**TABLE FORMAT CONVERTER**asdasd

RAPPORT

PIINF-3-TSYS-PS-TE1 •

11 janvier 2023

Boulahdjar Ahmed

ICT-3

CEFF Industrie

Table des matières

[1. Introduction 2](#_Toc124324749)

[2. Script 3](#_Toc124324750)

[2.1. Conversion .csv A .JSON 3](#_Toc124324751)

[2.2. Default delimiter 3](#_Toc124324752)

[2.3. Installation d’un module 4](#_Toc124324753)

[3. Conclusion 5](#_Toc124324754)

[4. Sources 6](#_Toc124324755)

[5. Cahier des charges 7](#_Toc124324756)

[6. Github 8](#_Toc124324757)

# Introduction

Pour le module PIINF-3-TSYS-PS il nous à était demandé de faire un petit projet sur PowerShell.

Je suis partie sur une des idées de monsieur Stalder qui était de faire un script qui permet de convertir plusieurs formats de tableau avec quelque fonctionnalité comme pouvoir choisir le délimiter et le type d’encodage.

J’ai fait le projet entier avec une interface graphique, j’ai utilisé Windows Form qui peut être intégrer dans PowerShell.

# Script

Mon script basé sur du PowerShell avec une interface graphique Windows Form, il y a quelques partie du code que je vais expliquer et détailler.

## Conversion .csv A .JSON

import-csv -Delimiter "$Delimiter" $FilePath.FileName | ConvertTo-Json | Out-File -Encoding $EncodingType -Path  "$Destionation\$OutputFileName.json"

**Import-csv :** La commande est assez simple elle sert à importer le fichier .CSV

**-Delimiter :** Le paramètre de la commande import-csv qui va nous permettre de choisir le delimiter

On redirige tout cela dans la commande suivante 🡪

**ConvertTo-Json** : Qui va convertir l’entrée et le sortir en .JSON

On redirige le tout dans la commande suivante 🡪

Out-File : Va servir à envoyer la sortie vers le fichier spécifié

## Default delimiter

(Get-Culture).Textinfo.ListSeparator

Cette commande est assez courte est simple mais ma était utile.

Get-Culture : Va nous permettre d’obtenir des paramètres tels que la langue, clavier, format d’affichage etc…

On place la commande entre parenthèse pour spécifier l’information souhaiter

**(Get-Culture).Textinfo.ListSeperator :** Textinfo.ListSeparator c’est l’endroit où est stocké l’information du délimiter.

## Installation d’un module

Pour la conversion de fichier avec XLSX le module ImportExcel était nécessaire et n’était pas disponible de base, alors il m’a fallu intégrer dans mon script une partie qui vérifie si le module est disponible sinon il l’Install avec la commande suivante.

Install-Module ImportExcel -AllowClobber -Force # Installation du module ImportExcel -AllowClobber(Persmission) -Force(Focer l'installation)

**Install-Module :** Permet d’installer des modules

**ImportExcel :** Le nom du module

**-AllowClobber :** Passe outre les messages de prévention

**-Force :** Force l’installation et passe outre tout

# Conclusion

J’ai beaucoup appris en faisant ce projet et ça va me permettre de pas faire les mêmes fautes, comme par exemple la partie graphique que j’aurai pu faire en xml et ça m’aurai fait un énorme gain de temps.

Le projet est fonctionnel dans l’ensemble mais il reste quelque détail à régler au niveau du choix du délimiter qui est incompatible avec le fichier et le choix de l’encodage qui ne fonctionne pas encore très bien.

Dans l’ensemble je suis assez satisfait de mon travail.

# Sources

Stack overflow

Reddit

Documentation Microsoft

# Cahier des charges

**Voilà les points du cahier des charges**

• Pouvoir convertir un fichier par exemple csv en format xml, json, xlsx etc...

• Pouvoir choisir le fichier que vous voulez convertir

• Pouvoir choisir la sortie du fichier converti

• Pouvoir choisir le nom du fichier de sortie

• Pourvoir choisir le délimiter

• Pouvoir choisir le type d’encodage

• Pouvoir Installer un module si il est manquant

# Github

J’ai travaillé avec GitHub pour sauvegarder mon travail, avec un bon suivi dessus et de maitriser l’outil, dorénavant je l’utiliserai pour tous mes projets.

**Lien du projet sur GitHub** 🡪 https://github.com/Bouly/Table-Format-Converter