Documentation Projet VTOMJOB

Table des matières

1. Base de données ……………………………………………………………….…2
   1. Schéma MCD ………………………………………………………………….2
   2. Tables et attributs …………………………………………………………..3
2. Application………………………………………………………………….…………5

2.1 Diagramme de classe ……………………………………………………..5

2.2 Diagramme de flux………………………………………………………….6

1. IHM………………………………….………………………………………………………..

3.1 Fenêtre Liste des Jobs……………………………………………………..…

3.2 Fenêtre Gestion du Job………………………………………………………

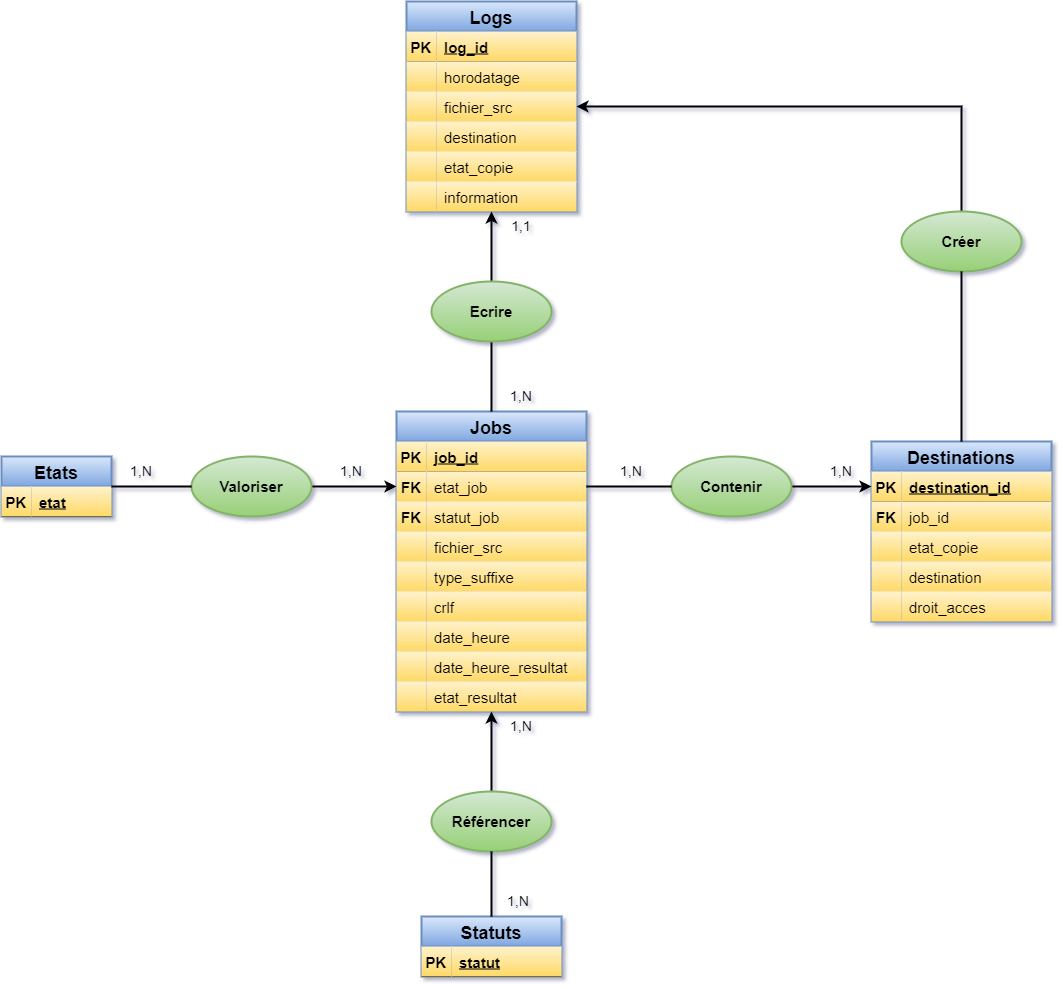
3.3 Fenêtre Logs……………………………………………………………….….….

1. Mise en œuvre…………………………………………………………..………
   1. Base de données………………………………………………………………
   2. Application………………………………………………………………………

1 - Base de données

La base de donnée est sur SQL Server

**1.1** - Schéma MCD



**1.2 -** Tables et attributs

* **Table Etats**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nom** | **Type de données** | **Description** |
| **PK** | etat | VARCHAR(25) NOT NULL | Clé primaire et libellé d’un **état**  Valeurs :  « Succès »  « Echec »  « En cours »  « En attente » |

* **Table Statuts**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nom** | **Type de données** | **Description** |
| **PK** | statut | VARCHAR(25) NOT NULL | Clé primaire et libellé d’un **statut**  Valeurs :  « Actif »  « Inactif » |

* **Table Jobs**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nom** | **Type de données** | **Description** |
| **PK** | **job\_id** | INT NOT NULL | Clé primaire |
| **FK** | etat\_job | VARCHAR(25) NOT NULL | Clé étrangère de **etat\_id**, change en fonction de l’état du job. Valeurs possibles  - En attente  - En cours |
| **FK** | statut\_job | VARCHAR(25) NOT NULL | Clé étrangère de **statut\_id**, change en fonction du statut du job |
|  | fichier\_src | VARCHAR(150) NOT NULL | Chemin du **fichier source** de la copie |
|  | type\_suffixe | TINYINT NOT NULL | Permet de savoir s’il faut la date ou la date et l’heure pour un job. Valeurs possibles :  **0** : ni date ni heure  **1** : date seulement  **2** : date et heure |
|  | crlf | TINYINT NOT NULL | Permet de savoir s’il faut **remplacer** les **sauts de ligne** (caractère « LF ») en **fin de ligne** (caractère « CRLF »). Valeurs possibles :  **0** : Pas de traitement  **1** : Traitement |
|  | date\_heure | VARCHAR(30) NOT NULL | Horodatage du **dernier lancement** du Job |
|  | date\_heure\_resultat | VARCHAR(25) NOT NULL | Horodatage du **dernier résultat** du Job |
|  | etat\_resultat | VARCHAR(25) NOT NULL | Etat du **dernier résultat** du Job. Valeurs possibles  - Succès  - Echec |

* **Table Destinations**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nom** | **Type de données** | **Description** |
| **PK** | **destination\_id** | INT NOT NULL | Clé primaire |
| **FK** | job\_id | INT NOT NULL | Clé étrangère de **job\_id** de la table **Job** |
|  | etat\_copie | VARCHAR(25) NOT NULL | **Etat** de la copie. Valeurs possibles  - Succès  - Echec |
|  | destination | VARCHAR(150) NOT NULL | **Destination** de la copie. Contient le **chemin complet** du fichier |
|  | droit\_acces | VARCHAR(25) | Droits d’accès du fichier collé. Valeurs possibles  - Lecture seule  - Lecture et écriture |

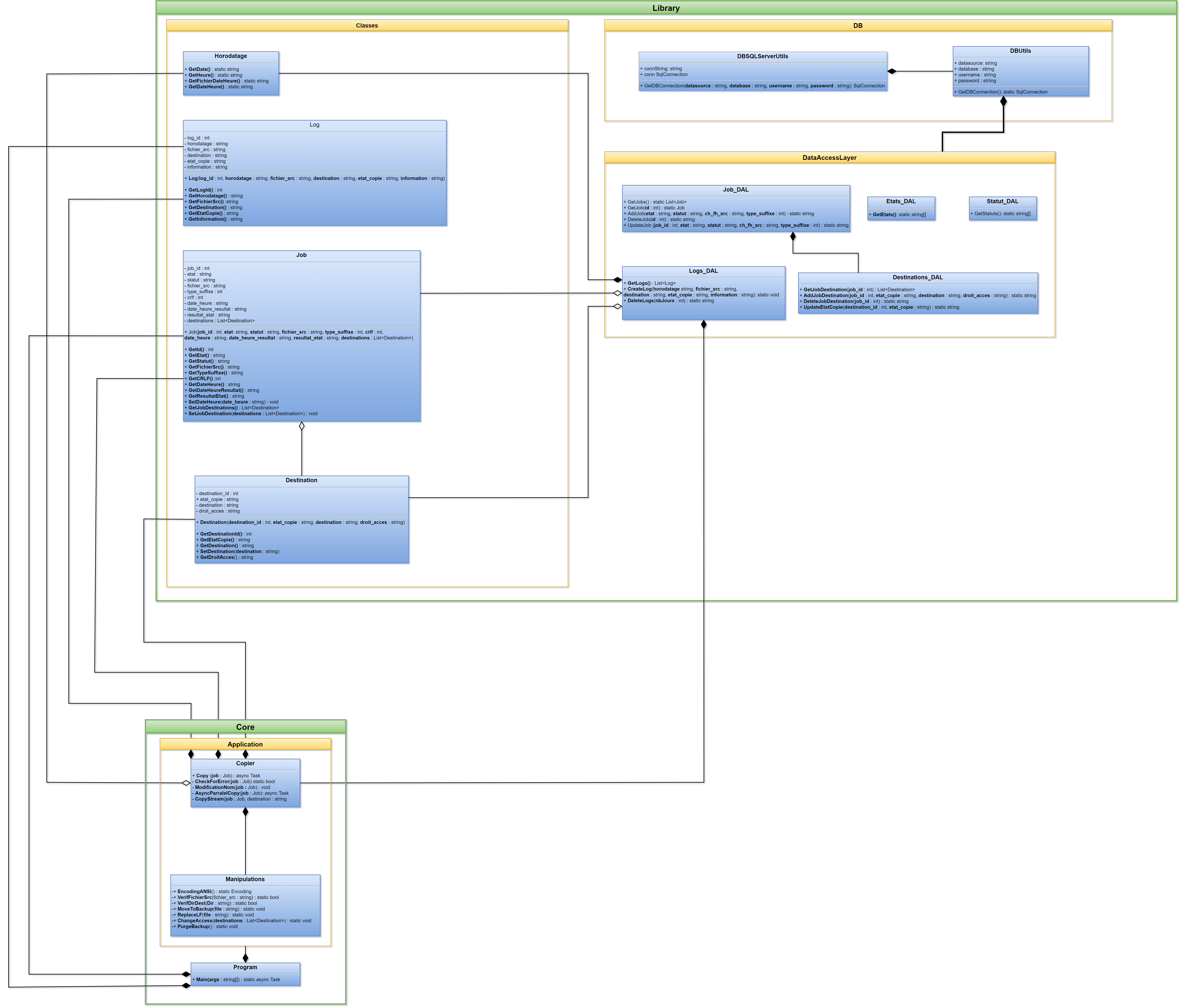
* **Table Log**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nom** | **Type de données** | **Description** |
| **PK** | **log\_id** | INT NOT NULL | Clé primaire |
|  | horodatage | DATETIME NOT NULL |  |
|  | fichier\_src | VARCHAR(150) | Fichier source du Job |
|  | destination | VARCHAR(150) | Destination de la copie |
|  | etat\_copie | VARCHAR(25) | Etat de la copie |
|  | information | TEXT | Message du log |

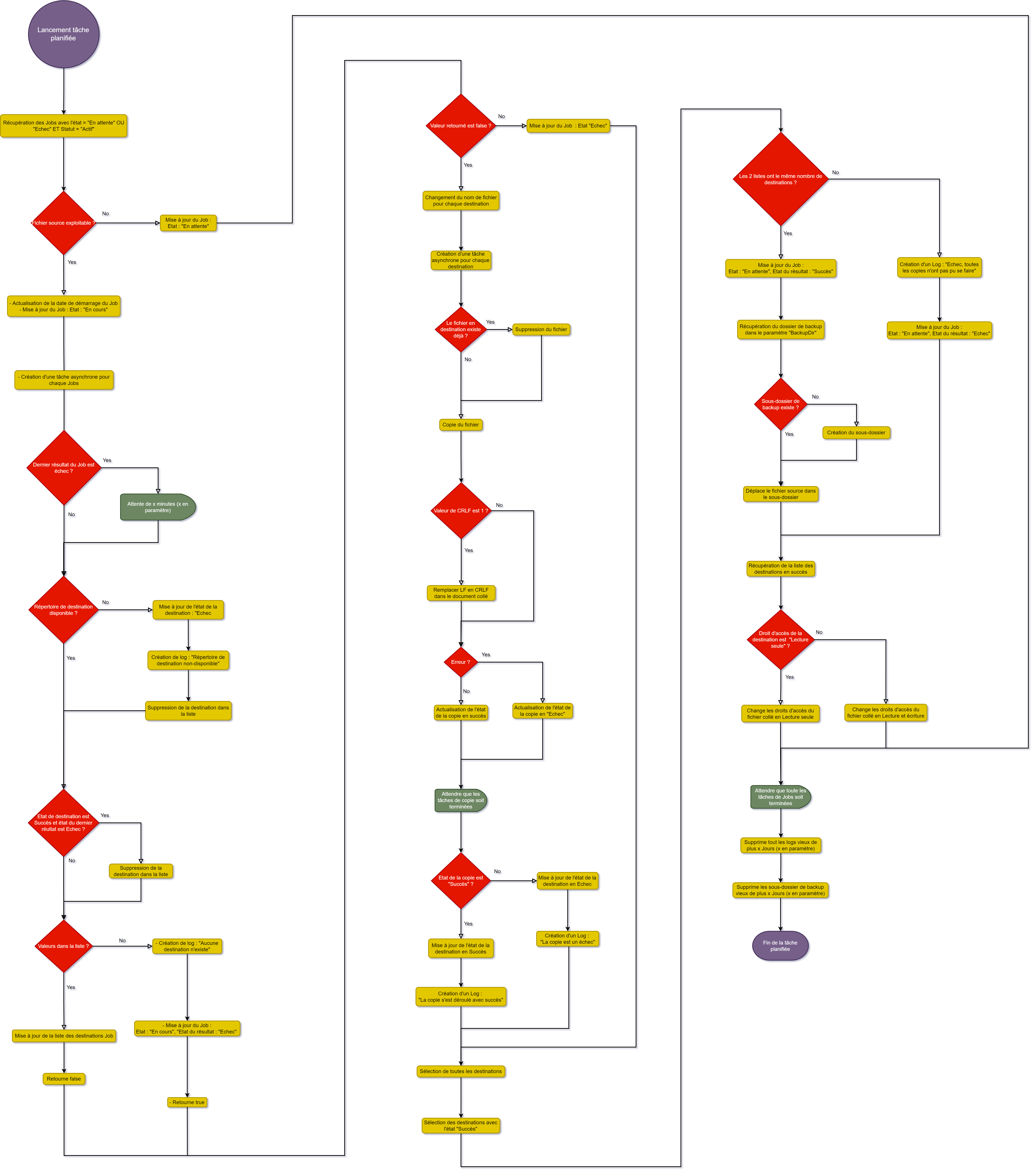
1. – Application

L’application est réalisée sur Visual Studio 2022 avec le Framework **.NET 5.0**

2.1- Diagramme de classe



2.2 - Diagramme de flux



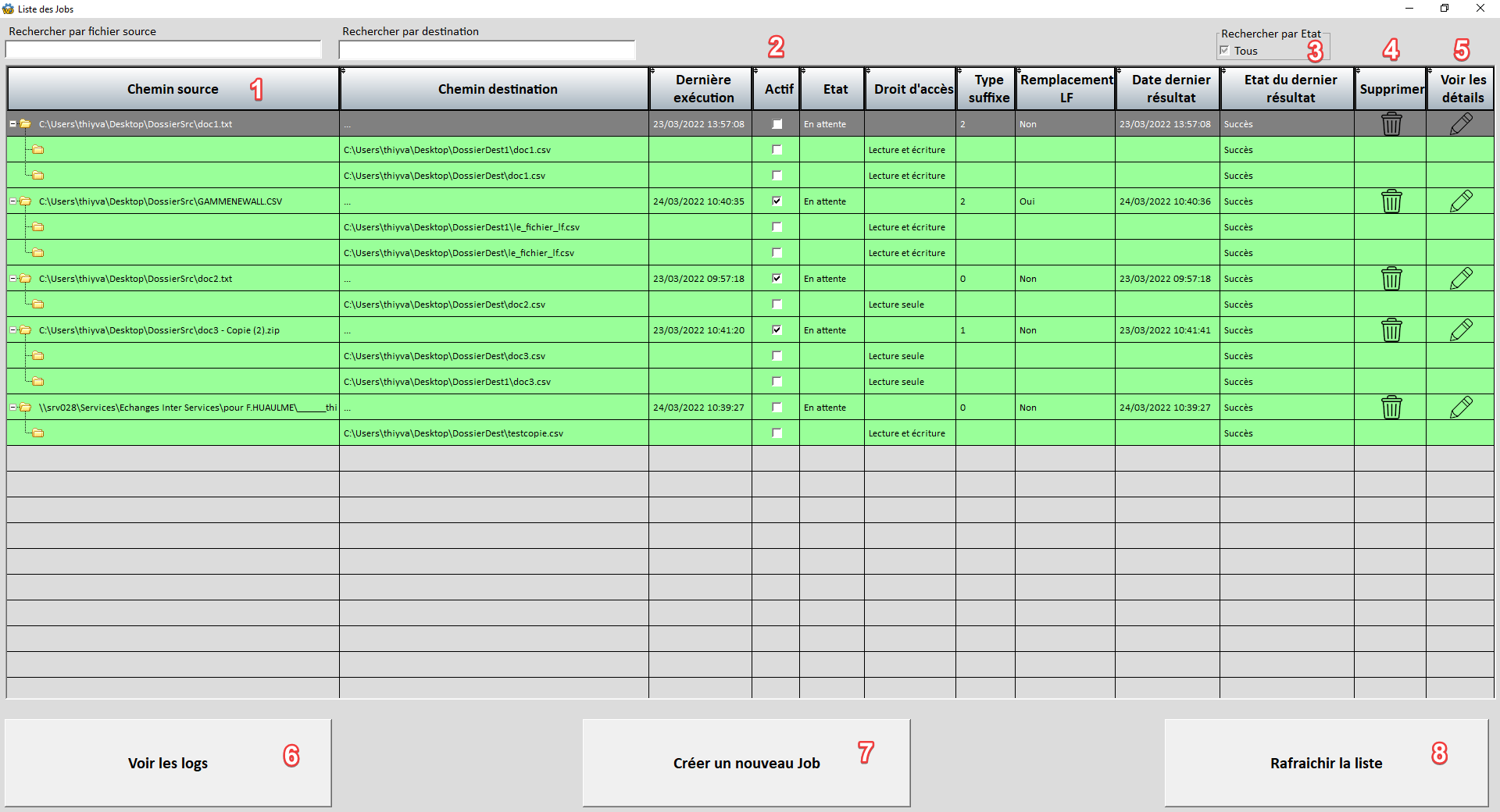
Limitations du programme :

Le fichier à copier ne doit pas faire plus de 2go soit la taille maximal d’un objet en C#

3- IHM

3.1 Fenêtre Liste des Jobs

Liste tous les Jobs de la base de données



1 – Chemins sources : Affiche les chemins sources et crée une ligne fils pour chaque destination. Une barre de recherche est disponible pour filtrer les Jobs en fonction de cette colonne.

2 – Colonne de l’activité des Jobs : cocher la case signifie que le Job est actif, sinon le job est inactif.

3 – Filtre en fonction de l’état en résultat. L’interrupteur a 3 états :

* Coché : Filtre en fonction des succès
* Décoché : Filtre en fonction des échecs
* Partiellement coché : Filtre en prenant en compte tous les états

4 – Colonne « Supprimer » : Supprime le Job sélectionné dans la liste et dans la base de données

5 – Colonne « Voir les détails » : Affiche la fenêtre Gestion Jobs avec les données du Job

6 – Bouton « Voir les Logs » Affiche la fenêtre Liste des logs

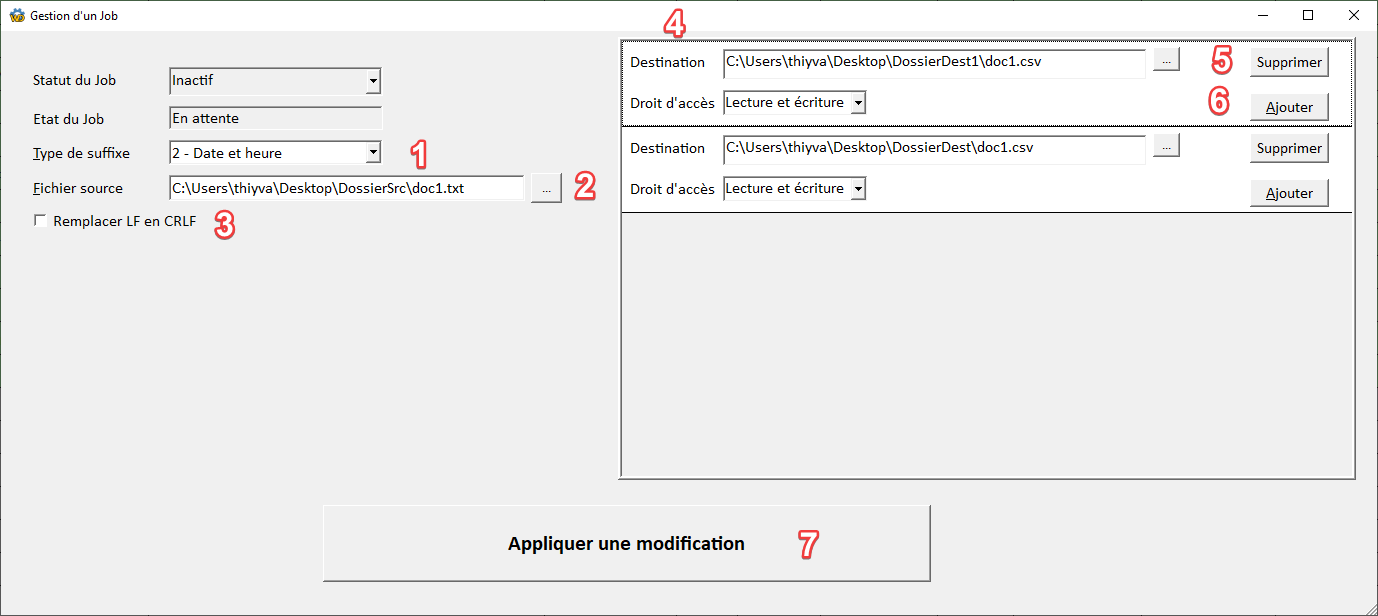
7 – Affiche la fenêtre Gestion Jobs pour créer un Job

8 – Bouton « Rafraichir la liste » Rafraichie la liste des Jobs

Autre : Une ligne se met en vert si c’est un succès ou en rouge si c’est un échec

3.2 Fenêtre Gestion Jobs

Cette fenêtre sert soit à créer ou modifier un Job.



1 – Liste Type de suffixe : 3 Choix possibles

* 0 – Ni la date ni l’heure : retourne la valeur 0 dans la base de données
* 1 – Seulement la date : retourne la valeur 1 dans la base de données
* 2 – Date et heure : retourne la valeur 2 dans la base de données

2 – Bouton Parcourir : Ouvre l’explorateur de fichier de Windows

3 – Interrupteur Remplacer LF en CRLF :

* Coché : Oui
* Décoché : Non

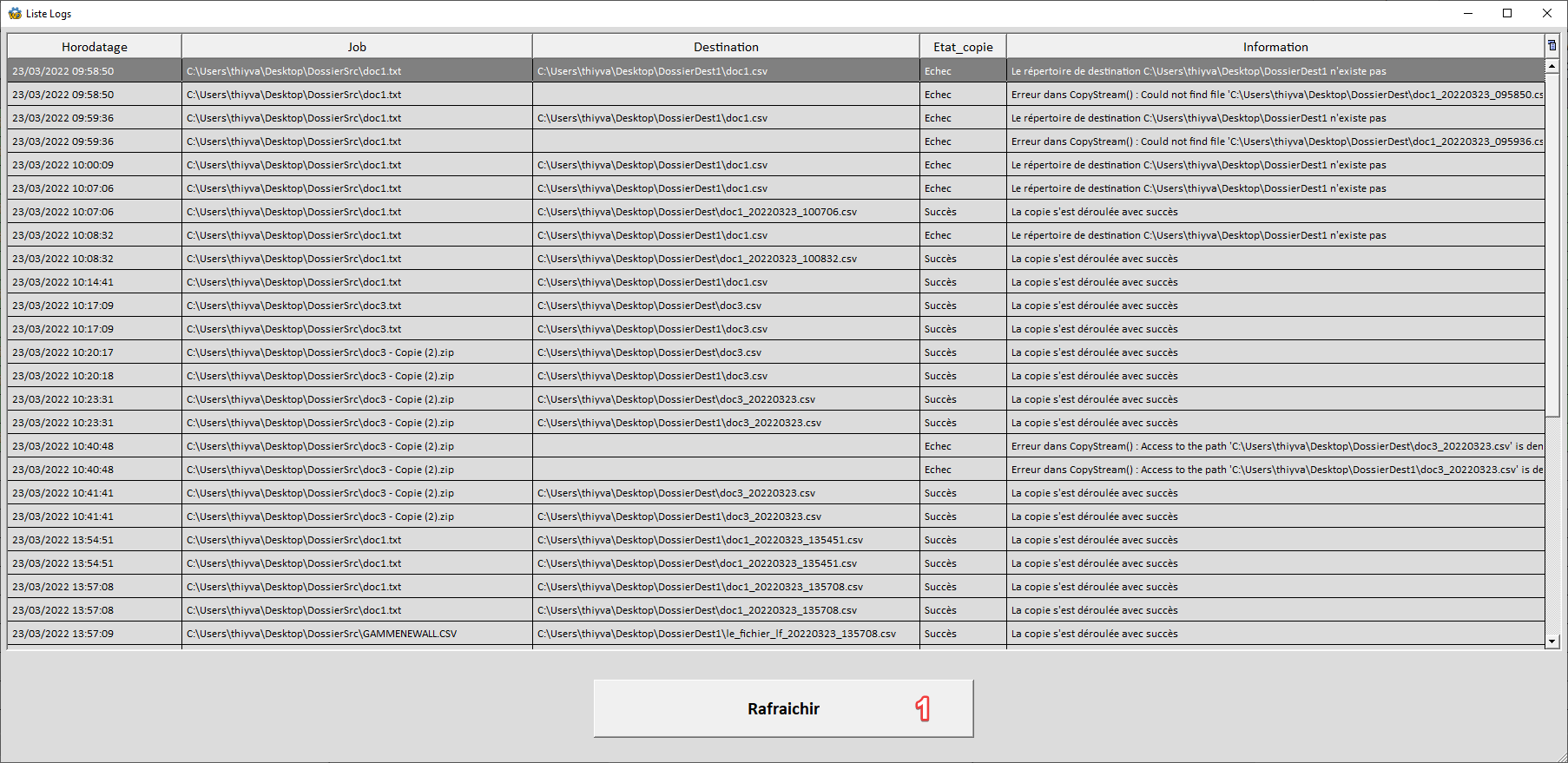
4 – Liste des destinations

5 – Bouton Supprimer : Supprime la destination sélectionné, ne fera rien s’il y en a qu’une dans la liste

6 – Bouton Ajouter : Ajoute une destination à remplir

7 – Bouton « Appliquer une modification » : Met à jour le Job dans la base de données ou crée en un nouveau. Ne fait rien si tous les champs ne sont pas remplis.

3.3 Fenêtre Logs

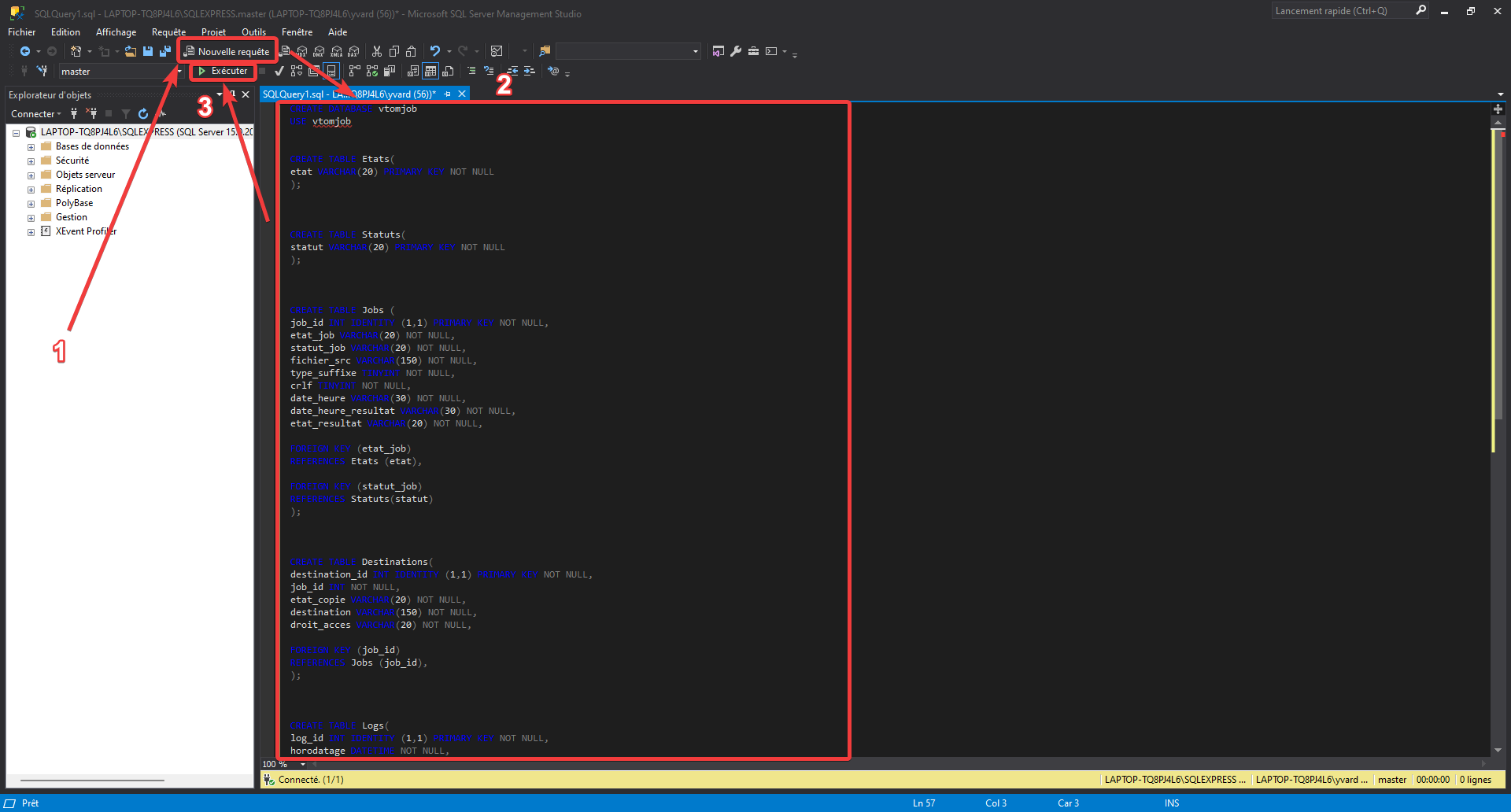
Liste tous les logs de la base de données

1 – Rafraichir la liste des logs

4– Mise en œuvre

4.1 – Base de données

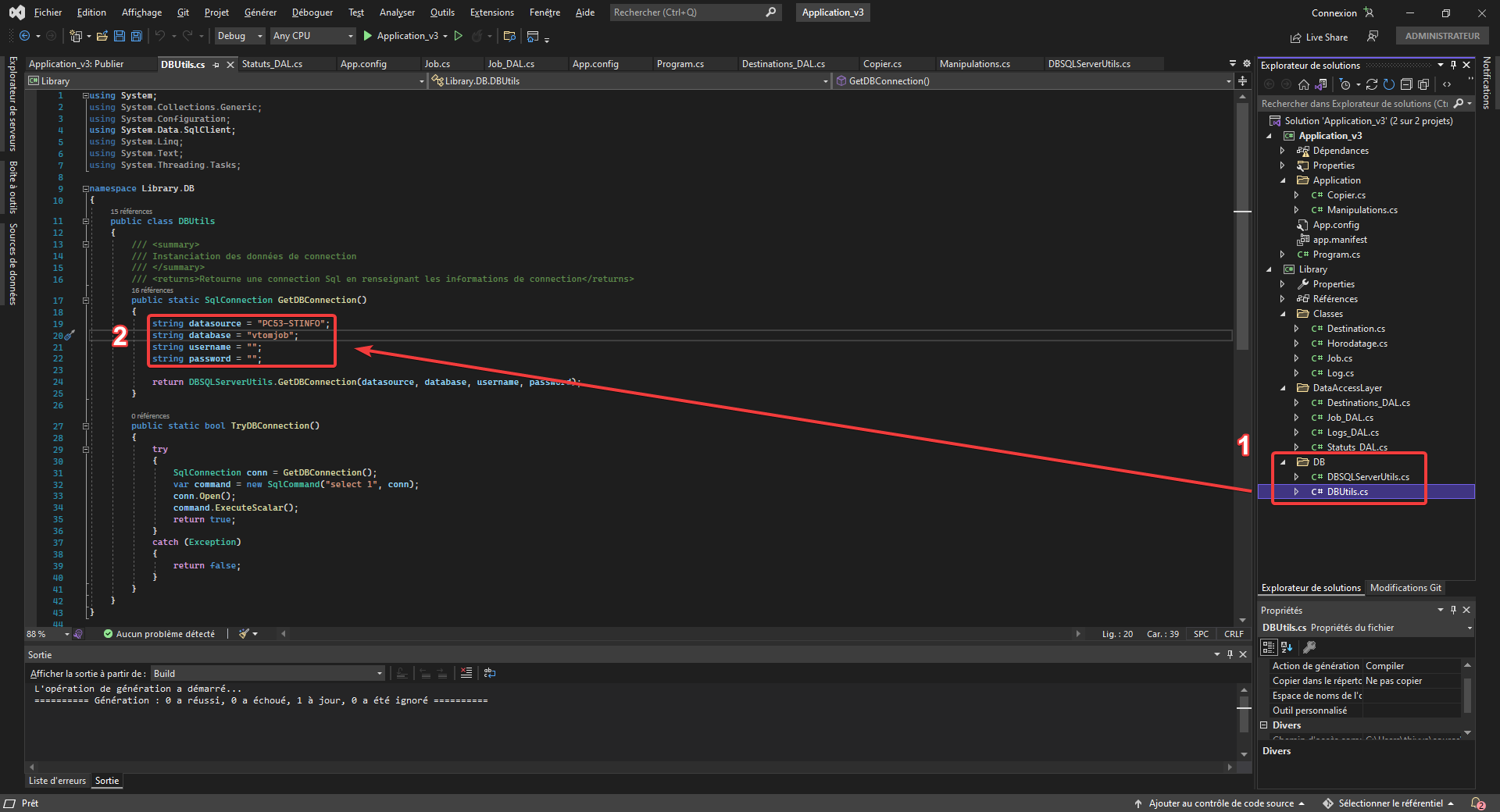
La base de données est sur SQL Server, il suffit de créer une nouvelle requête et entrer le script de création des tables puis l’exécuter.

NB - Il est possible qu’il faille en premier lieu créer la table, donc exécuter la première commande puis le reste.

4.2– Library

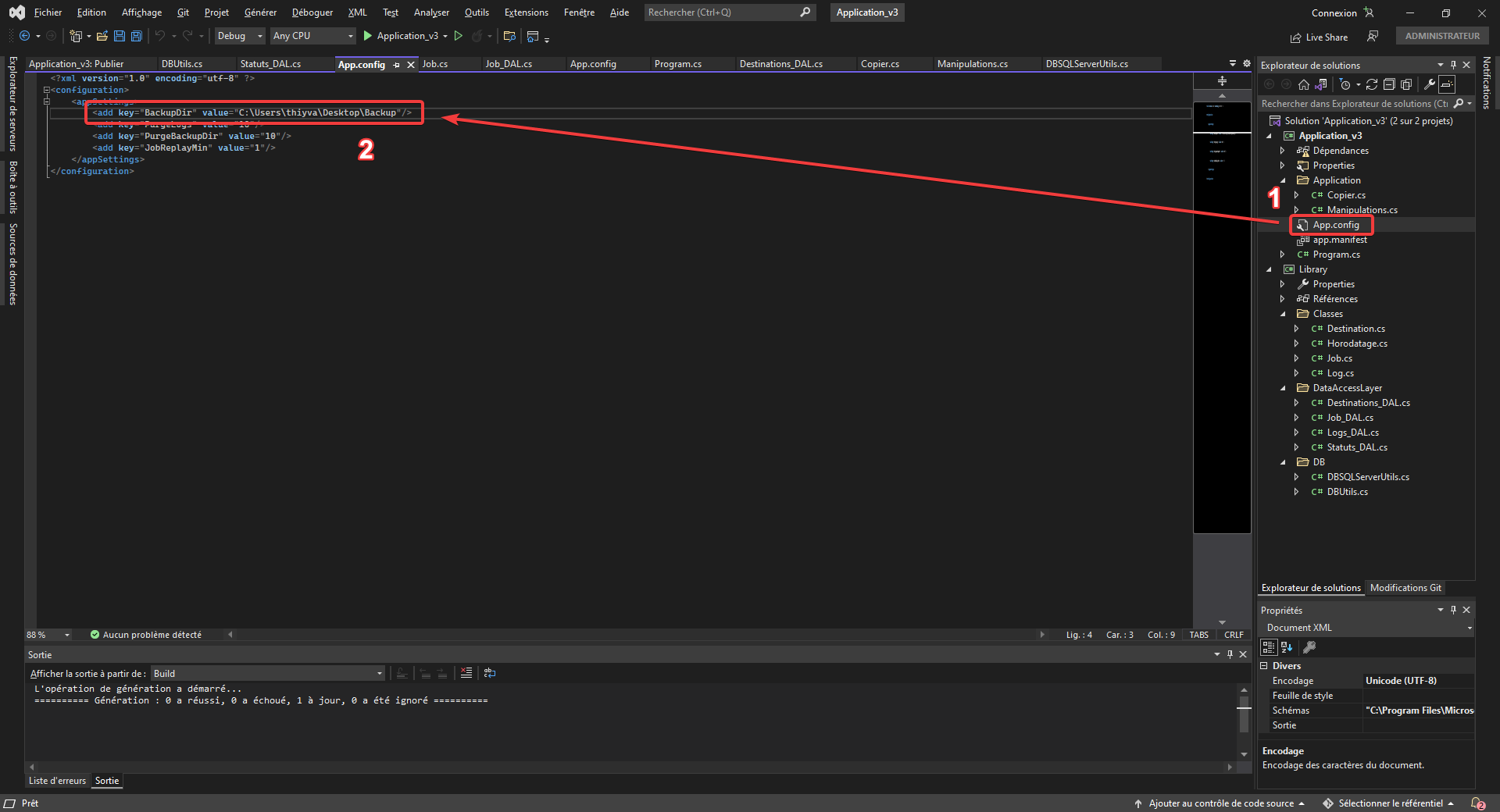
Il est nécessaire de changer la connexion à la base de données dans la Library, pour se faire, ouvrir la classe DBUtils et renseigner les champs.

* datasource : Nom de la source des données (Nom du poste)
* database : Nom de la base de données (vtomjob par défaut)
* username : Nom d’utilisateur (Laisser vide si l’authentification Windows est utilisé)
* password : Mot de passe (Laisser vide si l’authentification Windows est utilisé)



4.3- Application

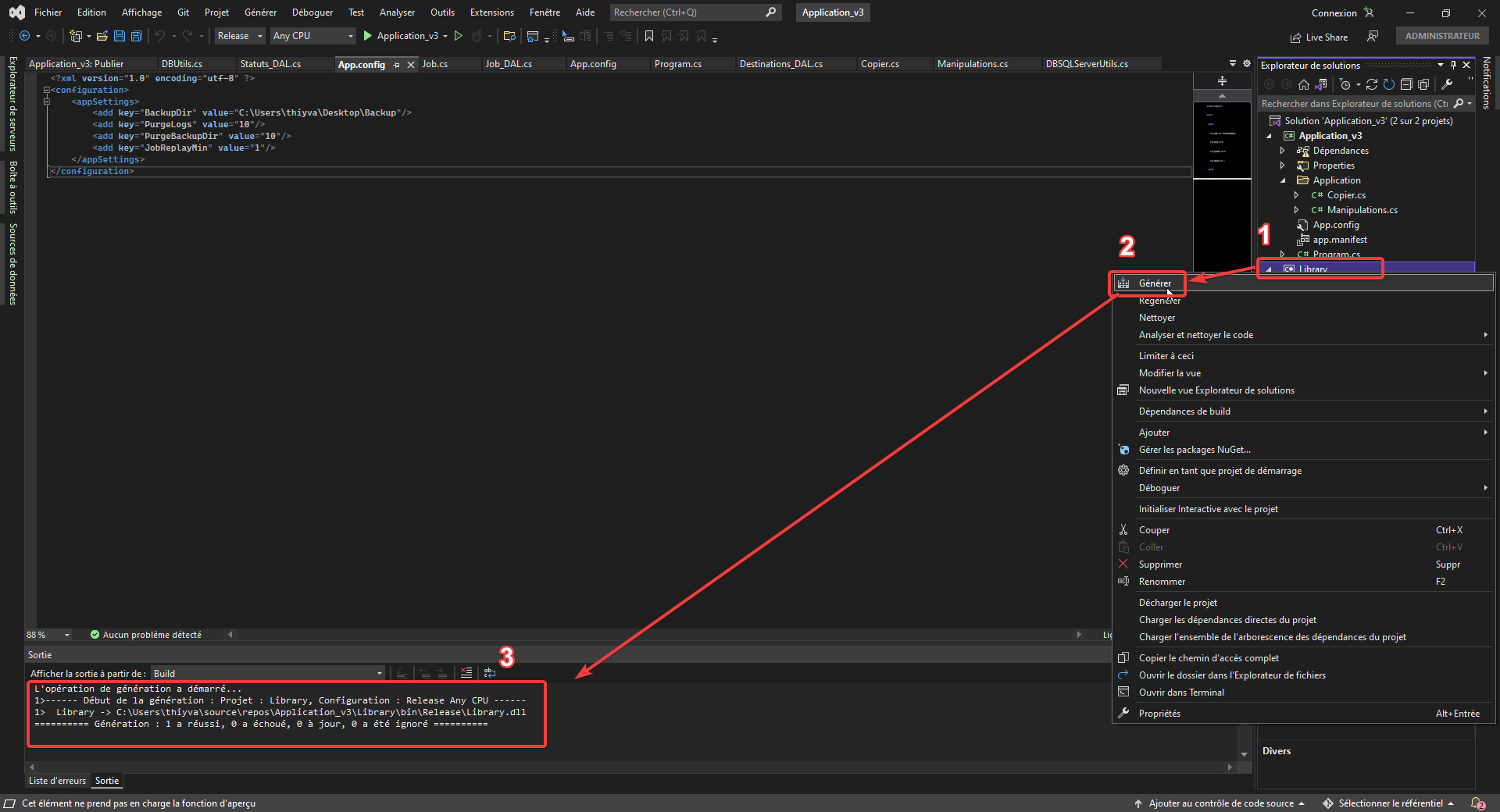
Renseigner le dossier de backup dans le fichier de configuration App.Config.



4.4-Actualisation

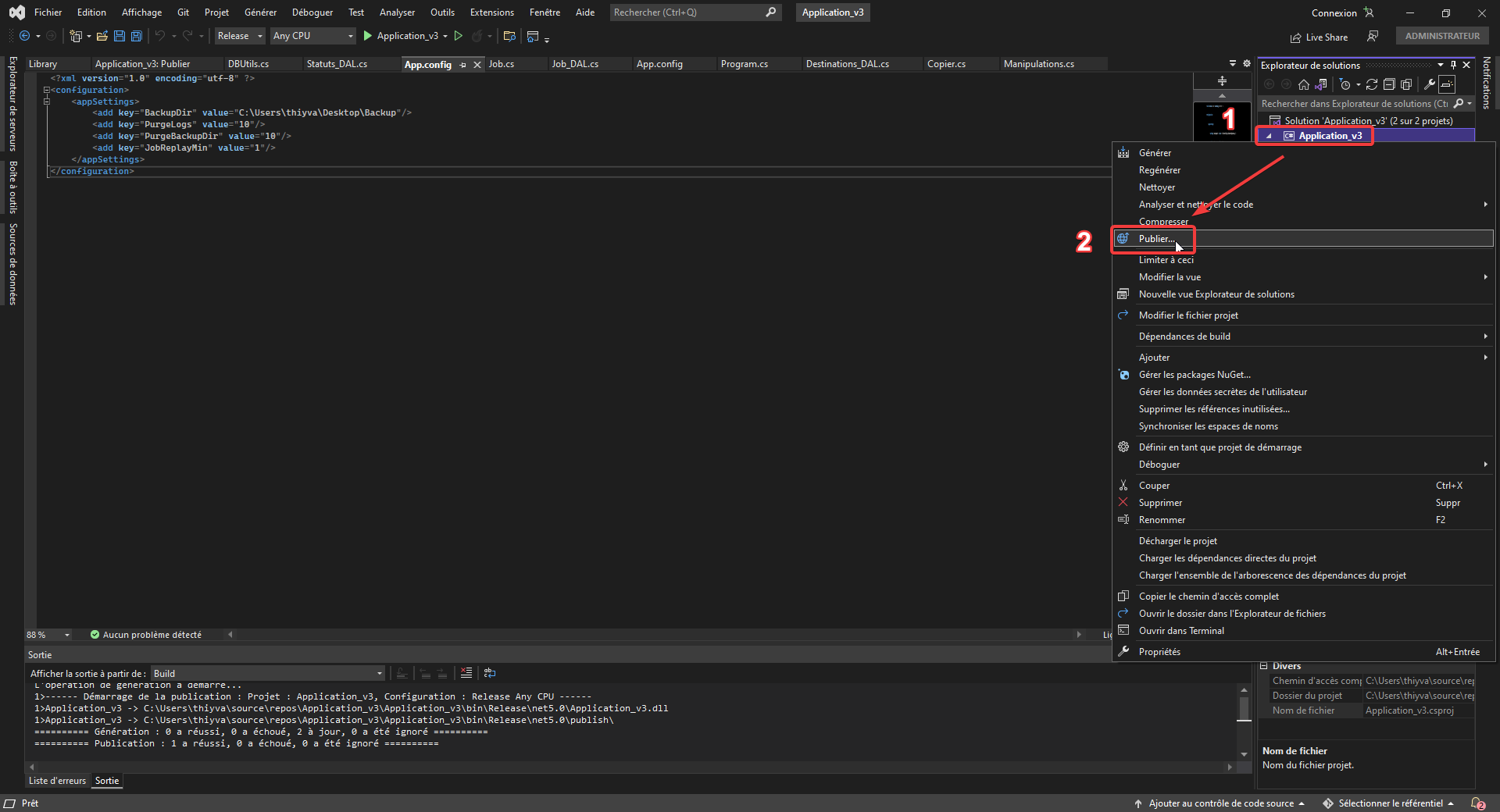
Afin que ces changements prennent effet, notamment la Library, il faut publier l’application.

En premier lieu, générer la nouvelle Library : Clic droit sur l’assembly Library puis Générer.

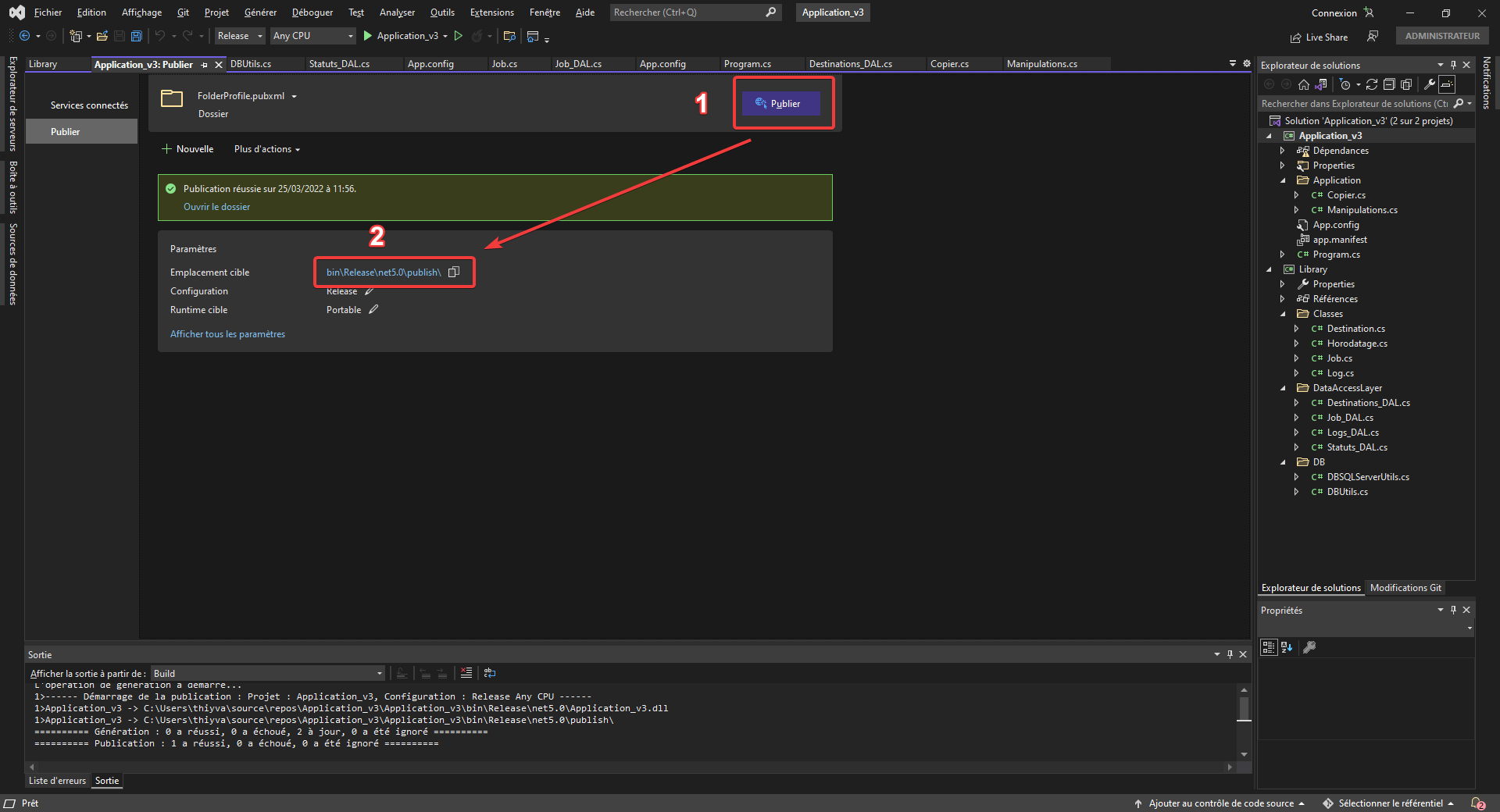


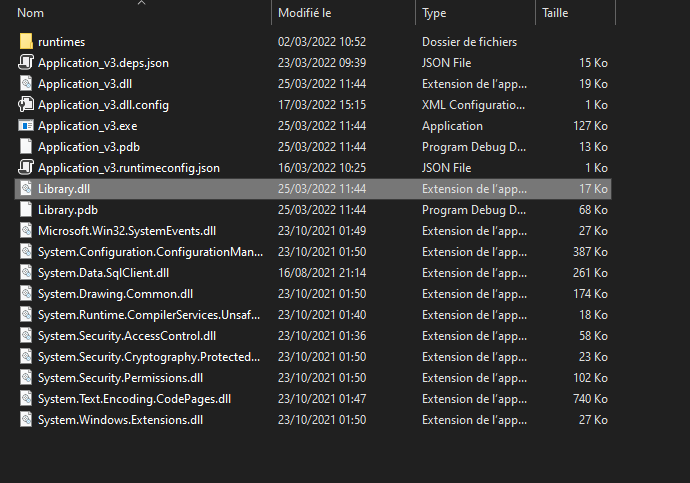
Vérifier que la génération a réussi en regardant la sortie.

Ensuite, il faut publier la solution

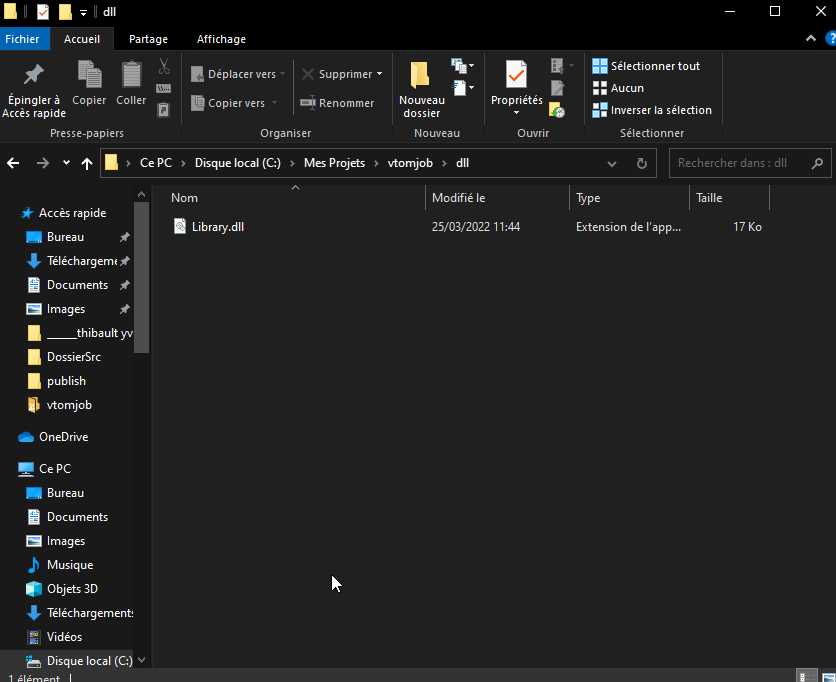


Arrivé sur la nouvelle fenêtre, cliquer sur Publier et aller dans le dossier de publication et récupérer le fichier Library.dll





Ouvrir le dossier du projet vtomjob sur WINDEV et remplacer la dll présente dans le dossier dll par la nouvelle



Ouvrir le projet vtomjob sur WINDEV, dans Explorateur de projet, Assemblage .NET puis Recharger les assemblages .NET

