

Fiche 02 : Utilisation des fichiers textes

1. Lecture d'un fichier

La commande Get-Content permet de lire en une seule étape les données du fichier.

```
Get-Content c:\temp\test.txt
Regnier/Daniel/Laon
Loisel/Pierre/Amiens
```

Get-Content traite les données lues du fichier comme un tableau, avec une ligne par élément du tableau. Exemple, lecture de la deuxième ligne du fichier (indice 1 du tableau \$ligne qui commence à 0) :

```
$ligne=Get-Content c:\temp\test.txt; $ligne[1]
Loisel/Pierre/Amiens
```

La même chose sans la variable \$ligne :

```
(Get-Content C:\temp\test.txt)[1]
```

Parcours des lignes du fichier comme une collection avec ForEach.

```
ForEach($ligne in Get-Content c:\temp\test.txt) {$ligne}
Regnier/Daniel/Laon
Loisel/Pierre/Amiens
```

Exemple, Parcourir le fichier pour afficher une information contenue dans la ligne :

```
ForEach($ligne in Get-Content c:\temp\test.txt) {$mot=$ligne.split("/");$mot[0]}
Regnier
Loisel
```

La méthode split retourne un tableau constitué des éléments délimités dans la ligne par le séparateur "/".

La même chose sans la variable \$mot :

```
ForEach($ligne in Get-Content c:\temp\test.txt) {($ligne.split("/"))[0]}
```

La même chose avec le pipeline d'objet et le descripteur \$_ :

```
Get-Content C:\temp\test.txt | ForEach-Object{$_ .split("/") [0]}
```

Encore plus fort, lire tous les fichiers .txt d'un dossier avec la commande Get-ChildItem (retourne la liste des fichiers).

```
Get-ChildItem c:\temp\*.txt | ForEach-Object {foreach($ligne in Get-Content $_){$ligne}}
```

La même chose avec deux pipelines d'objet (c'est chaud) :

```
Get-ChildItem c:\temp\*.txt | ForEach-Object {Get-Content $_} | ForEach-Object{$_}
```

2. Écrire dans un fichier

La commande Add-Content permet d'ajouter du texte dans un fichier.

```
Add-Content C:\temp\test1.txt "bonjour à tous"
```

La commande Clear-Content permet de vider un fichier.

```
Clear-Content C:\temp\test1.txt
```

La commande Out-File permet de rediriger le résultat d'un pipeline vers un fichier texte (unicode par défaut). Il est également possible d'utiliser les redirecteurs (> et >>).

```
ipconfig | findstr "Adresse" | Out-File c:\temp\test2.txt
```

```
ipconfig | findstr "Adresse" > c:\temp\test2.txt
```

Export-Csv permet de générer un fichier texte avec les données séparées par une virgule et Export-Clixml permet de créer un fichier de données au format XML.

```
Get-Process | Export-Clixml c:\temp\process.xml
```

L'exemple suivant remplace un texte "cherche" dans le fichier et place le résultat dans le même fichier avec la commande Set-Content qui écrase le fichier d'origine. Les () autour de Get-Content assurent que la commande est terminée avant la commande Set-Content qui accède au même fichier.

```
(Get-Content c:\temp\test3.txt) | ForEach-object {$ _ -replace "cherche", "trouve"} | Set-Content c:\temp\test3.txt
```