

DOCUMENTATION TECHNIQUE : USB COPIER SCRIPT

1. Description

Le script `usb_copier.py` est un utilitaire Python conçu pour détecter automatiquement une clé USB spécifique par son nom de volume et copier récursivement son contenu vers un répertoire local prédéfini.

2. Spécifications Techniques

2.1. Environnement Requis

- **Système d'exploitation** : Windows
- **Dépendances** :
 - Python 3.x
 - Modules natifs : `os`, `shutil`, `time`
 - Module optionnel : `pywin32` (pour la détection améliorée des volumes)

2.2. Structure du Code

Variables de Configuration

```
python
USB_NAME = "WINSETUP" # Nom de volume à rechercher
DESTINATION_DIR = r"C:\Users\HP\Desktop\kevin" # Chemin absolu de destination
```

Fonctions Principales

find_usb_drive()

Signature : **def find_usb_drive() -> Optional[str]**

Description :

Parcourt les lecteurs de A: à Z: pour trouver un volume dont le nom contient **USB_NAME**.

Retourne :

- Chemin racine du lecteur (ex: "E:") si trouvé
- **None** si non trouvé

get_volume_name(drive_letter: str)

Signature : **def get_volume_name(drive_letter: str) -> Optional[str]**

Description :

Récupère le nom de volume d'un lecteur en utilisant l'API Windows.

Comportement :

- Tente d'utiliser **win32api** si disponible
- Retourne silencieusement **None** en cas d'erreur

copy_usb_contents(usb_path: str, dest_dir: str)

Signature : **def copy_usb_contents(usb_path: str, dest_dir: str) -> None**

Description :

Copie récursive avec les caractéristiques :

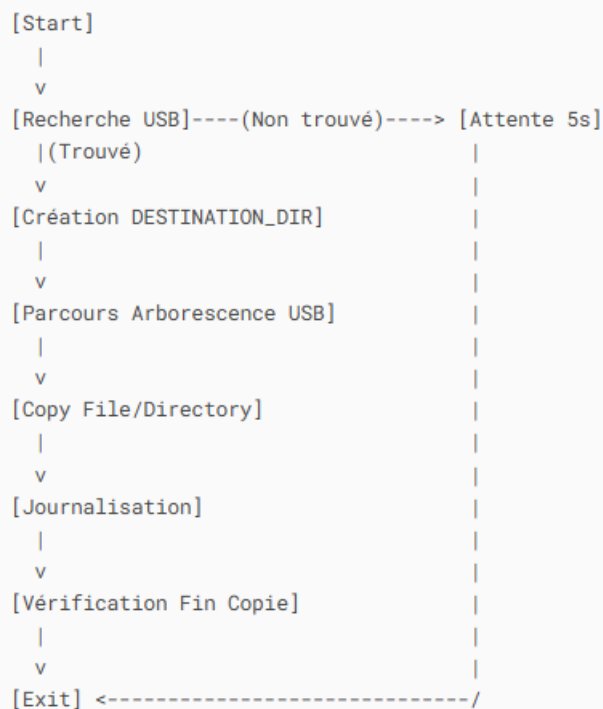
- Crée l'arborescence complète des répertoires
- Préserve les métadonnées des fichiers (**shutil.copy2**)
- Gestion d'erreurs par fichier/dossier
- Journalisation dans la console

2.3. Boucle Principale

Implémente un mécanisme de polling :

1. Intervalle : 5 secondes
2. Séquence :
 - Recherche du périphérique
 - Si trouvé → Copie → Sortie
 - Sinon → Attente → Nouvel essai

3. Diagramme de Flux



4. Journalisation

Le script produit des sorties console standard :

- Niveau INFO : Opérations normales
- Niveau ERROR : Échecs de copie (fichier verrouillé, permissions, etc.)

Format des messages :

```
[SUCCÈS] Copié : <source> -> <destination>  
[ERREUR] Échec copie de <chemin> : <message d'erreur>
```

5. Gestion des Erreurs

Le script gère :

- Absence de module **pywin32** (dégradation silencieuse)
- Fichiers/dossiers verrouillés
- Problèmes de permissions
- Espace disque insuffisant (via exceptions système)

6. Limitations Connues

1. Ne gère pas les montages réseau
2. Détection Windows uniquement
3. Taille maximale des fichiers limitée par la mémoire disponible
4. Unicode : dépend du système de fichiers USB

7. Sécurité

Considérations :

- Nécessite des droits d'écriture sur **DESTINATION_DIR**
- Ne valide pas l'intégrité des fichiers copiés
- Aucun chiffrement pendant le transfert

8. Extensibilité

Points de modification possibles :

1. Filtrage des fichiers par extension
2. Journalisation dans un fichier
3. Notification système en fin d'opération
4. Support multi-USB via configuration

9. Exemple d'Exécution

Sortie console typique :

```
Recherche de la clé USB 'WINSETUP'...  
Clé USB non détectée. Nouvelle tentative dans 5 secondes...  
Clé USB trouvée sur E:\  
Début de la copie...  
Copié : E:\setup.exe -> C:\Users\HP\Desktop\kevin\setup.exe  
Copié : E:\docs\ -> C:\Users\HP\Desktop\kevin\docs\  
[ERREUR] Échec copie de E:\locked.txt : [Errno 13] Permission denied  
Copie terminée avec succès !
```