

10/01/2023

# RAPPORT PROJET

## PIINF-3-TSYS-POWERSHELL

Anthony Stucki  
CEFF INDUSTRIE

## TABLE DES MATIERES

1.	INTRODUCTION	2
1.1.	CAHIERS DES CHARGES	2
1.2.	LIEN GITHUB	2
2.	CONTAINS / EXTENSION / TRIMSTART – CODE	3
2.1.	QUELLE EST LA FONCTION DE CE BOUT DE CODE ?	3
2.2.	CONTAINS	3
2.3.	EXTENSION	3
2.4.	TRIMSTART	3
3.	GET-CHILDITEM / [MATH] ::ROUND – CODE	4
3.1.	QUELLE EST LA FONCTION DE CE CODE ?	4
3.2.	GET-CHILDITEM	4
3.3.	[MATH] ::ROUND	4
4.	DIRECTORY / FULLNAME - CODE	5
4.1.	QUELLE EST LA FONCTION DE CE CODE ?	5
4.2.	DIRECTORY	5
4.3.	FULLNAME	5
5.	CONCLUSION	6
5.1.	CONCLUSION PERSONNELLE	6
6.	SOURCES	6

# 1. INTRODUCTION

Ce rapport décrit un projet visant à développer un script PowerShell pour automatiser la gestion des fichiers sur le bureau. Ce script permettra de trier les fichiers en fonction de leur extension, et d'afficher des statistiques sur les répertoires choisis.

Ce projet vise à rendre la gestion des fichiers sur le bureau plus efficace et plus facile pour l'utilisateur. Je vais présenter des éléments du développement et les fonctionnalités implémentées.

## 1.1. CAHIERS DES CHARGES

Voici les points principaux du projet :

- Créer un programme de gestion automatique des fichiers présents sur le bureau
- Afficher les statistiques d'un répertoire

### 1.1.1. EXPLICATIONS

- Ce programme présente un menu permettant de choisir entre la gestion des fichiers sur le bureau ou l'affichage des statistiques d'un répertoire.
- Le script de gestion automatique de fichiers trie les différents fichiers dans des dossiers en fonction de leur extension.
- Le script pour afficher les statistiques d'un répertoire est capable d'afficher les fichiers d'un dossier, de compter le nombre de fichiers présents dans un dossier et d'afficher la taille d'un répertoire en giga-octets.

## 1.2. LIEN GITHUB

Tu souhaites tester mon projet ? Retrouve tous les scripts via mon lien GitHub.

Lien : <https://github.com/AnthonySTI/Projet-PS>

## 2. CONTAINS / EXTENSION / TRIMSTART – CODE

```
foreach ($item in $files) {  
  
    if ($word.Contains($item.Extension.TrimStart('.'))) {  
        New-Item -ItemType Directory -Path "$desktop\Documents\Microsoft Office\Word"  
-ErrorAction Ignore  
        Move-Item -Path $item.FullName -Destination "$desktop\Documents\Microsoft Of-  
fice\Word"  
    }  
}
```

### 2.1. QUELLE EST LA FONCTION DE CE BOUT DE CODE ?

Si la variable « \$word » contient l'extension du fichier courant, un nouveau répertoire appelé "Documents/Microsoft Office/Word" sera créé sur le bureau de l'utilisateur (s'il n'existe pas déjà) avec l'aide de la commandelette New-Item et le fichier courant sera déplacé dans ce répertoire avec l'aide de la commandelette Move-Item.

### 2.2. CONTAINS

La méthode « Contains » est utilisée pour vérifier si une chaîne de caractères contient une autre chaîne de caractères. Dans le cas de mon code, elle est utilisée pour vérifier si la variable « \$word » contient l'extension du fichier courant.

### 2.3. EXTENSION

La propriété « Extension » retourne l'extension du fichier, c'est-à-dire la partie de son nom après le dernier point sous forme de chaîne de caractères. Dans le cas de mon code, il est utilisé pour obtenir l'extension du fichier courant dans la boucle foreach.

### 2.4. TRIMSTART

La méthode « TrimStart » est utilisée pour supprimer les caractères spécifiés en début de chaîne. Dans le cas de mon code, elle est utilisée pour supprimer le point de départ de l'extension obtenue afin de la rendre comparable avec la variable « \$word » qui n'a pas de point de départ.

### 3. GET-CHILDITEM / [MATH] ::ROUND – CODE

```
$tailleEnOctets = (Get-ChildItem -Path $path -Recurse | Measure-Object -Property Length -Sum).Sum  
$tailleEnGigaOctets = [math]::Round($tailleEnOctets / 1gb, 2)  
Write-Host $tailleEnGigaOctets "Go"
```

#### 3.1. QUELLE EST LA FONCTION DE CE CODE ?

Ce code utilise des commandelettes de PowerShell pour obtenir la taille totale d'un répertoire et de tous ses sous-répertoires et l'afficher en gigaoctets.

#### 3.2. GET-CHILDITEM

La commandelette « Get-ChildItem » obtient tous les éléments de fichiers et répertoires sous un chemin spécifié et l'option -Recurse pour également inclure les sous-répertoires. Le résultat est passé à Measure-Object qui utilise la propriété Length pour calculer la somme de toutes les tailles de fichier. La propriété Sum de l'objet retourné par Measure-Object est utilisée pour stocker la taille totale des fichiers (en octets) dans la variable \$tailleEnOctets.

#### 3.3. [MATH] ::ROUND

La classe « [math]::Round » arrondit un nombre à un certain nombre de décimales. Il prend en compte deux arguments, le premier est le nombre à arrondir (\$tailleEnOctets / 1gb) pour obtenir la taille en gigaoctets, le deuxième est le nombre de décimales à conserver (2). Cette ligne permet d'obtenir la taille en gigaoctets qui est stockée dans la variable « \$tailleEnGigaOctets ».

## 4. DIRECTORY / FULLNAME - CODE

```
elseif($item.Directory){  
    New-Item -ItemType Directory -Path $desktop -Name "Autres" -ErrorAction  
Ignore  
    Move-Item -Path $item.FullName -Destination "$desktop/Autres"
```

### 4.1. QUELLE EST LA FONCTION DE CE CODE ?

Ce code utilise des commandes pour manipuler des fichiers et des dossiers. Il vérifie si l'objet « \$item » est un répertoire en utilisant la propriété « .Directory », puis s'il est un répertoire, il crée un nouveau dossier nommé « Autres » sur le bureau de l'utilisateur, s'il existe déjà il ignore l'erreur. Ensuite, il déplace l'objet « \$item » dans le nouveau dossier « Autres » en utilisant la propriété « .FullName » pour obtenir le chemin complet de l'objet « \$item », et la commande Move-Item pour déplacer l'objet.

### 4.2. DIRECTORY

La propriété « .Directory » de l'objet FileInfo en PowerShell retourne un objet DirectoryInfo qui représente le répertoire contenant le fichier ou répertoire représenté par l'objet FileInfo. Cette propriété retourne un booléen qui indique si l'objet représente un répertoire ou un fichier. Si cette propriété retourne "True", cela signifie que l'objet est un répertoire et si cette propriété retourne "False", cela signifie que l'objet est un fichier.

### 4.3. FULLNAME

La propriété « .FullName » de l'objet FileInfo en PowerShell retourne une chaîne qui contient le chemin complet de l'objet. Ce chemin comprend le nom du disque, les répertoires et le nom de fichier. Il permet de retrouver le chemin complet d'un fichier sur le disque.



## 5. CONCLUSION

En conclusion, ce projet a permis de développer un script PowerShell pour automatiser la gestion des fichiers sur le bureau. Le script permet de trier les fichiers en fonction de leur extension, et d'afficher des statistiques sur les répertoires choisis. Les fonctionnalités implémentées ont été présentées dans les titres précédents, et le lien GitHub a été fourni pour vous permettre de tester et d'utiliser le script. Ce projet est un excellent exemple de la puissance de PowerShell en termes d'automatisation et de gestion des systèmes Windows.

### 5.1. CONCLUSION PERSONNELLE

Pour mon avis personnel, j'ai aimé travailler sur ce projet. Il n'a pas l'air si compliqué mais je trouve ça vraiment utile, surtout pour la gestion automatique des fichiers présents sur le bureau.

Je suis quelqu'un qui n'a pas beaucoup d'ordre sur mon bureau alors j'utilise ce script de temps en temps afin de restructurer mon environnement Windows.

## 6. SOURCES

Pour mes sources, j'ai cherché des informations sur Google Chrome et ensuite j'ai découvert ChatGPT, c'est une intelligence artificielle développée par OpenAI.