# Projet Powershell

## Présentation

Vous êtes sollicité pour réaliser un script permettant de gérer les fichiers partagés sur le serveur Windows de l’entreprise.

Ce script PowerShell doit permettre à un administrateur de choisir les fichiers à supprimer de manière récursive dans une partie de l’arborescence du serveur, en fonction de leur type (zip, exe, pdf…) et/ou d’une date d’antériorité.

Echange avec l’administrateur :

« Nous avons un dossier partagé dans lequel les utilisateurs (un dossier par personne) déposent les fichiers à sauvegarder. Je me suis rendu compte que les utilisateurs gardaient des fichiers qu’il n’est pas utile de sauvegarder, par exemple, des fichiers .exe, .bak, .zip, etc. et aussi des fichiers très anciens.

J’aurais donc besoin d’un petit script qui me permette, à partir d’un dossier et de ses sous-dossiers, de supprimer tous les fichiers ayant une certaine extension et/ou non modifiés depuis une certaine date.

Mais attention ! j’aimerais avoir une sorte de mode « simulation » qui me permette d’abord de voir les fichiers qui seront supprimés AVANT qu’ils ne le soient vraiment ! »

## Le script pas-à-pas : étape 1 réalisation du jeu d’essai

Pour réaliser et valider le script demandé, nous devez créer une arborescence de test.

Sur votre ordinateur, copiez-collez le script dans le PowerShell ISE, enregistrez-le dans un dossier de votre choix, et exécutez-le afin de :

* créer une arborescence à la racine de c:\
* créer des fichiers avec 2 types d’extension (.bak et .txt) et avec 2 dates de dernière modification (date du jour et 25/12/2010).

Contenu du script

|  |
| --- |
| Set-Location c:\  new-item -ItemType Directory -Name "sauvegarde" -Force  Set-Location .\sauvegarde  new-item -ItemType Directory -Name "sbaume" -Force  Set-Location .\sbaume  New-Item nomFichier1.txt -ItemType file -Value "Fichier 1" -Force  $(Get-Item nomFichier1.txt).lastwritetime=$(Get-Date "25/12/2010 01:00")  New-Item nomFichier2.bak -ItemType file -Value "Fichier 2" -Force  new-item -ItemType Directory -Name "travail" -Force  Set-Location .\travail  New-Item nomFichier3.txt -ItemType file -Value "Fichier 3" -Force  Set-Location c:\sauvegarde  new-item -ItemType Directory -Name "jgalopin" -Force  Set-Location .\jgalopin  New-Item nomFichier1.bak -ItemType file -Value "Fichier 1" -Force  $(Get-Item nomFichier1.bak).lastwritetime=$(Get-Date "25/12/2010 01:00")  new-item -ItemType Directory -Name "travail" -Force  Set-Location .\travail  New-Item nomFichier2.txt -ItemType file -Value "Fichier 2" -Force  New-Item nomFichier3.txt -ItemType file -Value "Fichier 3" -Force  Set-Location c:\sauvegarde |

### Questions sur les commandes PowerShell – 1

Quelle commande PowerShell permet de lister tous les éléments contenus dans un dossier ?

Quel paramètre de cette commande permet de faire une recherche récursive (dans les sous-dossiers) ?

Quel paramètre de cette commande permet de ne retenir que les fichiers ?

Quel paramètre de cette commande permet de filtrer sur les extensions ?

## Le script pas-à-pas : étape 2 Lister tous les fichiers récursivement selon une extension choisie

Lancez PowerShell ISE, saisissez le script suivant et enregistrez le script avec une extension .ps1.

|  |
| --- |
| $extension = "\*.txt"  Get-ChildItem -Recurse -File -Include $extension |

Positionnez-vous dans le dossier sauvegarde créé précédemment puis en tapez sur la touche F5 pour exécuter le script.

Le script déclare une variable $extension dans laquelle on met une chaîne de caractères qui représente l’extension recherchée.

La commande Get-ChildItem qui permet d’obtenir tous les objets situés dans un dossier est utilisée. On lui indique 3 paramètres :

* Recurse : pour descendre dans les sous-dossiers
* File : pour ne prendre que les fichiers (donc on exclut les dossiers)
* Include : pour limiter à une extension

### Question sur les commandes PowerShell - 2

Quelle commande PowerShell permet de filtrer les résultats renvoyés par Get-Childitem ?

## Le script pas-à-pas : étape 3 Lister tous les fichiers récursivement selon une extension choisie et une date :

Complètez le script précédent avec une nouvelle variable (attention au format de date : MM/JJ/AAAA). On utilise le mécanisme du « pipeline » (barre verticale |) pour envoyer le résultat du Get-ChildItem à la commande Where. Celle-ci ne retiendra que les fichiers dont la date de dernière modification (LastWriteTime) est inférieure (-lt) à la variable. On rend l’affichage plus lisible en ne mettant que le nom et la date (commande Select).

|  |
| --- |
| $extension = "\*.txt"  $datemax = "12/31/2015"  Get-ChildItem -Recurse -File -Include $extension | Where {$\_.LastWriteTime -lt $datemax} | Select fullname, lastwritetime |

A ce stade, vous pouvez essayer d’autres extensions et d’autres dates pour valider le fonctionnement du script.

### Questions sur les commandes PowerShell - 3

Quelle commande PowerShell permet de supprimer les résultats renvoyés par Get-Childitem ?

Le script pas-à-pas : étape 4 Suppression et mode simulation

Ajoutez une variable **simule** qui si elle est «false» (par défaut elle sera $true) permettra vraiment la suppression avec la commande (Remove-Item).

|  |
| --- |
| $extension = "\*.bak"  $datemax = "12/31/2025"  $simule = $false  if($simule -eq $true)  {  Write-Host "Les fichiers suivants seront supprimés :"  Get-ChildItem -Recurse -File -Include $extension | Where {$\_.LastWriteTime -lt $datemax} | Select fullname, lastwritetime  }  else  {  Get-ChildItem -Recurse -File -Include $extension | Where {$\_.LastWriteTime -lt $datemax} | Remove-Item -  Verbose  Write-Host "Les fichiers ont été supprimés !"  } |

Le script pas-à-pas : étape 5 Dernière étape : les paramètres

Le script doit être appelé avec des paramètres, donc on transforme les variables. On gère les cas où les paramètres ne sont pas renseignés, donc on met des valeurs par défaut :

* Extension : valeur par défaut \* qui veut dire tous les fichiers
* Datemax : valeur par défaut date infinie
* Simule : valeur par défaut à vrai.

|  |
| --- |
| param(  [string]$extension="\*",  [datetime]$datemax="12/31/9999",  [boolean]$simule=$true  )  if($simule -eq $true)  {  Write-Host "Les fichiers suivants seront supprimés :"  Get-ChildItem -Recurse -File -Include $extension | Where {$\_.LastWriteTime -lt $datemax} | Select fullname, lastwritetime  }  else  {  Get-ChildItem -Recurse -File -Include $extension | Where {$\_.LastWriteTime -lt $datemax} | Remove-Item -Verbose  Write-Host "Les fichiers ont été supprimés !"  } |

## Tests

Le script peut être appelé avec différents paramètres :

|  |
| --- |
| PS C:\> Set-Location c:\sauvegarde  PS C:\sauvegarde> .’C:\Users\supprFichiers.ps1’ -extension "\*.bak" -datemax "12/31/2020" -simule 1  Les fichiers suivants seront supprimés :  FullName LastWriteTime  -------- -------------  C:\sauvegarde\jgalopin\nomFichier1.bak 25/12/2010 01:00:00  C:\sauvegarde\sbaume\nomFichier2.bak 05/12/2020 09:00:57  PS C:\sauvegarde> .’C:\Users\supprFichiers.ps1’ -extension "\*.bak" -datemax "12/31/2020" -simule 0  COMMENTAIRES : Opération « Supprimer le fichier » en cours sur la cible « C:\sauvegarde\jgalopin\nomFichier1.bak ».  COMMENTAIRES : Opération « Supprimer le fichier » en cours sur la cible « C:\sauvegarde\sbaume\nomFichier2.bak ».  Les fichiers ont été supprimés !  PS C:\sauvegarde> |

### Réponses aux questions sur les commandes PowerShell – 1

Quelle commande PowerShell permet de lister tous les éléments contenus dans un dossier ?

get-childitem

Quel paramètre de cette commande permet de faire une recherche récursive (dans les sous-dossiers) ?

recurse

Quel paramètre de cette commande permet de ne retenir que les fichiers ?

file

Quel paramètre de cette commande permet de filtrer sur les extensions ?

filter

### Réponses aux questions sur les commandes PowerShell - 2

Quelle commande PowerShell permet de filtrer les résultats renvoyés par Get-Childitem ?

where

### Réponses aux questions sur les commandes PowerShell - 3

Quelle commande PowerShell permet de supprimer les résultats renvoyés par Get-Childitem ?

remove-item