



Hardis Group dans l'équipe Reflex

Rapport de stage

1^{er} année BTS SIO

Sommaire :

1. Présentation de l'entreprise	3
1.1. Situation géographique.....	4
1.2. Organigramme de l'entreprise	4
2. Mes missions	5
3. Interface.....	5
3.1. Exemple grossit	5
4. Page application	6
4.1. Interface graphique après la connexion à la base de données :	6
4.2. Connexion à une base de données :	6
4.3. Page des paramètres généraux :	7
4.4. Page contexte :	7
4.5. Page correspondance :	7
4.6. Page des paramètres techniques :	8
4.7. Pages Paramètre supplémentaire :	8
4.8. Page d'initialisation des interfaces	10
4.9. Page d'exportation	10
5. Cette semaine :	11
5.1. Connexion à une base de données AS400	11
5.1.1. Problème lors de la connexion	11
5.1.2. Pilote Manquant (driver)	12
5.2. Découpage des fichiers volumineux	13
5.2.1. Utilisation du découpage	13
5.2.2. Adaptation en fonction	13
6. Conclusion	14
7. Tableau Ressenti	14

1.Présentation de l'entreprise

Hardis est une société de services et de conseils en systèmes d'information, son siège social implantée à Seyssinet-Pariset. Créée en 1984, c'est l'un des leaders français du développement et de l'intégration de logiciels. L'entreprise compte aujourd'hui plus de 1300 collaborateurs répartis sur une trentaine d'agences répartit dans le monde. En 2022, elle a réalisé un chiffre d'affaires de 135 millions d'euros.

L'entreprise intervient dans divers secteurs : distribution, industrie, services, administrations. Ses domaines d'expertise couvrent la transformation digitale, les infrastructures cloud et le développement d'applications métiers. Cette polyvalence lui permet d'accompagner ses clients dans leurs projets numériques.

Résolument tournée vers l'innovation, Hardis dispose de centres d'expertise dans le cloud, le big data, la mobilité et les objets connectés. La société consacre chaque année 20% de son chiffre d'affaires à la R&D.

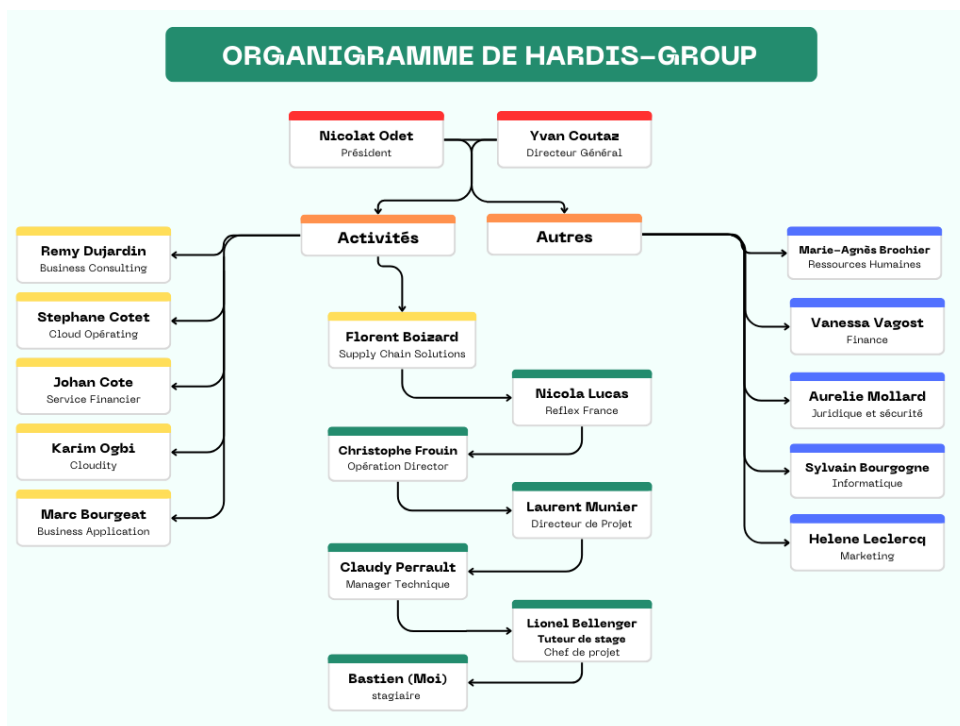
1.1. Situation géographique

Le siège social de l'entreprise est situé à Seyssinet-Pariset, ou je fais mon stage.



1.2. Organigramme de l'entreprise

Voici comment se compose de manière général l'entreprise :



2. Mes missions

Il y a un nombre d'interfaces déterminé par une base de données, et ces interfaces sont exécutables en ligne de commande, qui extrait de base de données différentes informations (des code activité et dépôts) qui sont enregistré dans un fichier txt. Le but est de rendre graphiques toutes ces opérations afin de rendre la tâche plus simple. Elle comprend la connexion à une base de données parmi trois types (MSSQL, AS400, ORACLE) qui ont chacun des paramètres de connexion différents et comme on peut le voir dans l'onglet 'Paramètres'.

3. Interface

Les interfaces sont des programmes en Python, initialement utilisés en ligne de commande, qui prennent divers paramètres lors de leur lancement (voir page "Général" ou "Correspondance"). Ces interfaces sont des EDI (Échanges de Données Informatisés) qui, à partir du contenu d'une base de données, génèrent des données de longueur fixe identifiées par un code rubrique.

3.1. Exemple grossit

INT_4D est une interface qui concerne les numéros de série de clients. Bien qu'elle ne contienne qu'une seule rubrique, le principe reste le même que les autres. Il y a une requête principale, et pour chaque rubrique, une autre requête est effectuée afin de vérifier si elle contient des données. Si c'est le cas, la rubrique est appelée pour écrire son contenu dans le fichier texte cette opération en boucle pour chaque lignes reçus par la requête principal ce qui peut prendre un peu de temps à l'opération.

```
def rubrique_110(ligne, sequence, jsonData):
    Code_activite = 
    Code_article = 
    Numero_de_serie = 
    Code_regroupement_NS = 
    # recherche s'il existe des correspondances pour le code activité
    if Code_activite in jsonData["code_act"]:
        Code_activite = jsonData["code_act"][Code_activite].ljust(3, ' ')

    rub_110 = str(sequence).zfill(
        7) + "HL4D110" + Code_activite + Code_article + Numero_de_serie + Code_regroupement_NS

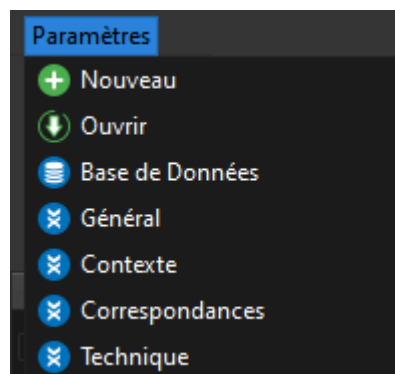
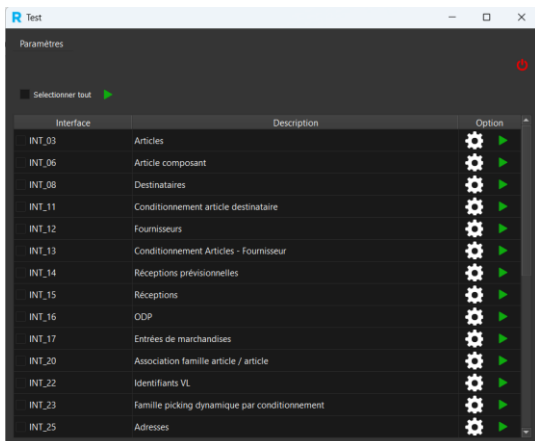
    output_file.write(rub_110 + "\n")
    sequence += 1
    return sequence
```

4. Page application

L'application contient différentes pages graphiques et voici à quoi elles servent.

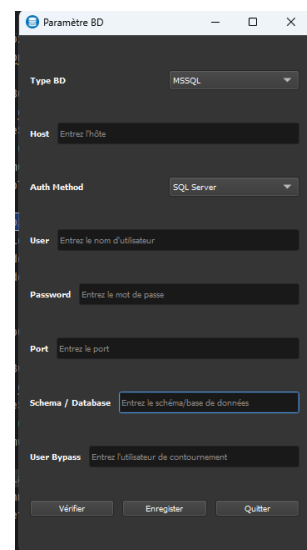
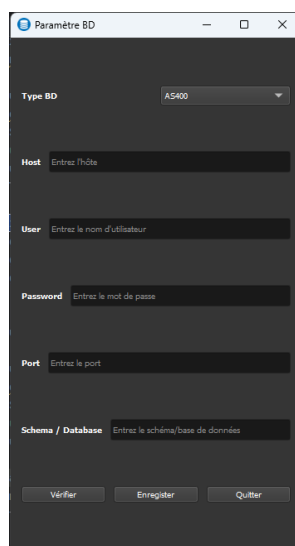
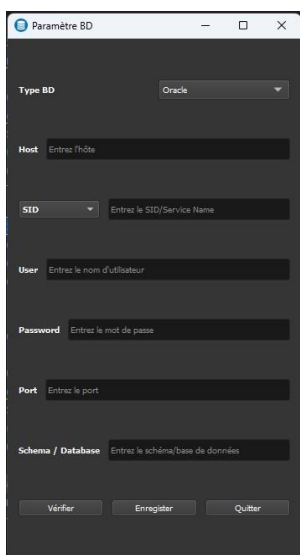
4.1. Interface graphique après la connexion à la base de données :

Ici on peut voir une partie des interfaces qui vont être exécuté ainsi que le menu déroulant.



4.2. Connexion à une base de données :

En fonction de la base choisit :



4.3. Page des paramètres généraux :

Paramètre de base lors de l'envoi de la requêtes SQL pour toute les interfaces

Paramètres Généraux

Code Do * ADM

MESD * RE+

Ref. Hvt Stock REPRISE

Include Art

Cor. Cmde CRC

DLC ver DDM ☒

Art. désactivés ☐

Rempl fourm ret ☒ 0

Enregistrer Quitter

4.4. Page contexte :

Permet de visualiser et d'enregistré les valeurs sélectionnées

Contexte

Tous les dépôts ☒ Toutes les activités ☐

Liste des dépôts

	Code	Libellé
<input type="checkbox"/>	CHA	CENPAC Disponible
<input type="checkbox"/>	CVB	CENPAC CVB
<input type="checkbox"/>	FR1	RAJA PARIS NORD 2

Listes des activités

	Code	Libellé
<input type="checkbox"/>	LGS	Raja France

Enregistrer Quitter

4.5. Page correspondance :

Initialiser via du JSON situé dans une base de données

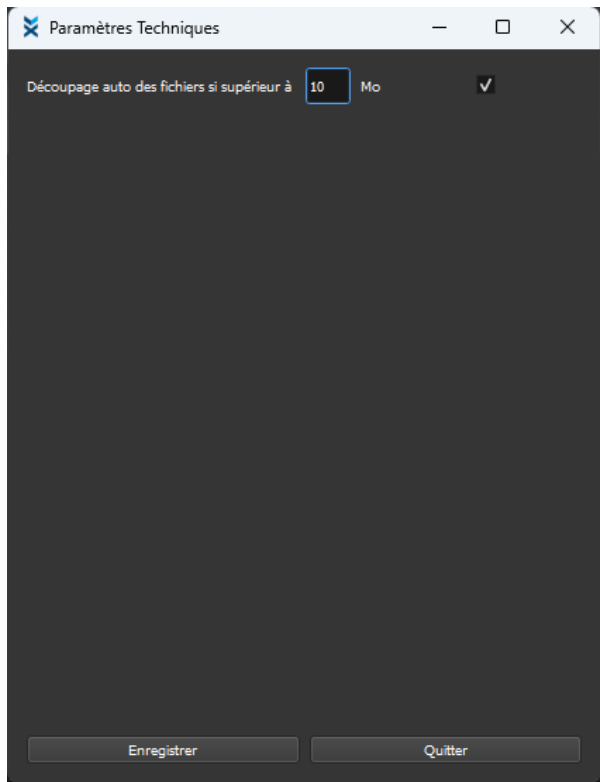
Correspondances

Key	Value
code_dpo	xxx
code_act	xxx
code_vl	xx
code_proprietaire	xxx
code_qual	xxx
code_type_supp	XXX YYY
code_taille	XXX YYY

Enregistrer Quitter

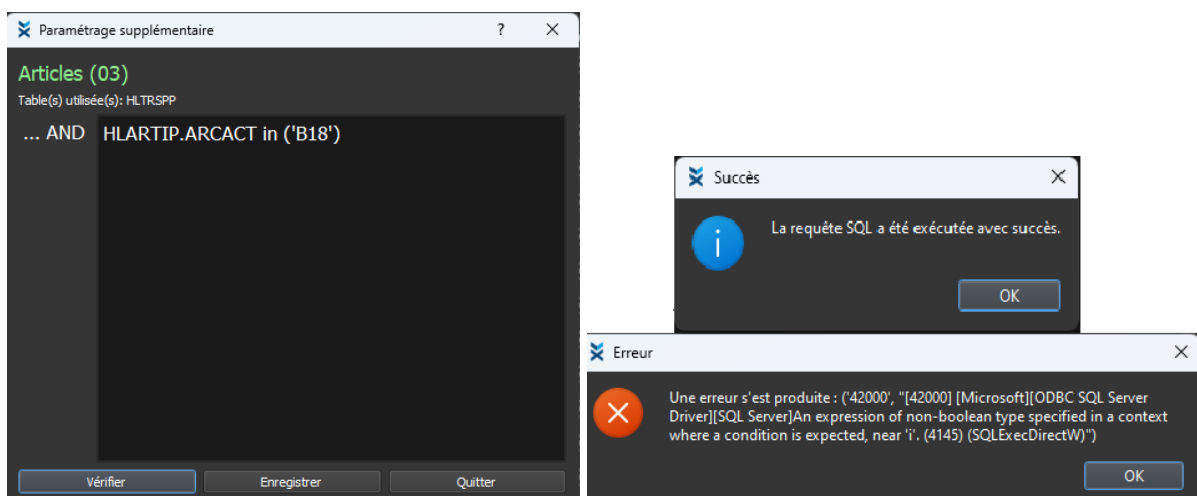
4.6. Page des paramètres techniques :

Permet le découpage du fichier généré par l'interface si la taille du fichier est supérieure à celle choisie (non obligatoire). Dans ce cas, le fichier est découpé via trois programmes Python déjà existants. Le processus de découpage comprend le tri des données des rubriques générés dans différents fichiers texte.



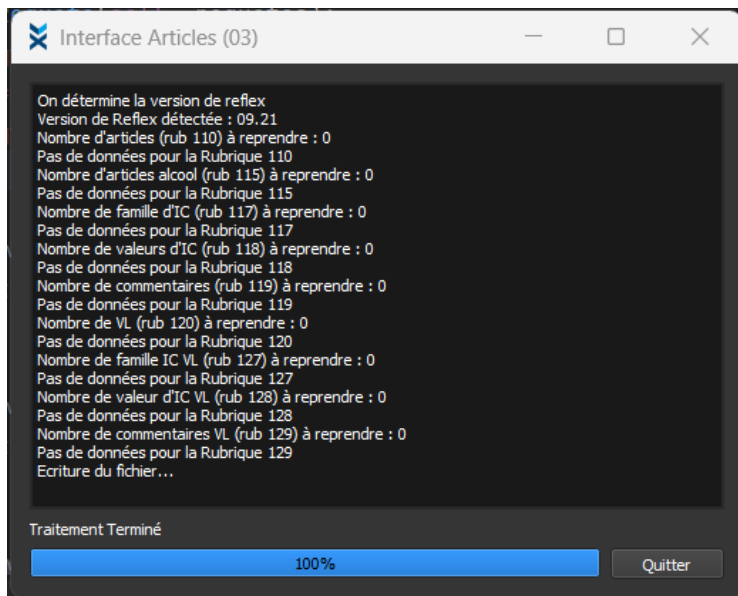
4.7. Pages Paramètre supplémentaire :

Permet d'ajouter du SQL en plus à la requête si l'on veut préciser la recherche et de pouvoir vérifier si la requête est juste ou non.



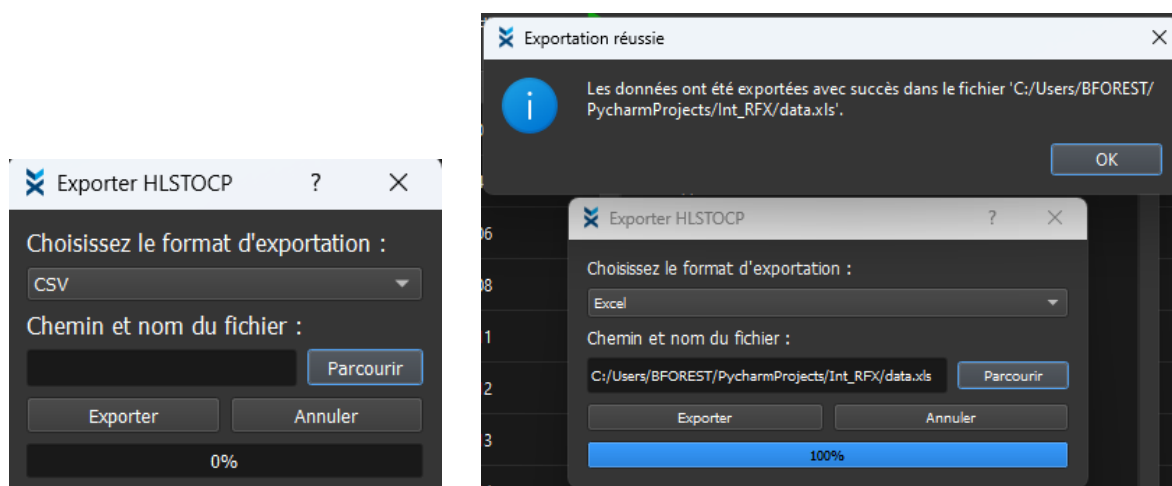
4.8. Page d'initialisation des interfaces

Cette page permet de voir le suivi de l'avancement de l'opération des interfaces. Le bouton quitter est non cliquable jusqu'à ce que l'opération soit finis comme ici. Le titre et le contenu change en fonction de l'interface choisit.



4.9. Page d'exportation

Cette page permet d'exporter les données de la table HLSTOCP à la fin de l'exécution de l'interface 17 dans fichier csv ou Excel (.xls) avec affichage en cas d'erreur ou de réussite.



5. Cette semaine :

Lors de ma quatrième semaine, j'ai commencé à finaliser l'application en cours. J'ai ajouté des messages d'erreur ou de réussite sur les validations d'actions, telles que la création d'un nouveau fichier en cas de première utilisation ou l'exportation des données, ainsi que sur les boutons de vérification pour les requêtes supplémentaires et la connexion aux bases de données. Il ne reste plus qu'à ajouter les quelques dernières fonctionnalités présentées dans ce rapport et à effectuer des tests pour vérifier qu'il n'y a pas d'erreur, quel que soit le type de base choisie ou l'interface utilisée.

5.1. Connexion à une base de données AS400

Lorsqu'on veut utiliser les différentes interfaces, il faut se connecter à une base de données du client. Il y a trois types de bases de données auxquelles on peut se connecter : Oracle, MSSQL et AS/400.

5.1.1. Problème lors de la connexion

Lors de la connexion à une base de données de type AS/400, j'ai rencontré de nombreuses erreurs. J'ai tout d'abord essayé le package `ibm_db`, mais j'ai pris énormément de temps à essayer de comprendre pourquoi je n'arrivais pas à l'installer. De plus, PyCharm (l'IDE que j'utilise) ne reconnaît pas le package et le temps de connexion est trop long.

5.2. Pilote Manquant (driver)

Après plusieurs heures de recherche, j'ai fini par comprendre qu'il manquait un pilote (IBM i Access ODBC Driver) pour se connecter à divers types de bases de données, dont AS/400. J'ai alors utilisé le package « `pyodbc` », que j'utilisais déjà pour les deux autres types de bases de données. J'ai donc ajouté la prise en charge d'AS/400 à l'application, en plus des autres types de bases déjà là.

5.3. Découpage des fichiers volumineux

Si le fichier généré par l'interface est trop volumineux, il est nécessaire de le découper sans altérer son contenu en évitant les coupures au mauvais endroit dans le fichier. Pour résoudre ce problème, un découpage automatique a été défini dans la page "Paramètres techniques".

5.3.1. Utilisation du découpage

Trois programmes de découpage existaient déjà mais elles n'étaient disponible qu'en ligne de commande :

- `Decoupe_dest` : Utilisé pour l'interface 8
- `Decoupe_stock` : Utilisé pour l'interface 17
- `Decoupe_art` : Utilisé pour toutes les autres interfaces

Ces programmes effectuent un "découpage" qui regroupe chaque rubrique ensemble dans le même fichier (5 rubriques, 5 fichiers).

5.3.2. Adaptation en fonction

J'ai rassemblé ces trois programmes Python en un seul fichier contenant trois fonctions pouvant être appelées en fonction du cas spécifique. La variable `num_int` (en orange) permet de récupérer le numéro de l'interface avec une expression régulière (regex) du nom du fichier texte. La variable `HLfile` (en bleu) contient le chemin du fichier texte généré et son nom. La variable `rows` (en vert) contient la taille minimale pour effectuer le découpage. Enfin, l'appel de la fonction qui convient (en rouge) est effectué avec ces paramètres.

```
file_size = round((os.path.getsize(file_path)) / (1024 * 1024), 2)
match = re.search(pattern, HLfile)
if match and rows[0][0] <= file_size:
    num_int = match.group(1)
    if str(num_int) == "17":
        decoupe_Stock(HLfile, rows[0][0])
    elif str(num_int) == "08":
        decoupe_Dest(HLfile, rows[0][0])
    else:
        decoupe_Art(HLfile, rows[0][0])
```

6. Conclusion

Ce stage dans le développement informatique chez Hardis Group a été une expérience enrichissante qui m'a permis d'approfondir mes connaissances en Python, notamment sur les paquetages pour les bases de données et de découvrir la communication entre plusieurs classes de façon dynamique. Je trouve cette façon d'apprendre vachement plus intéressante. C'est une expérience très positive pour ma part.

7. Tableau Ressenti

Lundi	Mardi	Mercredi	Jeudi	Vendredi
