Migration de données de  
Win7 vers Win10



Ahmad Jano & José Carlos – FIN2

Lausanne - ETML

10.02.2022 – 10.03.2022

M. Laurent Duding

Table des matières

[1 Spécifications 3](#_Toc97599413)

[1.1 Titre 3](#_Toc97599414)

[1.2 Description 3](#_Toc97599415)

[1.3 Matériel et logiciels à disposition 3](#_Toc97599416)

[1.4 Prérequis 3](#_Toc97599417)

[1.5 Cahier des charges 3](#_Toc97599418)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc97599419)

[1.5.2 Fonctionnalités requises (du point de vue de l’utilisateur) 3](#_Toc97599420)

[1.5.3 Méthodes de validation des solutions 3](#_Toc97599421)

[1.6 Les points suivants seront évalués 3](#_Toc97599422)

[1.7 Validation et conditions de réussite 3](#_Toc97599423)

[2 Réalisation 4](#_Toc97599424)

[2.1 Mise en contexte 4](#_Toc97599425)

[2.2 Développement du script 4](#_Toc97599426)

[2.2.1 Backup favoris Firefox 4](#_Toc97599427)

[2.2.2 Restore des favoris vers Edge 5](#_Toc97599428)

[2.2.3 Backup données XnView 6](#_Toc97599429)

[2.2.4 Restore des données XnView 6](#_Toc97599430)

[2.2.5 Backup données C:/Data 6](#_Toc97599431)

[2.2.6 Restore des données C:/Data 6](#_Toc97599432)

[2.2.7 Backup raccourcis bureau 6](#_Toc97599433)

[2.2.8 Restore des raccourcis bureau 6](#_Toc97599434)

[2.2.9 Check de version de Windows 7](#_Toc97599435)

[3 Tests 7](#_Toc97599436)

[3.1 Dossier des tests 7](#_Toc97599437)

[4 Conclusion 7](#_Toc97599438)

[4.1 Bilan des fonctionnalités demandées 7](#_Toc97599439)

[4.2 Bilan personnel 7](#_Toc97599440)

[5 Divers 7](#_Toc97599441)

[5.1 Webographie 7](#_Toc97599442)

[5.2 Annexes 7](#_Toc97599443)

# Spécifications

## Titre

Migration de données d’une machine Windows 7 à Windows 10

## Description

Ce projet consiste à migrer les données spécifiées par le client tels que les favoris d’un navigateur, raccourcis de bureau, et des fichiers de configuration de certains logiciels.

## Matériel et logiciels à disposition

* Environnement de travail avec plusieurs ordinateurs.
* VirtualBox
* Visual Studio Code
* Python 3.1.1

## Prérequis

Connaissances de bases sur le langage de programmation Python.

Connaissances de base sur le fonctionnement des fichiers Windows 7 et Windows 10.

Connaissances de base sur la syntaxe JSON et SQLITE.

## Cahier des charges

### Objectifs et portée du projet (objectifs SMART)

Par groupe de deux, créer un script Python qui effectue une migration de données et configurations d’une machine Windows 7 vers une machine Windows 10 avec un délai de quatre semaine à partir du xx.xx.xxxx jusqu’au xx.xx.xxxx à l’ETML afin de préserver les configurations auxquelles l’utilisateur final a été auparavant habitué.

### Fonctionnalités requises (du point de vue de l’utilisateur)

* Equivalence des configurations du programme XnView.
* Equivalence des données du répertoire C:\Data.
* Equivalence des raccourcis du bureau.
* Transfert des favoris Firefox vers Microsoft Edge.

### Méthodes de validation des solutions

Regarder manuellement si les fichiers et les configurations ont bien été migrées avec des techniques tel que :

Favoris Firefox à Edge -> ouvrir les deux et regarder si c’est pareil

Configs XNView -> ouvrir le programme et regarder si c’est pareil

Données C:\Data -> regarder si les données sont équivalentes

Raccourcis bureau -> regarder si les raccourcis sont les mêmes

## Les points suivants seront évalués

* Le rapport
* La présentation
* Mode d’emploi
* Le script

## Validation et conditions de réussite

* Compréhension du travail
* Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
* Etat de fonctionnement du produit livré

# Réalisation

## Mise en contexte

L’idée est d’avoir un type de stockage portable comme une clé USB contant le script de migration, on la branche dans la machine Windows 7, le script va transférer les données qui seront migrées temporairement sur la clé USB, retirer la clé et la brancher sur la machine Windows 10 pour ensuite relancer le script qui va migrer les données stockées précédemment. La clé USB a été simulée avec l’utilisation des partages de dossier VirtualBox.

Afin de simuler l’environnement de migration, nous avons créé une machine virtuelle Windows 7 ainsi qu’une machine virtuelle Windows 10. Par la suite, nous avons procédé à l’installation de XnView sur les deux machines, tout en modifiant au maximum les paramètres de l’interface graphique du programme sur la machine Windows 7. Ensuite, nous avons installé Firefox sur la machine Windows 7 en ajoutant des favoris. Après cela, il a fallu créer un répertoire de données qui sera migré sur la machine Windows 10. Finalement nous avons rajouté quelques raccourcis sur le bureau.

## Développement du script

Avant de commencer le développement, il faut vérifier si les librairies utilisées par le script peuvent fonctionner autant sur Windows 7 que sur Windows 10. La version de python 3.1.1 convient parfaitement pour notre utilisation.

### Backup favoris Firefox

Les favoris Firefox sont stockés dans un fichier de base de données SQLite appelé ***places.sqlite*** dans le répertoire suivant :

%APPDATA%\Mozilla\Firefox\Profiles\(nom\_de\_profile)\places.sqlite

Le moteur de base de données SQLite fonctionne quasiment de la même manière que MySQL donc la syntaxe des requêtes sont similaires.

Les favoris de Edge sont stockés dans un fichier JSON avec le nom « Bookmarks » dans l’endroit suivant :

%LOCALAPPDATA%\Microsoft\Edge\User Data\Default\Bookmarks

L’objectif est donc de transformer nos favoris Firefox qui sont en SQLite vers le format JSON qui sera lisible par Edge. Pour ce faire, nous avons utilisé la libraire native pour SQLite inclue avec python : sqlite3***.***

Les tables concernées pour la récupération des favoris sont moz\_bookmarks et moz\_places. Le *titre* du favori est stocké dans la colonne name dans la table moz\_bookmarks et le *lien* du favori se trouve dans la colonne « url » dans la table moz\_places référencé par la clé étrangère fk dans moz\_bookmarks.

La requête suivante permet de relier les deux tables et récupérer le titre et l’URL des favoris :

**SELECT** moz\_bookmarks.title, moz\_places.url  
**FROM** moz\_bookmarks  
**INNER** JOIN moz\_places **ON** moz\_bookmarks.fk = moz\_places.id

L’étape suivante serait de convertir ces informations en format JSON. Il est impératif de comprendre la structure JSON que Edge peut lire. Après avoir jeté un coup d’œil sur le fichier Bookmarks, les favoris doivent être importés dans la liste children selon la hiérarchie suivante :

roots → bookmark\_bar → children

Un favori a la syntaxe JSON suivante :

{

"name": "Google",

"url": "https://www.google.com/?gws\_rd=ssl",

"type": "url",

"show\_icon": false

}

La partie du script qui s’occupe de récupérer les favoris en sqlite :

Une image contenant texte

Description générée automatiquement

La partie du script qui s’occupe de convertir notre objet en string JSON :

Une image contenant texte

Description générée automatiquement

Partie du script qui s’occupe de lancer les deux fonctions définies plus haut :

Une image contenant texte

Description générée automatiquement

### Restore des favoris vers Edge

Une image contenant texte

Description générée automatiquement

### Backup données XnView

Une image contenant texte

Description générée automatiquement

### Restore des données XnView

Une image contenant texte

Description générée automatiquement

### Backup données C:/Data

Une image contenant texte

Description générée automatiquement

### Restore des données C:/Data

Une image contenant texte

Description générée automatiquement

### Backup raccourcis bureau

Une image contenant texte

Description générée automatiquement

### Restore des raccourcis bureau

Une image contenant texte

Description générée automatiquement

### Check de version de Windows

Si on est sur Windows 7, on va lancer les fonctions qui s’occupent du backup, alors que si on est sur Windows 10, on va lancer les fonctions de récupération

Une image contenant texte

Description générée automatiquement

# Tests

## Dossier des tests

Lancer le script a été la principale source de tests

# Conclusion

## Bilan des fonctionnalités demandées

Equivalence des configurations du programme XnView. -> réussi

Equivalence des données du répertoire C:\Data. -> réussi

Equivalence des raccourcis du bureau. -> réussi

Transfert des favoris Firefox vers Microsoft Edge. -> réussi

Bilan : 100% réussi

## Bilan personnel

Très bon défi de créer un script python compatible W7 et W10

# Divers

## Webographie

<https://docs.python.org/3/library/sqlite3.html>

<https://docs.python.org/3/library/json.html>

<https://docs.python.org/3/library/platform.html>

<https://docs.python.org/3/library/os.html>

<https://docs.python.org/3/library/shutil.html>

## Annexes

Vidéo démonstration : demo\_migration.mp4

Mode d’emploi : tutoriel.pdf