour écrire dans un fichier est Out-File ;
- On utilise la barre verticale pour rediriger un élément en entrée d'une commandlet.

Code script:

## GUIDE ETUDIANT

### ÉTAPE 2

Ajouter une condition au script précédent: si le fichier *result.log* existe déjà, le recopier en tant que *result.log.bak*.

Indices :

- On utilise Test-Path pour vérifier l'existence d'un répertoire ;
- L'opérateur + permet de concaténer des chaînes de caractères ;
- La commande Copy-Item permet de recopier un fichier.

Code script:

## GUIDE ETUDIANT

### ÉTAPE 3

Écrire la date du jour au début du fichier *result.log* et parcourir le répertoire *C:\Users\Public* (dans la variable *rep*). La chaîne « nom\_fichier trouvé. » apparaîtra dans ce fichier autant de fois qu'il y a de fichiers ou répertoires dans *C:\Users\Public*. (ex : *C:\Users\Public\test.mp4* trouvé.)

#### Indices :

- On utilise *Get-Date* pour récupérer la date courante ;
- La commande *Add-Content* permet d'ajouter du contenu à un fichier ;
- L'option *recurse* de la commande *Get-ChildItem* permet d'accéder aux sous-dossiers ;
- La boucle *foreach* permet de parcourir une collection d'éléments.

#### Code script:

## GUIDE ETUDIANT

### ÉTAPE 4

Le script ne doit enregistrer dans le fichier de log que les fichiers de plus de 5 Mo sous cette forme :

Chemin complet du fichier : taille en Mo (ex : C:\Users\Public\test.mp4 : 512Mo) .

La limite de taille fixée sera enregistrée dans une variable nommé *taille\_max*. Un compteur de fichiers trouvés (variable nommée *cpt*) est mis en place pour afficher le nombre total de fichiers trouvés à la fin du fichier de log.

Indices :

- La propriété `fullname` sur un `Item` permet de récupérer le chemin complet et `length` pour récupérer la taille d'un fichier ;
- On utilise l'opérateur `-ge` (supérieur ou égal) pour faire une comparaison ;

Code script:

## GUIDE ETUDIANT

### ÉTAPE 5

L'étape 6 peut être accomplie même si cette étape n'est pas réussie.

On souhaite ajouter les enregistrements dans le journal d'évènements Application : un événement de type information au début de l'exécution du script et de même en fin d'exécution avec le nombre de fichiers trouvés. La source d'évènements à créer sera nommée *MonScript*.

Indices :

- La commande New-EventLog permet de créer un nouveau journal d'évènements et Get-WinEvent pour tester son existence ;
- L'opérateur contains permet de vérifier si un élément fait partie d'une collection ;
- On utilise la commande Write-EventLog pour écrire dans un journal.

Code script:

## GUIDE ETUDIANT

### ÉTAPE 6

On ajoute l'écriture dans la base de registre de la date de dernier lancement du script et du nombre de fichiers trouvés à cet endroit : HKEY\_LOCAL\_MACHINE\Software\MonScript. Le script ajoute une ligne dans le fichier de log indiquant la date de dernier lancement si cela fait plus de 5 jours qu'il n'a pas été lancé et un enregistrement de type avertissement (warning) est enregistré dans le journal d'évènements.

Indices :

- La base de registre est traitée comme un fichier ;
- La commande Get-ItemProperty permet de récupérer la valeur associée à une clé et on utilise Set-ItemProperty pour modifier une valeur ;
- La méthode statique Parse de DateTime permet de convertir une chaîne de caractères en un objet de type DateTime ;
- Pour indiquer le type d'enregistrement dans le journal d'évènements, on utilise l'option entrytype.

Code script final: