

# Projet : Système d'Automatisation de Maintenance Informatique

## Documentation – Guide d'Utilisation et de Déploiement

Voici un Guide d'Utilisation et de Déploiement du script de Maintenance Informatique

« ***maintenance.py*** ».

<b>OBJECTIF .....</b>	<b>1</b>
<b>CONFIGURATION – PRÉREQUIS.....</b>	<b>1</b>
<b>INSTALLATION .....</b>	<b>2</b>
<b>FONCTIONNEMENT .....</b>	<b>2</b>
CONFIGURATION – FICHIERS TEMPORAIRES + OS .....	2
NETTOYAGE DE DOSSIER/FICHER TEMPORAIRE .....	3
RAPPORT.....	4
MAIN .....	5
EXÉCUTION .....	5
<b>ANNEXE.....</b>	<b>6</b>
CONSEILS D'UTILISATION .....	6
DÉPANNAGE.....	6
CONTACT .....	6

### Objectif

Le script « ***maintenance.py*** » permet de nettoyer les Fichiers et Dossiers temporaires sur différentes Systèmes d'exploitation (Windows, Linux). Il supprime les fichiers temporaires et génère un rapport d'activité, les actions réussies comme les erreurs rencontrées.

Avant tout, voyons les **Prérequis**.

### Configuration – Prérequis

Le script peut être exécuté :

- Sous Windows 10 et versions supérieures
- Sous Linux (Ubuntu et distributions similaires)

Avant de pouvoir exécuter le script « ***maintenance.py*** », il faut avoir :

- 1) Python 3.x installé
- 2) Les Droits Administrateur (Windows) ou lancer l'**Exécution** en « ***sudo*** » (Linux)
- 3) Comme OS : Windows 10+ ou Linux (Debian/Ubuntu/Centos)

Une fois ces prérequis remplis, nous pouvons passer à l'Installation et à l'Exécution du script.

## Installation

Pour Installer le script :

- 1) Récupérer le fichier « ***maintenance.py*** » sur la machine où l'on souhaite l'exécuter
- 2) Ouvrir un terminal et se déplacer avec la commande « `cmd` » vers le dossier où se trouve le script :
- 3) Exécuter le script :
  - a) Sous Windows avec la commande « ***python maintenance.py*** »
  - b) Sous Linux avec la commande « ***chmod +x maintenance.py && python3 maintenance.py*** » la 1<sup>ère</sup> fois puis « ***python3 maintenance.py*** » les fois suivantes

**Note :** Sous Linux, pour pouvoir exécuter un script (un programme), il faut le rendre exécutable (en donner les droits à l'utilisateur) ; c'est à ça que sert la commande « ***chmod +x maintenance.py*** ».

Voyons un peu comment fonctionne le script.

## Fonctionnement

Faisons un tour du script, avec ses fonctions et le reste de son contenu.

Commençons par la fonction en rapport avec la Configuration du système (quel est le système d'exploitation de la machine et quels sont donc les Dossiers temporaires à nettoyer).

### Configuration – Fichiers Temporaires + OS

Dans un 1<sup>er</sup> temps, le script vérifie quels sont les dossiers temporaires à nettoyer, en fonction du Système d'Exploitation (OS) ; ces fichiers varient selon si le script se lance sur un système Windows OU un système Linux :

Par défaut, le script nettoie les dossiers « ***/tmp*** » sous Linux. Sinon, il nettoie les dossiers « ***%TEMP%*** » sous Windows et tout autre dossier(s) similaire(s).

```
# Fonction pour déterminer les répertoires temporaires à nettoyer
def get_temp_dirs():
    system = platform.system() # Détection du système d'exploitation (Windows, Linux, etc.)
    temp_dirs = []             # Liste qui contiendra les chemins des Répertoires à nettoyer

    if system == "Windows":
        # Sous Windows, on ajoute %TEMP% et le répertoire Temp du système
        temp_dirs.append(tempfile.gettempdir()) # Ex: C:\\Users\\User\\AppData\\Local\\Temp
        temp_dirs.append(os.path.expandvars("%SystemRoot%\\Temp")) # Ex: C:\\Windows\\Temp
    elif system == "Linux":
        # Sous Linux, on ajoute /tmp et le répertoire temporaire par défaut
        temp_dirs.append("/tmp")
        temp_dirs.append(tempfile.gettempdir()) # Généralement aussi /tmp

    return temp_dirs # Retourne la liste des Répertoires temporaires à nettoyer
```

À la fin, il renvoie la liste des Dossiers temporaires à nettoyer.

Voyons maintenant comment le système (le script) nettoie les Dossiers temporaires trouvés (déterminés) avec les Fichiers (et les Sous-Dossiers) temporaires qui s'y trouvent.

## Nettoyage de Dossier/Fichier Temporaire

```
# Fonction qui nettoie un Répertoire temporaire donné (Fichiers + Sous-Dossiers)
def clean_temp_dir(path):
    files_deleted = 0      # Initialise un compteur de Fichiers temporaires supprimés
    dirs_deleted = 0       # Initialise un compteur de Dossiers temporaires supprimés
    logs = []              # Initialise une liste pour enregistrer les messages (succès ou erreurs) dans le Rapport

    # Vérifie si le chemin spécifié existe réellement sur le système
    if not os.path.exists(path):
        logs.append(f"Chemin inexistant : {path}") # Ajoute un message d'erreur si le chemin est invalide
        return files_deleted, dirs_deleted, logs    # Retourne les compteurs à zéro + log d'erreur

    # Parcours récursif du Répertoire, du bas vers le haut (topdown=False)
    # Cela permet de supprimer les Fichiers AVANT de tenter de supprimer les Dossiers
    for root, dirs, files in os.walk(path, topdown=False):

        # Boucle sur tous les Fichiers dans le Dossier courant
        for name in files:
            full_path = os.path.join(root, name) # Construit le chemin complet vers le Fichier
            try:
                os.remove(full_path)              # Tente de supprimer le Fichier
                logs.append(f"[OK] Fichier supprimé : {full_path}") # Ajoute un message de succès
                files_deleted += 1                # Incrmente le compteur de Fichiers supprimés
            except Exception as e:
                # En cas d'erreur (ex : permission refusée), enregistre l'erreur dans les logs
                logs.append(f"[ERREUR] Suppression Fichier {full_path} : {e}")

        # Boucle sur tous les Sous-Dossiers dans le Répertoire courant
        for name in dirs:
            full_path = os.path.join(root, name) # Construit le chemin complet vers le Sous-Dossier
            try:
                shutil.rmtree(full_path)          # Tente de supprimer récursivement le Sous-Dossier
                logs.append(f"[OK] Dossier supprimé : {full_path}") # Message de succès
                dirs_deleted += 1                # Incrmente le compteur de Dossiers supprimés
            except Exception as e:
                # En cas d'erreur (ex : permission refusée), enregistre l'erreur dans les logs
                logs.append(f"[ERREUR] Suppression Dossier {full_path} : {e}")

    # Retourne les résultats finaux : nombre de Fichiers et Dossiers supprimés + tous les logs (Succès ET Erreurs)
    return files_deleted, dirs_deleted, logs
```

- Dans un 1<sup>er</sup> temps, le script initialise à 0 le nombre de Dossiers et de Fichiers temporaires qui seront supprimés ainsi que le contenu qui sera ajouté au fichier de **Rapport** : voir plus bas.
- Dans un 2<sup>ème</sup> temps, le script recherche le Dossier temporaire donné en paramètre de la fonction sur le système ; si le dossier en question existe, le script s'y déplace. Sinon, il affiche un message d'erreur.
- Si le Dossier temporaire existe, en partant du dossier en question :
  - o Il supprime les Fichiers temporaires présents dans ce Dossier ; à chaque tentative, un message sera ajouté au Rapport, qu'elle ait réussi ou pas.
  - o Si le Dossier contient des Sous-Dossiers, il se déplace dans ces sous-dossiers en descendant un peu plus dans l'arborescence ; à chaque tentative, un message sera aussi ajouté au Rapport, qu'elle ait réussi ou pas
  - o ... Et ainsi de suite jusqu'à ce qu'il n'y ait plus de Fichiers ni de Sous-Dossiers temporaires qui puissent être supprimés (dans le Dossier temporaire donné en paramètre).

À la fin du script, pour chaque fois où il a exécuté une commande pour supprimer un Fichier ou un Dossier temporaire, un message est envoyé dans un fichier log de Rapport avec le retour (Succès ou Erreur) de la commande en question.

## Rapport

Quand le script est exécuté pour la 1<sup>ère</sup> fois, un fichier nommé « **rapport\_maintenance.txt** » est généré dans le répertoire utilisateur (ou dans le même dossier que celui où se trouve le fichier « **maintenance.py** ») : dans ce fichier, on ajoute tous les « log » ; si les commandes lancées tout au long de l'exécution du script ont Réussi (OK) ou échoué (ERREUR).

```
# Fonction qui écrit les résultats dans un fichier log
def generate_report(log_path, report_data):
    with open(log_path, "a", encoding="utf-8") as log_file:
        # Ajoute un titre avec l'horodatage
        log_file.write(f"\n\n--- Rapport de Maintenance ({datetime.now()}) ---\n") # "--- Rapport de Maintenance (2025-05-30 11:22:04.219161) ---"
        log_file.write(f"\n\n--- Rapport de Maintenance ({datetime.now().strftime('%d/%m/%Y %H:%M:%S')} ---\n") # "--- Rapport de Maintenance (30/05/2025 11:25:41) ---"
        for entry in report_data:
            log_file.write(entry + "\n") # Écrit chaque ligne du rapport
```

- D'abord, le script ajoute une (1<sup>ère</sup>) ligne sous la forme « --- **Rapport de Maintenance (jour/mois/année heure:minutes:secondes)**--- »
- Après quoi, pour chaque Dossier temporaire qu'il a parcouru et nettoyé avec ses Fichiers temporaires (et ses Sous-Dossiers) :
  - o Il donne son adresse « absolue » (ex : **C:\Users\matth\AppData\Local\Temp**)
  - o Il indique dans le Rapport le nombre de Fichiers ET de (Sous-)Dossiers temporaires qui ont été supprimés dans ce Dossier.
- Enfin, il ajoute les messages retournés par les commandes lancées au cours des suppressions des Fichiers/Dossiers temporaires, que les commandes aient réussi ou pas.

**Note :** À moins d'être supprimé, le fichier de Rapport sera rempli à chaque exécution du script en suivant le principe ci-dessus, avec de nouvelles lignes.

Voici un extrait de fichier « **rapport\_maintenance.txt** »

```
--- Rapport de Maintenance (30/05/2025 15:38:18) ---

[NETTOYAGE] Répertoire : C:\Users\matth\AppData\Local\Temp
-> Fichiers supprimés : 4
-> Dossiers supprimés : 0
-> Erreurs :
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\Diagnostics\WINWORD\Additional\Additional1748604199502272400_A92CFC12-CC0E-43DB-9771-946
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\Diagnostics\WINWORD\Additional\Additional1748604199503004700_A92CFC12-CC0E-43DB-9771-946
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\Diagnostics\WINWORD\Primary1748604199499906900_A92CFC12-CC0E-43DB-9771-94660F27E65D.log
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\Diagnostics\WINWORD\Primary1748604199500631400_A92CFC12-CC0E-43DB-9771-94660F27E65D.log
- [ERREUR] Suppression Dossier C:\Users\matth\AppData\Local\Temp\Diagnostics\WINWORD\Additional : [WinError 32] Le processus ne peut pas accéder au fichi
- [ERREUR] Suppression Dossier C:\Users\matth\AppData\Local\Temp\Diagnostics\WINWORD : [WinError 32] Le processus ne peut pas accéder au fichier car ce f
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\09d5e272-7596-4a4f-85d5-a027c6296daa.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\0c15e5a4-1ed1-448c-a014-891f2b7ed90f.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\114b9883-859a-4793-ae63-8e76a88f7d4b.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\18a55853-ef7f-428a-ad9f-a6430cf6a73b.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\1b306ae9-a191-4019-a4f2-f397d738710e.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\1c285fe-17ef-4fda-90da-9091060430f2.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\1f6b33cd-2ce8-4da8-99db-1417da1227a7.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\29ac8cd7-c840-4d92-8cf2-115350ba666d.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\36ef0dd1-f3e8-45a5-b4e0-504a41f8c609.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\4243ce81-d7e7-4419-9201-1697c38b6117.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\47e5db3f-87f0-4102-ab4b-742ae84383d3.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\57e4664a-7cd0-4780-bd96-8bc8a51c0620.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\5fcd7d21-fd72-45a5-a867-8ce06da91c89.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\62aebac-f814-41b6-840e-6ad34928683a.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\73e37f1e-19b8-4f79-8ea4-97a33b3e4fe8.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\7c7ef6b8-ede7-4229-beb6-a1d604a620d8.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\87ea4932-1b68-49c0-9fc0-ba9b2a670399.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\90091d8a-6698-4402-900f-83023b8a6da3.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\9192936f-62ea-4e14-a17d-75c509320afd.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\9a0088bf-23a4-4433-9ffb-b3956d01fda6.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\9da71534-1fe0-44ed-b25f-a02aa43dadcb.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\ae5db3f-87f0-4102-ab4b-742ae84383d3.tmp : [WinError 32] Le processus ne peut pas accède
- [ERREUR] Suppression Fichier C:\Users\matth\AppData\Local\Temp\afaa79fd-e3dc-43ec-af19-a33f842belfc.tmp : [WinError 32] Le processus ne peut pas accède
- [OK] Fichier supprimé : C:\Users\matth\AppData\Local\Temp\bc3902d8132f43e3ae086a0009979fa88.db
```

Jetons un œil à la fonction principale.

## Main

```
# Fonction principale exécutée lorsque le script démarre
def main():
    temp_dirs = get_temp_dirs() # Récupère les Répertoires temporaires à nettoyer
    report = []                 # Initialise la liste des lignes du rapport qui seront ajoutées au fichier "rapport_maintenance.txt"

    for temp_dir in temp_dirs:
        report.append(f"\n[NETTOYAGE] Répertoire : {temp_dir}")
        files_deleted, dirs_deleted, errors = clean_temp_dir(temp_dir)

        # Ajout dans le fichier "rapport_maintenance.txt" du nombre de Fichiers et Dossiers temporaires supprimés
        report.append(f"    -> Fichiers supprimés : {files_deleted}")
        report.append(f"    -> Dossiers supprimés : {dirs_deleted}")

        # En cas d'erreur (de nettoyage)
        if errors:
            report.append("    -> Erreurs :")
            # Ajout des erreurs détectées
            for err in errors:
                report.append(f"        - {err}")
        # Sinon
        else:
            report.append("    -> Aucune erreur détectée.")

    # Détermine le chemin du fichier Rapport dans le répertoire utilisateur OU dans le répertoire où se trouve le script actuellement exécuté
    # log_file = os.path.join(os.path.expanduser("~"), "rapport_maintenance.txt") # le répertoire utilisateur
    log_file = os.path.join(os.path.dirname(os.path.abspath(__file__)), "rapport_maintenance.txt") # le chemin absolu du script actuellement exécuté

    generate_report(log_file, report) # Génère le Rapport

    print(f"[INFO] Rapport généré : {log_file}") # Affiche l'emplacement du fichier "rapport_maintenance.txt"
```

- Dans un 1<sup>er</sup> temps, la fonction récupère la liste des Dossiers Temporaires à nettoyer.
- Dans un 2<sup>ème</sup> temps, elle prépare la liste des éléments (des lignes) qui seront ajoutés au Rapport.
- Ensuite, elle remplit petit à petit la liste pour le Rapport, en fonction des résultats retournés par les commandes exécutées dans les différentes fonctions.
- À la fin, elle remplit le fichier du Rapport avec le contenu de la liste initialisée au début de la fonction.

Pour terminer, revoyons comment exécuter le script.

## Exécution

Pour exécuter le script :

- 1) On récupère le script « ***maintenance.py*** » pour la mettre sur notre machine
- 2) On ouvre un terminal
- 3) On se déplace avec la commande « ***cd*** » jusqu'au dossier où se trouve le script « ***maintenance.py*** »
- 4) On lance la commande appropriée à notre Système d'Exploitation :
  - a) Sous Windows : ***python maintenance.py***

```
PS D:\POEI - Ingénieur Linux Embarqué\Formation\Test Logiciels\Projet> python .\maintenance.py
[INFO] Rapport généré : D:\POEI - Ingénieur Linux Embarqué\Formation\Test Logiciels\Projet\rapport_maintenance.txt
```

- b) Linux : « ***chmod +x maintenance.py && python3 maintenance.py*** » la 1<sup>ère</sup> fois puis « ***python3 maintenance.py*** » les fois suivantes.

```
matthieu@matthieu-VirtualBox:~/POEI/Test Logiciel$ chmod +x maintenance.py && python3 maintenance.py
[INFO] Rapport généré : /home/matthieu/POEI/Test Logiciel/rapport_maintenance.txt
```

## Annexe

Voyons quelques Conseils d'Utilisation.

### Conseils d'Utilisation

Il est conseillé de :

- Exécuter régulièrement le script pour éviter l'accumulation de fichiers temporaires.
- Si des erreurs de permissions apparaissent, vérifier les droits d'accès ou lancer le script avec des privilèges administratifs (avec la commande « **sudo** » sous Linux).
- Consulter le Rapport après chaque exécution, pour s'assurer que le nettoyage s'est déroulé correctement.

Parlons des problèmes potentiels, leurs causes et comment les résoudre.

### Dépannage

Problème	Cause possible	Solution recommandée
Fichiers non supprimés	Permissions insuffisantes	Lancer le script en administrateur (avec la commande « sudo » sous Linux)
Script non démarré	Python non installé ou mal configuré	Installer Python 3, vérifier le PATH
Rapport non généré	Problème d'écriture dans le dossier	Vérifier les droits d'écriture
Chemin temporaire inexistant	Système avec configuration non standard	Modifier le script si besoin

Si vous souhaitez contacter le développeur du script, vous trouverez ses coordonnées plus bas.

### Contact

Pour toute question ou problème lié au script, vous pouvez contacter :

- **Nom du développeur** : SAUVAGEOT Matthieu
- **Email** : [matthieu.sauvageot@outlook.fr](mailto:matthieu.sauvageot@outlook.fr)