

# Fuzible – Documentation générale

AUTEUR	Guillaume Tristant
CREATION	20171002
REVISION	20221227
RESSOURCES	<a href="http://www.fuzible-app.com">www.fuzible-app.com</a>
SUPPORT MAIL	guillaume@fuzible-app.com
FORUM	<a href="http://www.fuzible-app.com/wordpress/fuzibleForum/phpBB3/">www.fuzible-app.com/wordpress/fuzibleForum/phpBB3/</a>
LANGUE	FR / EN

## Table des matières

Préambule .....	5
Information à propos de la documentation.....	5
Principe .....	6
Installation .....	6
Configuration recommandée.....	6
Langue.....	6
Installation en Entreprise.....	6
Installation pour un Particulier .....	6
Architecture logicielle .....	7
Enregistrement du programme .....	8
Périmètre de License .....	9
Changement de License (upgrade) .....	9
Compatibilité du programme.....	10
Scénarios Supportés.....	10
Sécurité .....	10
Modes de fonctionnement, démarrage.....	11
Démarrer le programme en mode « UI » .....	11
Démarrer le programme en mode « CONSOLE » (exécution silencieuse d'un Job) .....	11
Premier lancement .....	11
Répertoire de fonctionnement.....	11
Assistance intégrée au logiciel .....	12
Jobs de démonstration .....	12
Boutons d'Aide Rapide.....	13
Tutoriaux .....	13
Assistant au requêtage .....	14
Aide en ligne.....	14
Configuration du programme .....	15
Connexions.....	15
Base de données .....	17

MongoDB .....	19
Fichiers .....	20
Webservice REST .....	21
Mail .....	23
Active Directory.....	24
Log.....	25
SHS Analyzer .....	26
SQL .....	27
File.....	28
Mail .....	29
Webservices .....	30
Service/Client App.....	31
Dev. ....	32
Outils .....	34
Export Job (XML) .....	34
Import Job (XML).....	34
Reorganize Jobs.....	34
Move as Sub Job.....	35
Move as Main Job .....	35
Move Down/Up.....	35
Import External Job Parameters .....	35
Load another Userspace (read-only) .....	36
Création de Jobs.....	37
Onglet Job Configuration .....	37
Job Type .....	38
Script Langage .....	40
Résumé du Job .....	41
Planifications.....	41
Calendrier des planifications.....	43
Onglet Source.....	44
BDD .....	45
MONGODB .....	45
Fichier CSV.....	45
Fichier Excel.....	46
Fichier XML et JSON .....	46
Webservice REST.....	47
Mailbox .....	47
Active Directory.....	48

Data Analyzer .....	48
Data Transformation.....	49
Pre and Post-Job Commands .....	52
Boucles des commandes Pré-Job.....	53
Onglet Cible.....	55
Options communes à toutes les cibles : .....	55
BDD .....	56
Fichier CSV.....	57
Fichier Excel.....	57
Fichier XML.....	57
Fichier JSON .....	58
Webservice REST.....	61
Mailbox .....	62
Active Directory.....	62
Onglet Queries .....	63
Output.....	64
Fonctionnement du mapping de champs .....	64
Cas particulier des requêtes de synchronisation :.....	65
SELECT – A partir d’une base de données, vers une base de données.....	67
SELECT – A partir d’une base de données, vers un fichier .....	67
SELECT – A partir d’une base de données, vers une adresse mail.....	67
SELECT – A partir d’un fichier.....	68
SELECT – A partir d’un webservice en mode Fuzible SQL (A) .....	68
SELECT – A partir d’un webservice en mode Fuzible SQL (B).....	69
SELECT – A partir de l’API Salesforce en mode SoQL.....	69
SELECT – A partir d’une boîte mail.....	70
SELECT – A partir de l’Active Directory .....	70
SELECT Multi-tables, multi-fichiers .....	73
Requêtes Multi-Cible.....	74
Requêtes Croisées.....	75
Menu Contextuel des requêtes .....	83
QUERY ANALYZER - Source Infos.....	84
QUERY ANALYZER - Target Infos .....	85
QUERY ANALYZER - View Data .....	86
QUERY ANALYZER - Query Details.....	87
EXECUTE QUERY - Run this individual Query.....	89
SYNCHRO-QUERY – Transcoded for Target.....	89
SYNCHRO-QUERY – Validity check for Synchro Query.....	89

SCRIPTING - Get Full Header Query and copy/paste it .....	89
SCRIPTING – Transformations.....	90
SCRIPTING – Add a Dynamic Parameter .....	91
SCRIPTING – Basic Query Builder .....	92
ADVANCED QUERY SCRIPTING – Add Cross-Connections Join .....	92
ADVANCED QUERY SCRIPTING – Create Dual Target.....	92
Onglet Log Viewer .....	93
Exécution d'un Job .....	94
Application « Service ».....	96
Configuration du planificateur de tâches Windows .....	96
Configuration d'un Job externe (hors Fuzible) avec l'application « Service » .....	98
Application « Client » .....	99
Fuzible SQL : Lexique.....	103
Fonctions supportées.....	103
Fonctions courantes non supportées .....	103

## Préambule

Avant tout, je souhaite vous remercier chaleureusement d'utiliser (ou d'essayer) Fuzible. Ce logiciel est l'aboutissement de plusieurs années de travail, et du constat que beaucoup de tâches liées à la manipulation de données sont trop souvent chronophages et redondantes.

Il existe de nombreuses alternatives, et l'idée n'est pas de créer une concurrence, mais une autre façon de concevoir les échanges et la manipulation de données.

Le programme s'adresse aux développeurs ayant une connaissance minimale du langage SQL.

**La philosophie du programme est simple :**

**Toute Source de données est une base de données et se requête comme tel !**

Quelques cas d'usage :

- Réplication de données (copie)
- Synchronisation de données (comparaison intelligente de 2 sources)
- Interfaces (ex : récupérer les données du webservice d'un logiciel en SaaS et les envoyer dans un autre logiciel sous forme de fichiers XML)
- Migration de données d'une BDD à une autre (peu importe qu'elles soient différentes)
- Extraction rapide de données (ex : SQL vers CSV)
- Import rapide de données (ex : EXCEL vers BDD)
- Comparaison de données (ex : pour contrôler l'intégrité de 2 BDD)
- Croisement de données provenant de différentes sources en temps réel
- Remplissage d'un entrepôt de données (ex : l'intégration d'un fichier dans une BDD est totalement automatisé : de l'analyse des données à la création de table)
- Synchronisation d'un environnement de pré-production avec un environnement de production

## Information à propos de la documentation

La documentation a été rédigée en Français, mais les captures d'écran s'appuient sur l'interface du logiciel définie en anglais. Le travail est trop chronophage pour que je fasse 2 versions pour chaque capture, vous verrez donc les noms des options en anglais, mais l'explication en Français. Le comportement de l'interface utilisateur est rigoureusement le même d'une version à l'autre.

Merci de votre compréhension !

## Principe

Fuzible est un outil qui permet d'importer, exporter, mixer, synchroniser, répliquer, comparer des données. Le principe général s'appuie sur la définition d'une Source de données (que l'on va requêter), et d'une Cible (dans laquelle on va copier des données).

L'outil est capable de travailler avec plusieurs SGBD (voir tableau de compatibilité), aussi bien pour l'export que pour l'import des données. Il offre une possibilité d'analyse de l'ensemble des données requêtées pour créer des champs d'import intelligents (types et longueurs cohérents). Mais au-delà de ça, il peut lire et écrire dans des fichiers, webservices, boîtes mail, et dans l'Active Directory !

Pour finir, il est conçu comme modulaire, du traitement des données à la gestion du LOG et de l'orchestration de Jobs. Un couteau-suisse, en somme. Il peut répondre aux besoins les plus élémentaires (la copie d'un fichier CSV dans une base de données) aux plus complexes (la synchronisation de 2 environnements de bases de données).

## Installation

Le programme peut s'installer n'importe où sur le disque dur. Vous pouvez au choix utiliser la version portable ou la version avec installateur. Dans les 2 cas, aucune donnée ne sera écrite ailleurs que dans le répertoire d'installation.

## Configuration recommandée

Cette évaluation est basée sur mes tests. A noter que le multithreading augmente de manière exponentielle les besoins de la machine faisant tourner le programme. Configuration indiquée pour 4 threads.

- OS Microsoft Windows (à partir de Windows 7, ou Server 2008r2 et plus)
- CPU Intel I5 de 4ème generation avec au minimum 2 threads d'exécution
- 8 GO de RAM (grand minimum)
- HDD 7200tr/min avec ~ 10Go disponibles.
- Accès Internet
- Microsoft NetCore Framework v3.1 (<https://dotnet.microsoft.com/download>)

## Langue

Le programme est disponible en 2 langues (Français, Anglais). Par défaut, il est configuré pour utiliser la langue système mais vous pouvez basculer d'une langue à l'autre dans le menu « Fichier ».

## Installation en Entreprise

Dans un environnement d'entreprise, Fuzible étant multi-utilisateur (et délivre toutes ses possibilités lorsqu'il est utilisé dans ce sens), il est conseillé de l'installer sur un serveur accessible en RDP. Par ailleurs, un poste serveur aura souvent bien plus de droits d'accès vers des instances SQL, chemins réseau, FTP... qu'un poste client.

De la même manière, si vous faites des répliquions d'une BDD à une autre, et que Fuzible est installé sur un poste client, les temps de traitement pourraient s'avérer horriblement lents car les données devront être d'abord récupérées du serveur Source vers le poste Client, puis de celui-ci vers le serveur Cible. Si vous êtes derrière un VPN, ou que votre connexion à l'entreprise est lente, les performances seront extrêmement dégradées.

## Installation pour un Particulier

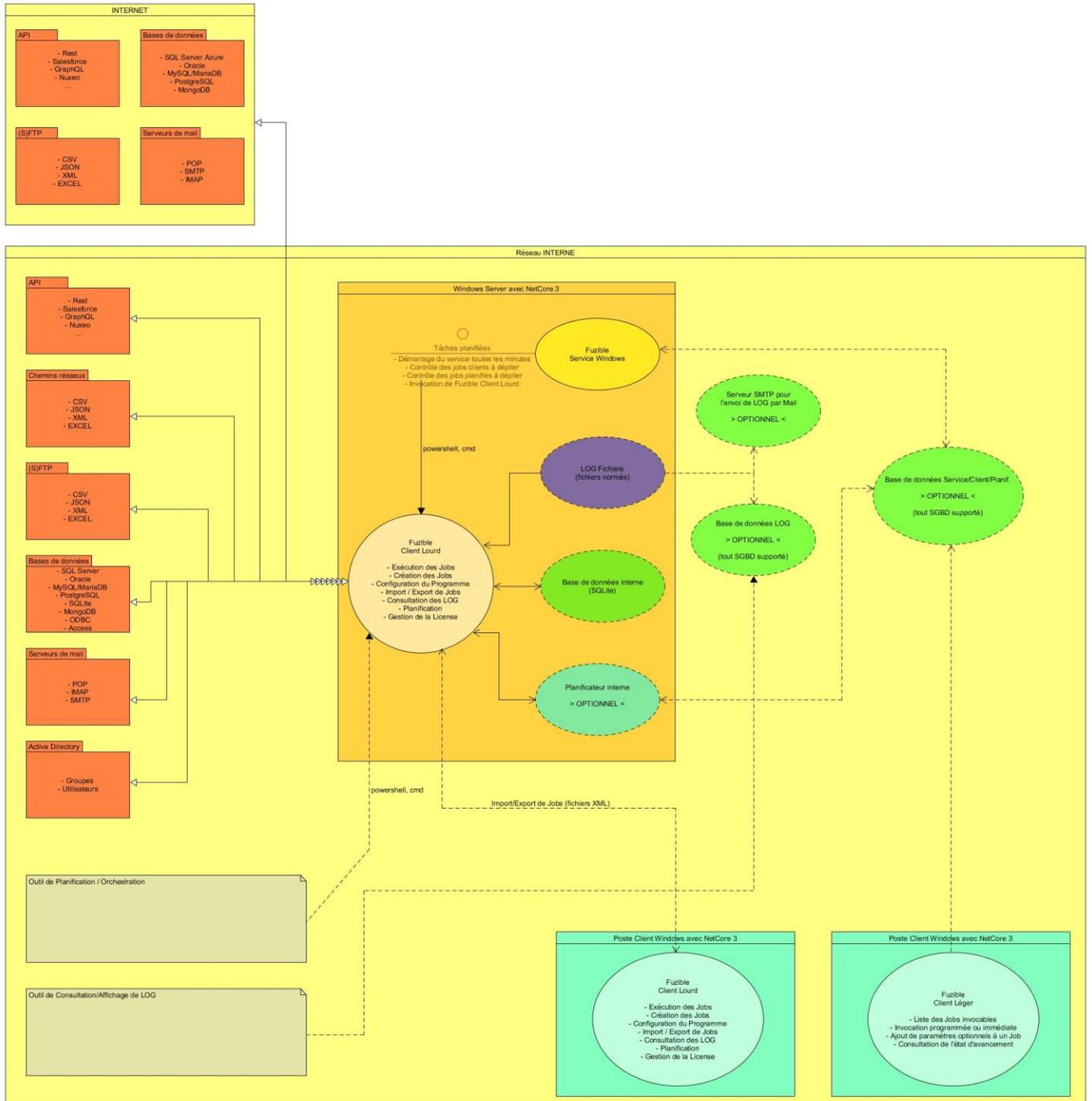
Assurez-vous simplement que l'ordinateur sur lequel vous installez l'application accède à toutes les sources de données que vous allez manipuler (instances SQL, réseau local, domaine Active Directory, connexion Internet...) et que le compte utilisateur est « administrateur » du Poste.

# Architecture logicielle

Le diagramme présente les différentes briques de l'application. Certaines sont optionnelles et ne nécessitent donc pas d'être configurées.

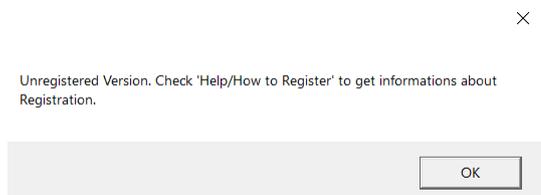
L'application principale se trouve dans le rectangle orange.

Toutes les lignes pointillées représentent les briques optionnelles. En vert, celles qui sont gérées et incluses avec Fuzible, en gris, les outils externes.



## Enregistrement du programme

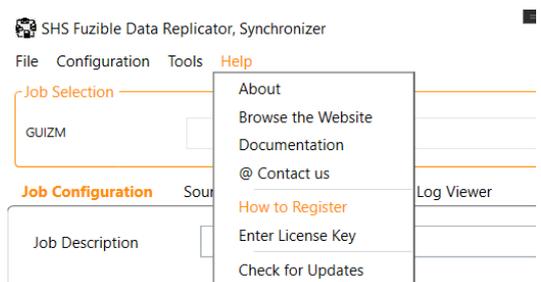
Au démarrage du programme, que celui-ci ait été acheté ou non, il démarrera en mode de démonstration. Il est limité dans son usage et un message vous informera à chaque lancement.



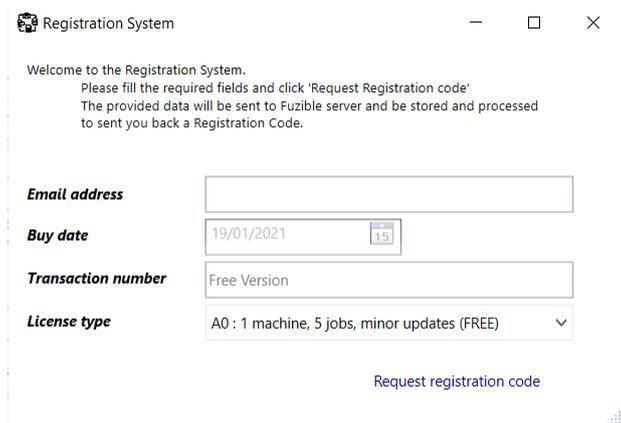
Il vous sera impossible de faire les actions suivantes :

- Orchestrer des Jobs
- Importer et exporter des Jobs
- Utiliser l'application Client & Service
- Créer plus de 3 Jobs

Pour enregistrer le programme, il vous suffit de vous rendre dans le menu d'Aide :



Un menu s'ouvre et vous propose alors de saisir votre adresse mail afin de vous enregistrer.



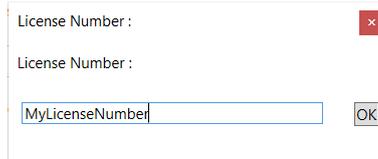
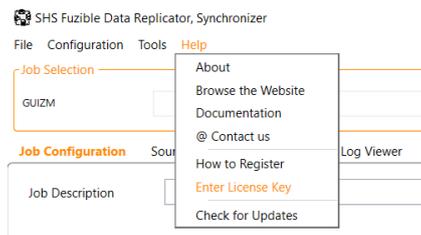
Si vous avez acheté le programme, vous devrez choisir en plus le type de License que vous avez achetée dans la liste déroulante, et spécifier alors en plus le numéro de transaction qui vous a été fourni lors de l'achat, ainsi que la date à laquelle la License a été achetée.

*Note : Vos informations sont uniquement stockées sur le serveur de Fuzible, les données ne sont transmises en aucun cas à qui que ce soit. Le fait d'enregistrer la version gratuite permet de mesurer la quantité d'utilisateurs actifs sur l'application.*

*Les données envoyées sont les suivantes : adresse mail, modèle de CPU, quantité de RAM, version de Windows, version du Framework NetCore, identifiants uniques du PC (disque dur principal, CPU), quantité de Jobs dans Fuzible, quantité de connexions, l'adresse IP locale, l'adresse IP publique, le chemin d'installation du programme, la version du programme.*

Cliquez sur « Request Registration Code » pour envoyer vos données au serveur Fuzible. Votre demande d'enregistrement sera traitée manuellement sous quelques minutes, et un code d'enregistrement vous sera alors envoyé en retour par mail.

Celui-ci devra être saisi dans le menu d'Aide :



L'authenticité du code est contrôlée localement, et un message vous informera de la validité de celui-ci.

**La version gratuite libère toutes les fonctionnalités de l'application, mais le nombre de Jobs qu'il est possible de créer est limité à 5. Les versions payantes s'articulent autour de la quantité de Jobs qu'il est possible de créer, ainsi que du type de mises à jour qu'il sera possible de télécharger dans l'avenir (MAJ correctives, évolutions mineurs, majeures).**

## Périmètre de License

Une License Fuzible est multi-utilisateur (plusieurs utilisateurs sur le même ordinateur peuvent utiliser l'application de manière indépendante) mais se limite à l'ordinateur sur lequel l'application est installée. Les changements suivants provoqueront un retour du programme en mode de démonstration :

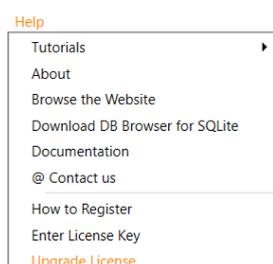
- Changement du répertoire d'installation
- Changement de CPU
- Changement de disque dur
- Modifications manuelles dans la base de données interne du programme

Si un changement de situation s'avère légitime, vous pouvez demander de l'assistance sur le site web :

[www.fuzible-app.com](http://www.fuzible-app.com)

## Changement de License (upgrade)

Si vous avez déjà une License, et que vous venez d'acquérir une License supérieure à partir de notre site web, vous pouvez vous rendre dans le menu ci-dessous pour la mettre à jour. Le processus sera le même qu'à votre premier enregistrement.



## Compatibilité du programme

SGBD	Support
SQL Server	SOURCE : Support total CIBLE : Support total
MySQL	SOURCE : Support total CIBLE : Support total
Postgres SQL	SOURCE : Support total CIBLE : Support total
ODBC	SOURCE : Support total CIBLE : Support partiel : paramétrage des requêtes système à réaliser dans le fichier de connexion
Oracle	SOURCE : Support total CIBLE : Support total
SQLite	SOURCE : Support total CIBLE : Support total
Access	SOURCE : Support total CIBLE : Support total
MongoDB	SOURCE : Support total CIBLE : Support total
FICHER	SOURCE : Support XML, JSON, CSV, XLS, XLSX (+S)FTP CIBLE : Support XML, JSON, CSV, XLS, XLSX (+S)FTP
WEBSERVICES	SOURCE : REST API CIBLE : Support partiel (REST, http)
MAILBOX	SOURCE : POP, IMAP CIBLE : SMTP
ACTIVE DIRECTORY	SOURCE : users, groups CIBLE : users, groups

## Scénarios Supportés

SOURCE	CIBLE	REPLICATION	SYNCHRONISATION
Base de données	Base de données	OUI	OUI
Base de données	Fichier	OUI	OUI
Base de données	Webservice	OUI	NON
Base de données	Mailbox	OUI	NON
Base de données	Active Directory	OUI	OUI
Fichier	Base de données	OUI	OUI
Fichier	Fichier	OUI	OUI
Fichier	Webservice	OUI	NON
Fichier	Mailbox	OUI	NON
Fichier	Active Directory	OUI	OUI
Webservice	Base de données	OUI	OUI
Webservice	Fichier	OUI	OUI
Webservice	Webservice	OUI	NON
Webservice	Mailbox	OUI	NON
Webservice	Active Directory	OUI	OUI
Mailbox	Base de données	OUI	OUI
Mailbox	Fichier	OUI	OUI
Mailbox	Webservice	OUI	NON
Mailbox	Mailbox	OUI	NON
Mailbox	Active Directory	OUI	OUI
Active Directory	Base de données	OUI	OUI
Active Directory	Fichier	OUI	OUI
Active Directory	Webservice	OUI	NON
Active Directory	Mailbox	OUI	NON
Active Directory	Active Directory	OUI	OUI

## Sécurité

Le programme utilise une base de données SQLite pour fonctionner. Un certain nombre d'informations y sont chiffrées (AES), notamment les chaînes de connexion et les mots de passe afin de protéger la confidentialité des données pour chaque session utilisateur.

## Modes de fonctionnement, démarrage

### Démarrer le programme en mode « UI »

Exécuter **Fuzible.exe**

### Démarrer le programme en mode « CONSOLE » (exécution silencieuse d'un Job)

Exécuter **Fuzible.exe** avec des arguments

1 - Nom de l'Userspace (basiquement, l'utilisateur connecté)

2- ID du Job à exécuter au démarrage (si besoin)

3 - Mot de passe (en chiffré)

4 - Arguments dynamiques (voir section « Script Language »)

*Exemple :*

**Fuzible.exe «GUILLAUME» «[10]» «Apza==7824&éhfjflzefze=»**

- Va lancer le job N°10 de l'utilisateur « GUILLAUME »

Compte-tenu de l'austérité de la saisie des arguments, le programme propose, dans son UI, de vous afficher le code « lancer » de chaque Job afin que vous puissiez le copier-coller directement (dans un outil d'ordonnancement par exemple).

Le programme possède donc en résumé 2 modes de fonctionnement :

- En mode paramétré (nom du job à exécuter), il ne charge pas l'interface graphique et réalise le Job en tâche de fond
- En mode non paramétré, il ouvre l'interface graphique : celle-ci permet de modifier, ajouter et lancer des Jobs

### Premier lancement

Au premier lancement du programme, celui-ci va créer :

- Votre configuration générale par défaut
- Deux Jobs de démo contenant une vingtaine de requêtes de test, pour vous présenter les fonctionnalités
- 3 connexions « fichier » locales, 1 connexion SQLite, 1 connexion vers un webservice de démo

Vous serez également notifié que vous n'êtes pas enregistré, et qu'en l'état, vous ne pourrez pas créer plus de 3 Jobs. En outre, certaines fonctionnalités seront désactivées.

### Répertoire de fonctionnement

Fuzible stocke ses données ici : **C:\Users\Public\Documents\Fuzible.**

On y retrouve la base de données interne du programme, les fichiers de LOG, les binaires de l'application Client ainsi que les fichiers de démonstration utilisés par les connexions « fichier » créés par défaut.

## Assistance intégrée au logiciel

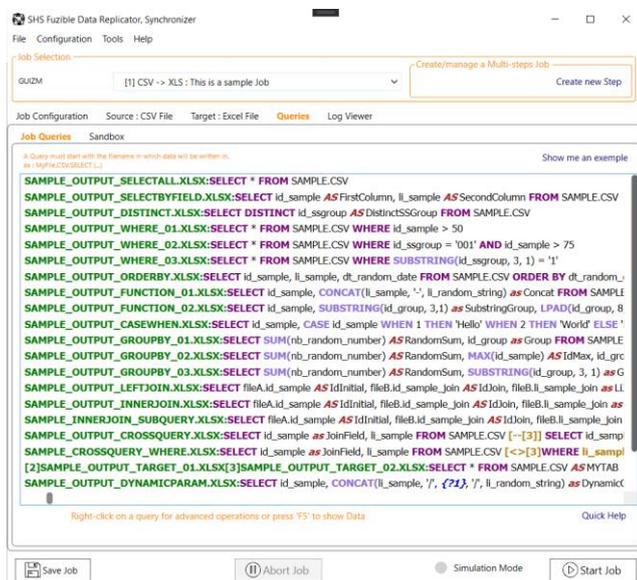
Comme tout programme, la première utilisation peut parfois rebuter. Fuzible ne déroge pas à la règle, mais avant de fermer définitivement le programme, laissez-moi vous présenter quelques pistes d'aide :

## Jobs de démonstration

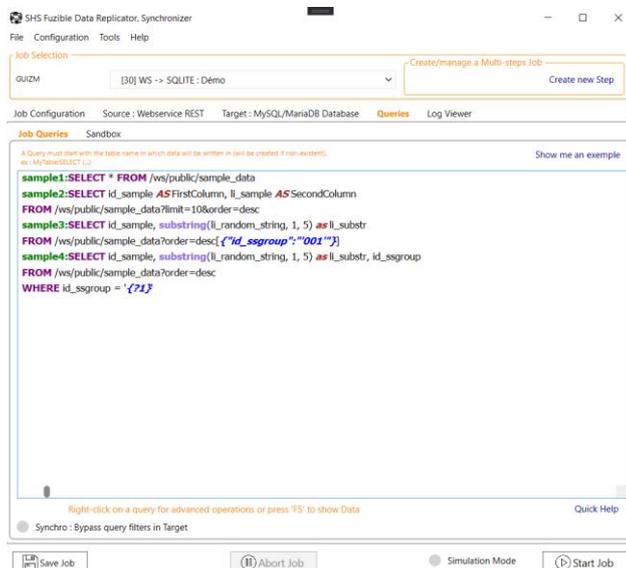
Au premier lancement, Fuzible va créer pour vous 2 Jobs de démonstration qui vous aideront à comprendre comment on requête une autre Source qu'une base de données en utilisant le langage SQL. On accède à ces 2 Jobs par le menu de sélection :



- Un Job qui copie les données de fichiers CSV vers des fichiers EXCEL (XLSX) :

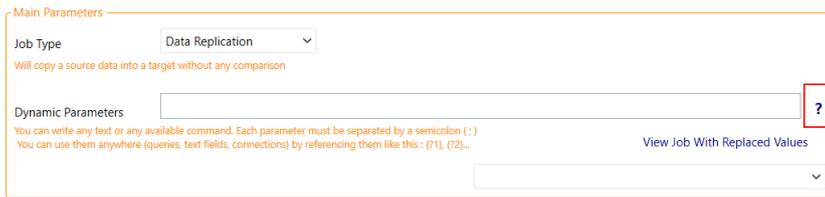


- Un Job qui récupère les données du Webservice de démonstration pour les intégrer dans la base de données locale de Fuzible :



## Boutons d'Aide Rapide

Certaines fonctionnalités possèdent un petit bouton « ? » pour vous donner quelques explications rapides sans avoir à consulter toute la documentation :



## Tutoriaux

Trois tutoriaux ont été programmés de façon à vous permettre de prendre en main le logiciel plus facilement la première fois. Vous pourrez les retrouver ici :

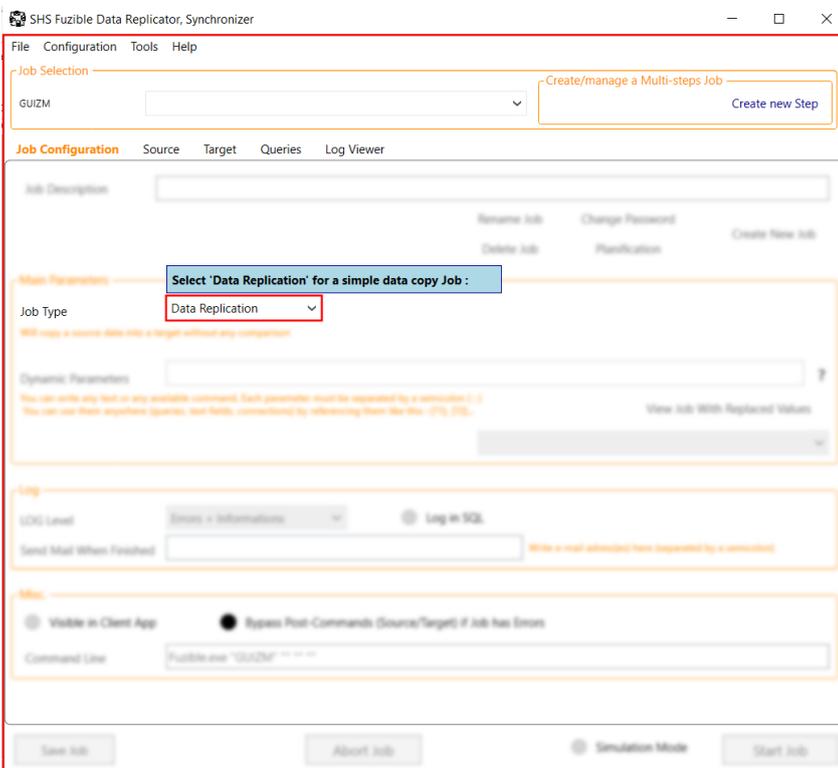
tor, Synchronizer

Help

<b>Tutorials</b>	1-File to SQL Replication (step by step job creation)
About	2-SQL to File Replication (step by step job creation)
Browse the Website	3-File to SQL Synchronization (step by step job creation)
Download DB Browser for SQLite	
Documentation	
@ Contact us	
How to Register	
Enter License Key	

Un script se déclenche alors et vous serez guidé intégralement dans la création d'un Job.

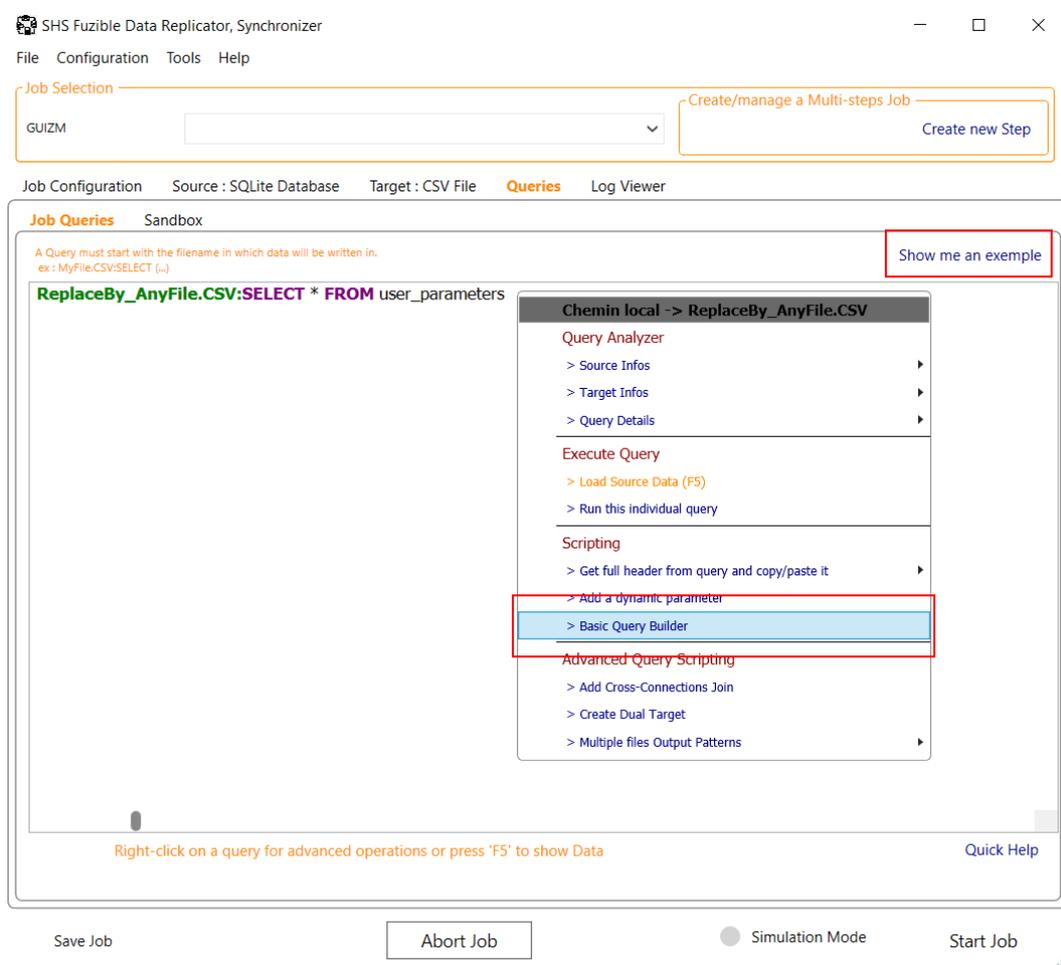
- Touche « Echap » : Quitter le mode Tutorial
- Touche « F1 » : Avancer au pas suivant
- Touche « F2 » : Reculer au pas précédent



## Assistant au requêtage

Lorsque vous créez votre Job, et que vous en serez à saisir les requêtes, 2 options sont mises à disposition pour vous aider à comprendre la philosophie de Fuzible.

- « Show Me an exemple » vous montrera une requête « générique » adaptée au scénario que vous avez programmé dans le Job
- « Basic Query Builder » vous accompagnera dans la création d'une requête simple



Néanmoins, une fois que vous aurez compris la logique du programme, ces 2 options d'assistance ne vous seront d'aucune utilité, elles sont simplement là pour vous faciliter la compréhension du fonctionnement de Fuzible, mais je suis certains que les Tutoriaux et les Jobs de démonstration seront bien plus efficaces !

## Aide en ligne

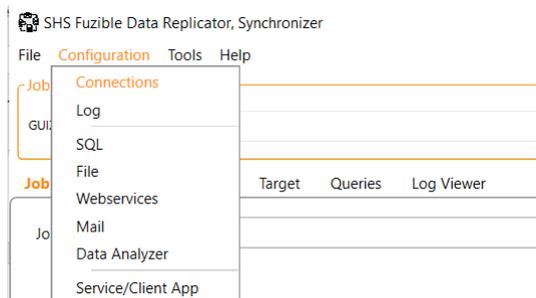
Si vous avez besoin d'aide complémentaire, n'hésitez pas à aller sur le forum (adresse en haut de cette documentation), de me contacter, ou de télécharger des Jobs de démonstration mis à disposition en ligne.

En effet, Fuzible permet de créer des scénarios particulièrement pointus. Sa conception hautement modulaire lui permet de répondre à un grand nombre de besoins, mais il peut être utile de demander de l'aide si vous n'êtes pas sûr du meilleur moyen pour parvenir à vos fins.

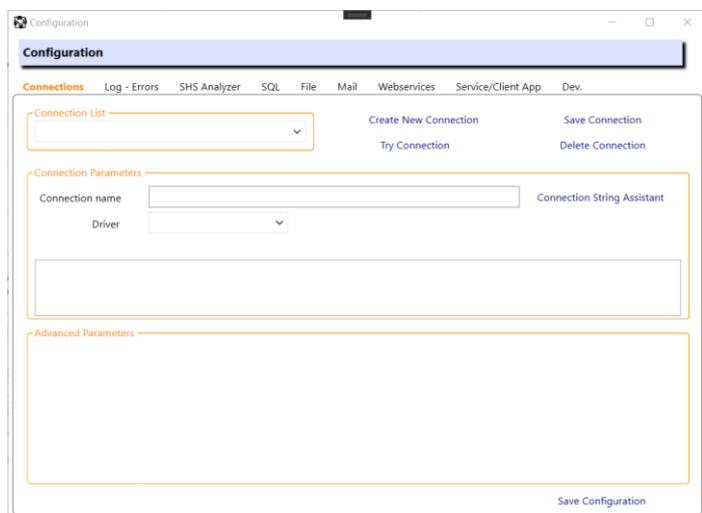
# Configuration du programme

## Connexions

Rendez-vous dans le menu « Configuration / Connexions » pour paramétrer les chaînes de connexion.

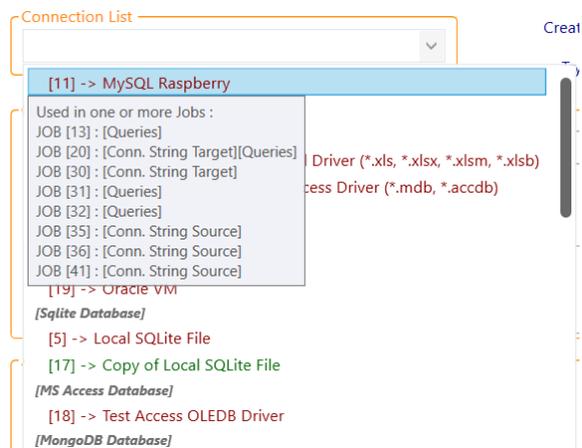


Un menu s'ouvre alors :



Vous pouvez, soit choisir de modifier ou supprimer une connexion existante.

Attention aux impacts, les connexions en rouge dans la liste sont utilisées dans au moins 1 job ! Vous pouvez voir cet impact en survolant le nom de la connexion avec la souris :



Chaque connexion est identifiée par un ID (ex : [10]) ce qui permet de l'utiliser dans les requêtes (voir chapitre associé aux requêtes)

The screenshot shows a user interface for managing database connections. At the top, there is a 'Connection List' section with a dropdown menu showing '[11] -> MySQL Raspberry'. To the right of this list are four buttons: 'Create New Connection', 'Save Connection', 'Try Connection', and 'Delete Connection'. Below the list is the 'Connection Parameters' section for the selected connection. It includes a text input for 'Connection name' (MySQL Raspberry), a 'Connection String Assistant' button, a 'Driver' dropdown (MySQL/MariaDB), a 'Scan Network for instances' checkbox, 'Dates' and 'Decimals' dropdowns (dd/MM/yyyy and ,(comma) respectively).

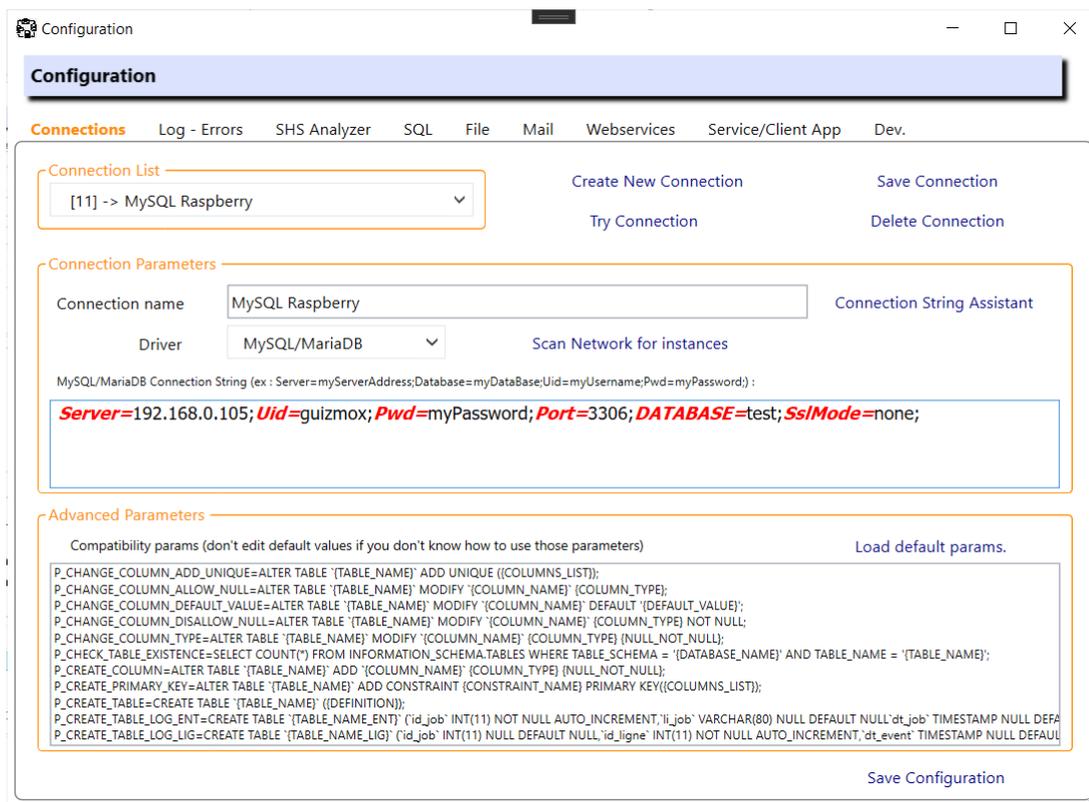
- Les paramètres communs permettent de créer une nouvelle connexion :
  - 1/ A partir d'une connexion existante si déjà sélectionnée
  - 2/ A partir des données saisies par l'utilisateur
- On peut également tester la connexion en cours pour vérifier son bon fonctionnement.
- L'assistant de chaîne de connexion permet de vous accompagner à la création d'une nouvelle connexion.
- Vous pouvez également changer les propriétés de localisation de la connexion : vous pourriez par exemple avoir besoin d'extraire des données US (dates au format MM/dd/yyyy et décimaux séparés par un point) et les importer dans une base de données européenne (dates au format dd/MM/yyyy et décimaux séparés par une virgule)

*Ça ne change pas la localisation de votre OS – seulement la façon dont Fuzible interprète les données de la connexion*

Note : par défaut, ces paramètres sont réglés en fonction de la localisation de votre OS.

- Enfin, dans le cas d'une base de données, une option vous permet de scanner le réseau local pour trouver d'éventuelles instances disponibles (le programme scanne le réseau à la recherche des ports ouverts : ceux-ci sont différents selon le type de BDD choisi).

## Base de données



On peut trouver facilement en ligne la liste des chaînes de connexion : <https://www.connectionstrings.com/>

Toutefois, voici un exemple basique des différents SGBD supportés

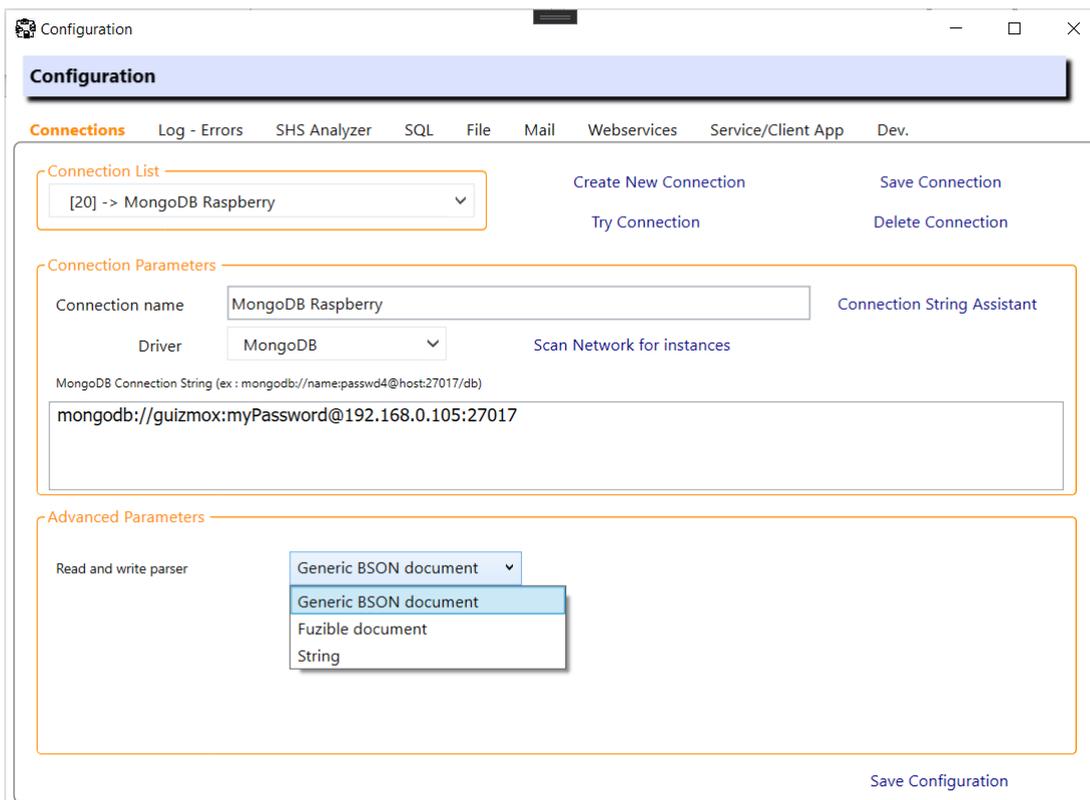
TYPE DE DRIVER	CHAINE ATTENDUE
BDD SQL Server	<b>server=MyServer;DATABASE=MyDatabase;User ID=user;Password=password;Trusted_Connection=True;Connection Timeout=60;Integrated Security=false;</b>
BDD MySQL	<b>server= MyServer;uid= user;pwd= password;DATABASE= MyDatabase;Convert Zero Datetime=True;SslMode=none;</b>
BDD Postgres	<b>Server= MyServer;Port=5432;DATABASE=MyDatabase;Userid= user;Password= password;Ssl Mode=Require;Trust Server Certificate=true;</b>
BDD ODBC	<b>DRIVER={HyperfileSQL};Server Name=MyServer;Server Port=4900;DATABASE=MyDatabase;UID=user;PWD=password;</b>
BDD SQLite	<b>Data Source=C:\Tools\Fuzible\Fuzible.db;Version=3;Foreign Keys=true;</b>
BDD Access	<b>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Tools\Fuzible\accessDB.accdb;Persist Security Info=False;</b>
BDD Oracle	<b>Data Source=WIN81VIRTUEL;User Id=guizmox;Password=myPassword;</b>

Cas des drivers ODBC : Fuzible utilise des requêtes et paramétrages spécifiques pour chaque type de base de données pour son bon fonctionnement. Par définition, ces informations sont propres à chaque SGBD et vous pouvez avoir envie de bénéficier malgré tout de tous les avantages de Fuzible avec un driver non supporté en natif.

La zone « Connection Params » est là pour répondre à ce besoin. Elle permet de paramétrer chaque requête interne de Fuzible afin de la rendre compatible avec n'importe quel SGBD.

PARAMETRE	VALEUR
P_ESCAPE_CHAR	Caractère d'échappement pour délimiter les noms de tables et de champs (souvent, des guillemets)
P_DATE_FORMAT	Format de date natif (DATE, TIMESTAMP, DATETIME)
P_STRING_FORMAT	Format de caractères natif (VARCHAR, VARCHAR2, NVARCHAR)
P_CREATE_PRIMARY_KEY	Code SQL pour la création d'une clé primaire dans une table
P_CREATE_TABLE	Code SQL pour créer une table
P_CHANGE_COLUMN_ALLOW_NULL	Code SQL pour autoriser les valeurs NULL dans une colonne spécifique
P_CHANGE_COLUMN_DISALLOW_NULL	Code SQL pour interdire les valeurs NULL dans une colonne spécifique
P_CREATE_COLUMN	Code SQL pour créer une colonne dans une table
P_CHANGE_COLUMN_TYPE	Code SQL pour altérer le format d'une colonne dans une table
P_CHANGE_COLUMN_DEFAULT_VALUE	Code SQL pour changer ou mettre une valeur par défaut dans une colonne dans une table
P_CHANGE_COLUMN_ADD_UNIQUE	Code SQL pour ajouter une clé unique sur une colonne dans une table
P_SHRINK_TABLE	Code SQL pour nettoyer une table
P_GET_COLUMNS_LIST	Code SQL pour récupérer la liste des colonnes d'une table (attendu : nom, type, taille max, nullable, identité, type, précision numérique, valeur par défaut, unique, clé)
P_GET_COLUMN	Code SQL pour récupérer les informations d'une colonne d'une table (attendu : nom, type, taille max, nullable, identité, type, précision numérique, valeur par défaut, unique, clé)
P_DISABLE_TABLE_CONSTRAINTS	Code SQL pour désactiver les contraintes d'une table (permettre l'insertion sans contrôle de clé par exemple)
P_ENABLE_TABLE_CONSTRAINTS	Code SQL pour activer les contraintes d'une table (contrôle des clés)
P_GET_TABLES	Code SQL pour récupérer la liste des tables d'une BDD
P_GET_TABLES_FILTERED	Code SQL pour récupérer la liste filtrée des tables d'une BDD
P_GET_PRIMARY_KEY	Code SQL pour récupérer la clé primaire d'une table
P_GET_FOREIGN_KEYS	Code SQL pour récupérer la ou les clés étrangères d'une table
P_CHECK_TABLE_EXISTENCE	Code SQL pour contrôler l'existence d'une table sur une BDD
P_DELETE_LOG_EVENTS	Code SQL pour supprimer une ligne de LOG du programme
P_INSERT_LOG_EVENT	Code SQL pour insérer une ligne de LOG du programme
P_UPDATE_LOG_EVENT	Code SQL pour mettre à jour une ligne de LOG du programme
P_CREATE_TABLE_LOG_ENT	Code SQL pour créer la table de LOG du programme (entête)
P_CREATE_TABLE_LOG_LIG	Code SQL pour créer la table de LOG du programme (lignes)
P_CREATE_FROM_SELECT	Code SQL pour créer une table à partir d'une autre (ex: SELECT * INTO [TABLE] FROM [ORIGINAL_TABLE] WHERE 0=1)

*Note : Vous pouvez variabiliser les chaînes de connexion en utilisant { ?1}, { ?2}... dans la chaîne de connexion ; ces paramètres seront remplacés par ceux qui sont mentionnés dans le Job (expliqué plus loin)*



On peut trouver facilement en ligne la syntaxe d'une chaîne de connexion MongoDB :

<https://www.connectionstrings.com/>

Par ailleurs, le pilote MongoDB traite par défaut les données de collection en tant que documents BSON. Vous pouvez néanmoins choisir d'utiliser un type propre à Fuzible (contenant des METADATA additionnelles) ou bien simplement en tant que chaînes de caractères. Tout dépend de l'usage.

# Fichiers

The screenshot shows the 'Configuration' window with the 'Connections' tab selected. The 'Connection List' dropdown shows '[1] -> Local Path'. The 'Connection Parameters' section includes a 'Connection name' field with 'Local Path', a 'Driver' dropdown set to 'CSV File', and a 'Browse Filesystem' button. Below this is a text field containing the local path: 'C:\Users\guizm\source\Workspaces\Fuzible\Fuzible\bin\x64\Debug\FILES\'. The 'Advanced Parameters' section shows a 'Source Type' dropdown menu with 'Local Path' selected. At the bottom right, there is a 'Save Configuration' button.

On attend ici le chemin local ou réseau qui conduit aux fichiers qu'on souhaite traiter dans un Job

Si on change « Source Type », on peut accéder aux fonctions (S)FTP :

The screenshot shows the 'Configuration' window with the 'Connections' tab selected. The 'Connection List' dropdown shows '[23] -> FTP Raspberry'. The 'Connection Parameters' section includes a 'Connection name' field with 'FTP Raspberry', a 'Driver' dropdown set to 'CSV File', and a 'Browse Filesystem' button. Below this is a text field containing the IP address: '192.168.0.101'. The 'Advanced Parameters' section shows a 'Source Type' dropdown set to 'FTP', a 'Username' field with 'ubuntu', a 'Password' field with masked characters, a 'Port' field with '21', and a 'Use SSL connection' radio button. At the bottom right, there is a 'Save Configuration' button.

## Webservice REST

The screenshot shows the 'Configuration' window with the 'Connections' tab selected. The 'Connection List' shows a dropdown menu with '[9] -> Fuzible Webservice de démonstration'. Below it, the 'Connection Parameters' section is filled out: 'Connection name' is 'Fuzible Webservice de démonstration', 'Driver' is 'Rest API', 'Dates' is 'dd/MM/yyyy', and 'Decimals' is '(comma)'. The 'Webservice URL' is 'https://www.fuzible-app.com'. The 'Advanced Parameters' section shows 'Choose Template (optional)' set to 'Generic Rest API', 'Authorization Method' set to 'API auth (header)', and 'Key' set to 'Authorization'. The 'Value' field contains a series of dots, and the 'Optional URL to get value dynamically' field is empty. The 'Proxy URL', 'Proxy port', and 'Proxy Type' (set to 'NONE') are also visible.

Un webservice est un peu plus compliqué à paramétrer car il existe plusieurs manières de dialoguer avec eux.

C'est pourquoi un menu déroulant vous permet de vous simplifier la tâche en chargeant un « template » pour quelques API connues

This screenshot shows the 'Configuration' window with the 'Facebook Graph API' template selected in the 'Choose Template (optional)' dropdown. The 'Connection Parameters' section is filled out: 'Connection name' is 'Facebook API', 'Driver' is 'Rest API', 'Dates' is 'dd/MM/yyyy', and 'Decimals' is '(comma)'. The 'Webservice URL' is 'https://graph.facebook.com/'. The 'Advanced Parameters' section shows 'Authorization Method' set to 'API auth (header)', 'Key' set to 'Authorization', and 'Scope' set to 'grant\_type=client\_credentials'. The 'Value' field contains a series of dots, and the 'Optional URL to get value dynamically' field is 'https://graph.facebook.com/oauth/access\_token'. The 'Proxy URL', 'Proxy port', and 'Proxy Type' (set to 'NONE') are also visible.

En choisissant un exemple dans la liste et en le chargeant, les champs vont automatiquement se pré-remplir pour vous simplifier le paramétrage et la compréhension du gestionnaire de connexion des API.

Si vous souhaitez configurer vous-même l'accès à une API non référencée, utilisez « Generic Rest API », et configurez manuellement les options, ainsi que le mode d'authentification parmi la liste de ceux qui vous sont proposés.

L'exemple ci-dessous présente par exemple le paramétrage de l'API v3 de Youtube.

The screenshot shows the configuration interface for a 'Youtube API v3' connection. The 'Connection Parameters' section includes a 'Connection name' field with 'Youtube API v3', a 'Driver' dropdown set to 'Rest API', and a 'Webservice URL' field containing 'https://www.googleapis.com/youtube/v3/'. The 'Advanced Parameters' section shows a 'Choose Template' dropdown set to 'Youtube v3 API', an 'Authorization Method' dropdown set to 'API auth (param)', and a 'Key' field with the value 'key'. There is also an 'Optional URL to get value dynamically' field with a question mark icon.

Autre exemple, pour interroger GLPI, on doit au préalable contacter un service qui va fournir un ID de session. Il faut donc, en plus d'intégrer à l'entête les paramètres d'authentification fournis par le prestataire, indiquer l'URL qui permet de récupérer le jeton de session (Optional URL to get value dynamically) : Chaque connexion au webservice passera alors de manière automatique par la récupération du token de session, et permettra de récupérer des données de tous les objets exposés.

The screenshot shows the 'Main Parameters' section for an 'API GLPI' connection. The 'Connection List' dropdown is set to '[93] -> API GLPI'. The 'Connection Name' field contains 'API GLPI'. The 'Driver' dropdown is set to 'Webservice REST'. The 'Webservice URL' field contains 'http://[redacted]/glpi/apirest.php/'. The 'Parameters' section shows a 'Proxy URL' field, a 'Proxy Port' field, and a 'Headers' field with the value 'Authorization:user\_token 1da3wdPG098gmJowcjZTiu4JAzqrIF3IPVM4noji;app-token:NUJjYrbZik3s0YYW6jKxis9TvyfZTgwFVUB0S33U'. The 'Authorization Method' dropdown is set to 'API Auth (Param)', and the 'Key' field contains 'session\_token'. The 'Optional URL to get value dynamically' field contains 'http://[redacted]/glpi/apirest.php/initSession/'.

Dernier exemple, pour interroger l'API Graph de Microsoft, on utilise une authentification OAuth 2.0 avec un paramétrage spécifique qui est décrit dans la documentation de l'API disponible en ligne :

The screenshot shows the configuration interface for a 'Microsoft GraphQL' connection. The 'Connection name' field contains 'Microsoft GraphQL'. The 'Driver' dropdown is set to 'Rest API'. The 'Webservice URL' field contains 'https://graph.microsoft.com/v1.0/'. The 'Advanced Parameters' section shows a 'Choose Template' dropdown set to 'Microsoft Graph API', an 'Authorization Method' dropdown set to 'OAuth 2.0', and a 'Scope' field with the value 'https://graph.microsoft.com/default'. The 'HTTP Params' field contains 'grant\_type=client\_credentials'. The 'Token URL' field contains 'https://login.microsoftonline.com/a65e48768-p189-8999-j4547-54898f15g5edf/oa'. The 'Client ID + Client Secret' field contains '565489f2ef-azeffg5684-4543fgo-547741' and a masked password field.

# Mail

The screenshot shows a 'Configuration' window with a menu bar (Connections, Log - Errors, SHS Analyzer, SQL, File, Mail, Webservices, Service/Client App, Advanced). The main area is divided into three sections:

- Connection List:** A dropdown menu showing '[10] -> Fuzible Email'. Buttons for 'Create New Connection', 'Save Connection', 'Try Connection', and 'Delete Connection' are present.
- Connection Parameters:** Includes a 'Connection name' field (Fuzible Email), a 'Driver' dropdown (Mailbox), and a 'Connection String Assistant' button. Below is a 'Customized Connection String (Use assistant):' field containing: `SERVER_SEND=ssl0.ovh.net;SERVER_RECEIVE=ssl0.ovh.net;get_protocol=IMAP;AUTH_PROTOCOL=IMAP;PORT_SEND=465;PORT_RECEIVE=993;USERNAME=guillaume@fuzible-app.com;PASSWORD=myPassword;ssl=1`
- Advanced Parameters:** Includes 'Protocol' (IMAP), 'Authentication' (NONE), a radio button for 'Use SSL connection' (checked), and fields for 'Proxy URL (if needed)', 'Proxy port', 'Proxy Type' (NONE), and 'User,Pwd'.

A 'Save Configuration' button is located at the bottom right of the configuration area.

La chaîne de connexion « Mail » est propre à Fuzible. On peut utiliser le « Connection String Assistant » pour s'aider à la construire, mais en substance, elle attend les paramètres suivants :

SERVER\_SEND : URL du service SMTP

SERVER\_RECEIVE : URL du service POP ou IMAP

GET\_PROTOCOL : Le protocole de réception : IMAP ou POP

AUTH\_PROTOCOL : Le protocole de sécurisation : TLS (10,11,12,13), NONE

PORT\_SEND : Port d'envoi

PORT\_RECEIVE : Port de réception

USERNAME : l'adresse mail associée

PASSWORD : le mot de passe associé à l'adresse mail

SSL : 1 ou 0

Vous pouvez éventuellement configurer un proxy.

## Active Directory

Configuration

Connections Log - Errors SHS Analyzer SQL File Mail Webservices Service/Client App Dev.

Connection List  
[21] -> Fuzible Active Directory Demo

Create New Connection Save Connection  
Try Connection Delete Connection

Connection Parameters

Connection name: Fuzible Active Directory Demo  
Driver: Active Directory  
LDAP URL (ex : LDAP://DC=mydomain,DC=en) (ex2 : LDAP://mydomain) :  
LDAP://fuzible.lan

Advanced Parameters

Compatibility params (don't edit default values if you don't know how to use those parameters) Load default params.

```
P_AD_SEARCH_GROUP=(&(objectClass=group){[SEARCH_PROPERTY]=*})
P_AD_SEARCH_GROUPS=(&(objectClass=group))
P_AD_SEARCH_USER=(&(objectClass=user)(objectCategory=person)(CN=users){[SEARCH_PROPERTY]=*})
P_AD_SEARCH_USERS=(&(objectClass=user)(objectCategory=person))
```

Save Configuration

La chaîne de connexion est très basique, voici deux exemples :

**LDAP://DC=fuzible,DC=fr**

**LDAP://fuzible.lan**

En plus de la chaîne de connexion, vous pouvez intervenir sur les paramètres de compatibilité :

P_AD_SEARCH_GROUP	Requête de recherche d'un groupe en particulier	SEARCH_PROPERTY sera remplacé par la valeur définie dans le Job (voir plus loin). Il s'agit du champ « clé » disponible dans l'AD qui permet de définir l'unicité d'un groupe
P_AD_SEARCH_GROUPS	Requête de recherche des groupes	
P_AD_SEARCH_USER	Requête de recherche d'un utilisateur en particulier	SEARCH_PROPERTY sera remplacé par la valeur définie dans le Job (voir plus loin). Il s'agit du champ « clé » disponible dans l'AD qui permet de définir l'unicité d'un utilisateur
P_AD_SEARCH_USERS	Requête de recherche des utilisateurs	

## Log

Cet onglet permet de paramétrer toutes les options relatives au LOG du programme. Celui-ci produit par défaut des fichiers « .TXT » dans le répertoire de l'application :

**C:\Users\Public\Documents\Fuzible\LOG**

... mais on peut également l'intégrer dans une base de données, ou bien l'envoyer par mail.

The screenshot shows the 'Configuration' window with the 'Log - Errors' tab selected. The window has a title bar with 'Configuration' and standard window controls. Below the title bar is a menu bar with 'Connections', 'Log - Errors', 'SHS Analyzer', 'SQL', 'File', 'Mail', 'Webservices', 'Service/Client App', and 'Dev.'. The main area is divided into sections for 'SQL Log - Connection String' and 'Mail Log - Connection String'. The SQL Log section has a text field for the connection string containing 'Data Source=C:\Users\guizm\source\Workspaces\fuzible\Fuzible\bin\x64\Debug\win-x64\Fuzible.db;Version=3;foreign keys=true' and a dropdown for 'Driver' set to 'SQLite'. The Mail Log section has a text field for the connection string containing 'SERVER\_SEND=smtp.office365.com;SERVER\_RECEIVE=outlook.office365.com;GET\_PROTOCOL=IMAP;auth\_protocol=NONE;PORT\_SEND=587;PORT\_RECEIVE=993;USER\_NAME=myMail@hotmail.com;PASSWORD=myPassword;ssl=1'. Below this are radio buttons for 'Configure your mail' (selected), 'Use SSL connection', and a dropdown for 'Authentication' set to 'NONE'. At the bottom, there are input fields for 'Keep log until (days)' (8), 'Keep processed files until (days)' (30), a radio button for 'Don't send mail if job finished without any error', a radio button for 'Show CPU/RAM alerts' (selected), and 'Abort Job when errors exceeds' (10). A 'Save Configuration' button is at the bottom right.

### SQL Log

Il est fortement conseillé de changer d'instance SQL pour stocker les LOG, par défaut, c'est la base locale de Fuzible qui est utilisée, mais si vous souhaitez exploiter le LOG ailleurs, vous devriez utiliser une base de donnée réseau (MySQL, SQL Server, Oracle, Postgres...)

Dans ce cas, le programme créera automatiquement les tables requises dans cette nouvelle instance.

### Mail Log

Par défaut, ce champ est vide. Cette connexion à une boîte mail permet d'envoyer des comptes-rendus de Job à une ou plusieurs adresses mail. La connexion se configure comme n'importe quelle connexion (vu précédemment).

### Options

Abort Job when errors exceeds	On définit le nombre d'erreurs maximal qu'on tolère avant de considérer qu'un Job n'a pas fonctionné. Si ce nombre est dépassé pendant l'exécution du Job, alors celui-ci est interrompu
Keep log until (days)	Le LOG "fichier" et "SQL" peut se nettoyer tout seul au-delà d'un nombre de jours paramétré ici
Keep processed files until (days)	Avec une chaîne de connexion de type « Fichier », Fuzible peut faire le ménage dans le répertoire en effaçant des fichiers trop anciens. On définit ici le nombre de jours de préservation des fichiers.
Don't send mail if job finished without any error	Si le job est paramétré pour un envoi de mail en fin de traitement, si celui-ci s'est déroulé sans erreurs, on peut bloquer l'envoi du mail
Show CPU/RAM alerts	Lorsque le CPU atteint 100%, l'avertissement est inscrit dans le LOG, ce qui vous permet d'évaluer le besoin de ressources supplémentaires sur le serveur qui héberge l'application

## SHS Analyzer

Fuzible contient un moteur d'analyse de données. C'est-à-dire qu'il analyse toutes les données source, notamment pour pouvoir créer avec le plus de justesse possible des tables SQL lorsque celles-ci sont inexistantes et qu'un Job possède comme destination une base de données.

Ces opérations sont très consommatrices de ressources et peuvent facilement solliciter la machine à 100%. Pour éviter cette saturation, on peut indiquer à Fuzible de n'utiliser qu'une partie des ressources.

Configuration

Connections Log - Errors **SHS Analyzer** SQL File Mail Webservices Service/Client App Advanced

Multithreading (cores)

Parallel Threads when loading/copying huge SQL datasets or CSV files

Cast rounded .0 decimals as integers (ex : 5.00 will be interpreted as an integer)

Cast numbers starting with '0' as integers (ex : 0150 will be interpreted as an integer)

Multi-target Queries : write both targets in parallel (CPU-RAM intensive)

Save Configuration

### Options

Multithreading	Nombre de coeurs CPU utilisables par le programme (minimum = 2)
Parallel Threads when loading/copying huge CSV datasets or CSV	Normalement, Fuzible charge toutes les données Source en RAM, puis les traite pour les intégrer dans la Cible. Dans le cas de gros fichiers CSV ou tables SQL (millions de lignes), le programme charge et intègre les données par morceaux. Ce traitement particulier peut s'effectuer en plusieurs threads : chargement des données à suivre pendant insertion du batch précédent afin d'accélérer le processus. Plus vous ajoutez de threads, plus les besoins en RAM+CPU seront importants.
Cast rounded .0 decimals as integers	Lors de l'analyse des données, on peut affiner la détection des entiers en considérant qu'une valeur telle que 5.00 est finalement un nombre entier (5)
Cast numbers starting with '0' as integers	Lors de l'analyse des données, on peut affiner la détection des entiers en considérant qu'une valeur telle que 0546 ou bien 0000477 est finalement un nombre entier
Multi-Target Queries : Write both targets in parallel	Sur un job possédant des requêtes multi-cibles, on peut choisir d'écrire en parallèle dans les 2 cibles pour aller plus vite (nécessite BEAUCOUP de puissance)

Sont regroupés ici tous les paramètres relatifs aux connexions « SQL »

The screenshot shows a 'Configuration' window with a tabbed interface. The 'SQL' tab is active. The parameters are as follows:

- Command timeout: 360
- Transaction size: 128
- Bulk Insert Mode Rows/Batch: 1000
- Rows quantity to get before copying (Direct Stream Mode): 100
- Synchro. table Log: fuzible\_synchro\_records
- Auto shrink tables:

A 'Save Configuration' button is located at the bottom right of the configuration area.

### Options

Command Timeout	Timeout d'exécution d'une commande SQL
Transaction size	Lorsqu'un Job dont la cible est une BDD SQL, les opérations « INSERT/DELETE/UPDATE » sont encadrées dans des transactions (ce qui permet de réaliser un ROLLBACK si la requête échoue). Le nombre d'opérations encadrées par une transaction est à définir ici. Plus ce nombre est grand, plus les ressources de la BDD cible seront sollicitées
Bulk Insert Mode Rows/Batch	S'applique quand on choisit le mode « Bulk Insert » dans un Job. Cela définit le nombre de lignes qui seront copiées par paquet (ex : si il y a 10 000 lignes à copier et que vous avez choisi des batchs de 1000 lignes, 10 batchs seront donc envoyés à la base de données Cible)
Row quantity to get before copying (Direct Stream Mode)	Le mode « Direct Stream » se paramètre dans le Job et s'utilise quand les tables SQL à requêter sont énormes (plusieurs millions de lignes) : pour éviter une surcharge mémoire, les données sont chargées et copiées dans la Cible à la volée. Vous pouvez définir ici la taille de chaque « morceau » à transférer (en nombre de lignes) Augmenter cette valeur demandera plus de RAM.
Syncho Table LOG	Pour les Job paramétrés en mode « Synchronization », le résultat de la synchronisation est stocké sur une table séparée (pour consultation et information). Le nom de la table est paramétrable
Auto shrink tables	Option qui permet de nettoyer une table (cible) après l'avoir traitée

# File

Sont regroupés ici tous les paramètres relatifs aux connexions « FICHIER »

Configuration

Connections Log - Errors SHS Analyzer SQL **File** Mail Webservices Service/Client App Dev.

Working path

Source files move path   Add datetime prefix

CSV separators

CSV/XLS : Force Integration of row(s) not matching the header length

Save Configuration

## Options

Working path	Lorsqu'un Job a pour source une connexion « Fichier », et qu'on a paramétré celui-ci pour qu'il déplace ces fichiers en fin de traitement, le répertoire dans lequel ils sont déplacés se définit ici
Source files move path	Inutilisé pour le moment
Add datetime prefix	En plus de déplacer les fichiers en fin de traitement sur les Jobs paramétrés tels quel, ces fichiers peuvent en plus avoir un préfixe sous la forme « YYYYMMddHHMMss_Monfichier.xxx », pour éventuellement faciliter leur recherche ultérieure (si besoin était)
CSV separators	Liste des séparateurs CSV acceptés (\t signifie "tabulation"). On peut en ajouter si toutefois on traite de fichiers possédant des séparateurs non-standards
CSV/XLS : Force Integration of row(s) not matching the header length	Si vos fichiers source (CSV ou XLS) ont des lignes inconsistantes (différences de taille avec l'entête), on peut forcer leur intégration, ou les éviter. Dans tous les cas, un "WARNING" sera inscrit dans le LOG.

## Mail

Sont regroupés ici tous les paramètres relatifs aux connexions « MAIL »

Configuration - □ ×

**Configuration**

Connections Log - Errors SHS Analyzer SQL File **Mail** Webservices Service/Client App Dev.

Admin email address

Max. length before sending sheet as an XLS attachment

Max. attachment file size (in kB)

Timeout

[Save Configuration](#)

### Options

Admin email address	Adresse mail de l'administrateur du programme (il sera alors en copie cachée de tous les mails produits par un Job)
Max. length before sending sheet as an XLS attachment	Lorsqu'un Job est configuré avec comme cible une adresse mail, on peut choisir que les données qui y sont incluses soient intégrées dans le corps du mail (se paramètre dans le Job). Néanmoins, si ce contenu est trop volumineux, Fuzible peut, à la place, créer un fichier EXCEL qui sera alors joint au mail. Ce paramètre indique le nombre de caractères maximal produits par les données source avant que le Job ne bascule en mode « pièce jointe »
Max. attachment file size	Taille maximale d'un fichier en PJ d'un mail. Au-delà de cette limite, la PJ ne sera pas envoyée
Timeout	Timeout d'exécution d'une opération sur le serveur mail

# Webservices

Sont regroupés ici tous les paramètres relatifs aux connexions « WEBSERVICES »

The screenshot shows a configuration window titled 'Configuration' with a standard Windows-style title bar (minimize, maximize, close). Below the title bar is a navigation menu with the following items: 'Connections', 'Log - Errors', 'SHS Analyzer', 'SQL', 'File', 'Mail', 'Webservices' (highlighted in orange), 'Service/Client App', and 'Advanced'. The main content area contains the following settings:

- Timeout:** A text input field containing the value '30000'.
- Default encoding:** Two dropdown menus, the first containing 'ISO-8859-1' and the second containing 'UTF-8'.
- Save responses in source:** A radio button selected next to the text 'Allow schema alteration'.
- SOQL (Salesforce) Queries:** A radio button selected next to the text 'Get Records Datatable Only'.

At the bottom right of the configuration area, there is a 'Save Configuration' button.

## Options

Timeout	Timeout d'exécution d'une opération sur le serveur distant
Default Encoding	Lorsque les données d'un WS sont traitées, le programme détermine l'encodage des caractères par ce que le serveur distant lui fournit. Si cette information n'est pas présente, on peut forcer par défaut un encodage en particulier. 2 valeurs à indiquer : la 1 <sup>ère</sup> pour les WS de type « REST » et la 2 <sup>ème</sup> pour le WS NUXEO.
Save responses in Source : Allow add+alter columns	Lors de l'envoi de données vers un webservice, celui-ci produit généralement un retour d'information (au format XML ou JSON). Ces réponses peuvent être récupérées et stockées dans la base de données SOURCE (si SOURCE=BDD) ou bien sous forme de fichier(s) (si SOURCE<>BDD). Dans le cas où la source est une base de données, on peut autoriser à Fuzible de changer le format des colonnes cible si les données renvoyées par le webservice ne sont pas ou plus compatibles avec la table dans laquelle elles vont être intégrées
SOQL Queries : Get Records Datatable Only	Cette option permet, en cas d'utilisation du langage SOQL, de ne récupérer que la table de données en se passant de celle des attributs et celle du résumé de requête

## Service/Client App

Fuzible est fourni avec un module « Service/Client ».

L'application « Service » fait l'objet d'un paragraphe dédié, mais en substance, elle permet de dépiler les Jobs qui sont invoqués ou par un utilisateur (via l'application « Client ») ou par l'orchestrateur (planificateur de Jobs)

L'application « Client », quant à elle, est une mini-application qui se contente de mettre à disposition d'utilisateurs « lambda » des Jobs afin qu'ils déclenchent leur exécution quand ils le souhaitent.

Configuration

Connections Log - Errors SHS Analyzer SQL File Mail Webservices **Service/Client App** Dev.

SQL Service app - Connection string `Data Source=C:\Users\Guizm\source\Workspaces\Fuzible\Fuzible\bin\x64\Debug\Fuzible.db;Version=3;`

Driver SQLite

Create Windows Task (Task Manager) Create required service tables (planification, stack, jobs)

Working userspace GUIZM

Parallel Jobs 2

Keep stack until (days) 30

Client app flooding delay (min) 15

Sharing the 'Client' app requires you to provide 'FuzibleClient.exe', as well as 'CLIENTAPP.INI' files to users

Save Configuration

Chaîne de connexion : par défaut, Fuzible utilisera la connexion SQLite locale mais ce mode de fonctionnement est très déconseillé. En effet, l'application « Service » a pour vocation de dialoguer avec l'application « Client », distribuée à de nombreux utilisateurs. Dans un environnement d'entreprise, vous n'avez probablement pas envie que n'importe qui puisse avoir le chemin réseau pointant vers Fuzible ouvert. En revanche, votre organisation pourra être configurée de telle sorte que les postes client puissent dialoguer avec un serveur de BDD en ouvrant le port qui pointe vers une instance SQL.

### Options

Create Required Service Tables	Ce bouton vous permet de créer automatiquement toutes les tables dont Fuzible a besoin pour que l'application de service fonctionne correctement (en cas de changement de chaîne de connexion)
Create Windows Task	Crée la tâche planifiée « Fuzible Service App » dans le planificateur de tâches Windows afin de permettre à l'application de service de fonctionner.
Working User	Il s'agit du compte que l'application « Service » utilise pour fonctionner. Ce compte correspond à l'un des utilisateurs de Fuzible : l'application de service ne peut dialoguer qu'avec un des utilisateurs, pour éviter une gestion anarchique de l'orchestration des Jobs programmés et des Jobs mis à disposition de l'application Client.
Parallel Jobs	L'application de Service dépile les Jobs invoqués au fur-et-à mesure. Elle peut en exécuter plusieurs en parallèle mais attention au risque de surcharge. Vous choisissez ici le nombre de Jobs qu'elle peut lancer en parallèle : Ce paramètre se définit en fonction des ressources allouées au serveur/ordinateur qui reçoit l'application. C'est à votre appréciation.
Keep Stack Until (days)	Durée de conservation des données de pile des Jobs invoqués. Passé ce délai, les précédentes entrées seront supprimées.
Client App Flooding Delay	L'application « Client » permet aux utilisateurs de lancer des Jobs. Ils peuvent donc ainsi « s'amuser » à les déclencher plusieurs fois de suite. Ce paramètre permet de laisser un délai entre 2 lancements successifs du même Job, afin de les empêcher de « spammer » la file d'attente et de surcharger la machine de traitement.

## Dev.

Vous trouverez ici un certain nombre de paramètres propres au fonctionnement du programme. Si celui-ci s'avérait capricieux pour tel ou tel cas de figure, il serait possible d'intervenir sur ces paramètres. Il est néanmoins déconseillé de les manipuler sans comprendre leur fonctionnement, au risque de compromettre vos Jobs existants. Le forum en ligne vous permettra peut-être d'échanger avec d'autres utilisateurs à ce sujet.

### Configuration

Connections
Log - Errors
SHS Analyzer
SQL
File
Mail
Webservices
Service/Client App
Advanced

**CSV/EXCEL File Analysis**

Maximum rows analyzer (header, separator)

Source CSV splits when row count exceeds   Don't check files coming from (S)FTP

Header analyzer - Depth analysis resemblance offset

Header analyzer - Depth analysis unicity offset

Header analyzer - Row offset before depth analysis

**SQL Operations**

Errors overflow (more failed queries will throw an error)

Max decimals (more will be rounded)

Query characters (debug)

**Security**

Shared Users (all users shares the same session)

Software Registration Method

**Miscellaneous**

Enable Query Assistant      Analyze Files - Max. Size :       Shell operations timeout (min)

Json parser : replace special chars in columns names

[Save Configuration](#)

XLS/CSV maximum rows analyzer	Fuzible analyse automatiquement les fichiers EXCEL et CSV pour déterminer si la première ligne est un entête. Dans le cas des fichiers avec un grand nombre de lignes, l'analyseur d'entêtes n'a pas forcément besoin d'analyser toutes les lignes pour le détecter. On peut paramétrer ici le nombre maximal de lignes à analyser
Source CSV split when row count exceeds	Si la source est un fichier CSV, le programme compte systématiquement le nombre de lignes de celui-ci. Si ce nombre excède une certaine quantité, Fuzible peut traiter le fichier en plusieurs « morceaux » afin de ne pas surcharger la RAM du serveur. La valeur indique le nombre de lignes maximum contenues dans chaque « morceau »
Don't check files coming from (S)FTP	En lien avec l'option précédente, si les fichiers proviennent d'un serveur (S)FTP, il est possible de bypasser le comptage des lignes, car cette analyse nécessite de télécharger le fichier, ce qui peut pénaliser fortement les performances du programme.
Header Analyzer – Depth Analysis Resemblance Offset	Seuil qui détermine le pourcentage de ressemblance entre la première ligne du fichier analysé et toutes les autres. La ressemblance est calculée par en interne en faisant plusieurs tests sur le fichier. L'analyse peut être affichée si le Job est configuré en mode « Debug »
Header Analyzer – Depth Analysis Unicity Offset	Lors de l'analyse automatique du contenu d'un fichier EXCEL ou CSV, Fuzible détermine l'unicité de la première ligne du fichier par rapport aux autres. - Si le pourcentage d'unicité dépasse le seuil, et qu'en plus, le pourcentage de ressemblance de la

	<p>première ligne est supérieur au seuil « Resemblance Offset », on considère que la première ligne n'est pas un header</p> <ul style="list-style-type: none"> <li>- Si le pourcentage d'unicité dépasse le seuil, et qu'en plus le pourcentage de ressemblance de la première ligne est inférieur au seuil « Resemblance Offset », on considère que la première ligne est un header</li> <li>- Si le pourcentage d'unicité est en dessous du seuil, et qu'en plus le pourcentage de ressemblance de la première ligne est inférieur ou égal au seuil « Resemblance Offset », on considère que la première ligne est un header</li> <li>- Si le pourcentage d'unicité est en dessous du seuil, et qu'en plus le pourcentage de ressemblance de la première ligne est supérieur au seuil « Resemblance Offset », on considère que la première ligne n'est pas un header</li> </ul>
Header Analyzer – Row offset before depth analysis	Dans le cas de fichiers avec peu de lignes, la détection de l'header peut s'avérer délicate (échantillon trop faible pour calculer un pourcentage de ressemblance entre la première ligne et toutes les autres). En dessous du seuil (nombre de lignes) saisi ici, le programme passe en mode « fin ». Il va analyser finement le fichier, et jouer sur les 2 paramètres mentionnés précédemment pour déterminer l'entête
Max Decimals	En mode « cible = BDD », définit le nombre de décimales maximum toléré lors de l'insertion des données (cas où les données source auraient par exemple 35 chiffres après la virgule et qu'on souhaite limiter cette quantité)
Errors Overflow	Si une requête exécutée n'a pas fonctionné, Fuzible la marque en « warning ». Vous pouvez définir ici un seuil de requêtes qui n'ont pas abouti à partir duquel Fuzible considère qu'il s'agit d'une erreur et non plus d'un « warning » (déclenchera un message d'erreur)
Query characters (Debug)	Lorsqu'un Job est paramétré avec le LOG en mode « Debug », et que la cible du Job est une BDD, toutes les requêtes INSERT/DELETE/UPDATE vont être inscrites dans le fichier de DEBUG. Le volume peut être considérable, il est donc possible de limiter la quantité de caractères intégrés au LOG
Shared Users	En temps normal, chaque utilisateur Windows possède sa propre session : les connexions, configuration et Jobs sont propres à chacun, les fonctions d'import permettant à chaque utilisateur de se passer des Jobs. Même s'il est possible de charger un autre espace utilisateur, celui-ci sera en lecture seule. Dans votre organisation, vous pourriez avoir besoin d'une session partagée : tous les utilisateurs seront automatiquement routés vers une seule et unique session, celle qui est définie ici. Ainsi, l'ensemble du programme est partagé, accessible à tous. Cela implique que les paramètres de confidentialité (notamment les chaînes de connexion) sont également visibles de tous.
Registration Mode	Méthode de communication avec le serveur de Fuzible quand vous souhaitez enregistrer le programme. Vous devriez laisser « Webservice » par défaut. Le mode « Mail » ne servira que dans le cas où il vous serait impossible de communiquer avec le serveur (mode hors-ligne)
Enable Query Assistant	Active ou désactive l'assistant aux requêtes de l'interface (colorisation, contrôles de cohérence, propositions de saisie). L'assistant peut consommer beaucoup de ressources car dans le cas où la source est une BDD, il l'interroge pour connaître les tables disponibles et les champs de chaque table. Si c'est une source de type « fichier », il parcourt le répertoire pour connaître le nom des fichiers, et scanne chacun d'entre eux pour trouver les entêtes
Shell Operations Timeout	Lorsque qu'une commande Shell Pré ou Post-Job est programmée, le temps d'exécution maximum par défaut est de 60 minutes. Au-delà de ce temps, la tâche est interrompue. On peut donc ici intervenir sur ce paramètre Ce paramètre est également utilisé par l'application « Service ». Lorsqu'elle exécute des Jobs, elle lance Fuzible de manière paramétrée. Le programme est alors soumis aux mêmes règles de Timeout
Json Parser : Replace special chars in columns names	Lorsqu'un fichier JSON (ou une réponse webservice) est interprété, les noms de colonne contiennent parfois des caractères spéciaux : on peut décider ici de les remplacer par un caractère plus conventionnel (l'underscore)
Analyze Files – Max Size	Quand vous utilisez le « Query Assistant » et que la Source du Job est un fichier, celui-ci va être analysé pour déterminer ses colonnes et les ajouter à l'auto-complétion. Si le fichier dépasse une taille max (en Kb), l'analyse sera évitée afin de ne pas consommer trop de puissance (CPU/RAM). Dans le cas où vous ne connaissez pas les colonnes à l'avance, il est alors recommandé de faire un « SELECT * FROM... », consulter les données afin de pouvoir utiliser les noms de colonnes après-coup

# Outils

## Export Job (XML)

Ce menu permet d'exporter un Job au format XML : Cette extraction intègre tous les paramètres du Job, les connexions associées (ainsi que celles qui peuvent être présentes dans les champs de script), et les requêtes. Le fichier est chiffré de telle sorte qu'il ne puisse être importé dans un autre environnement que si l'utilisateur connaît son mot de passe : il s'agit du mot de passe par défaut du Job.

## Import Job (XML)

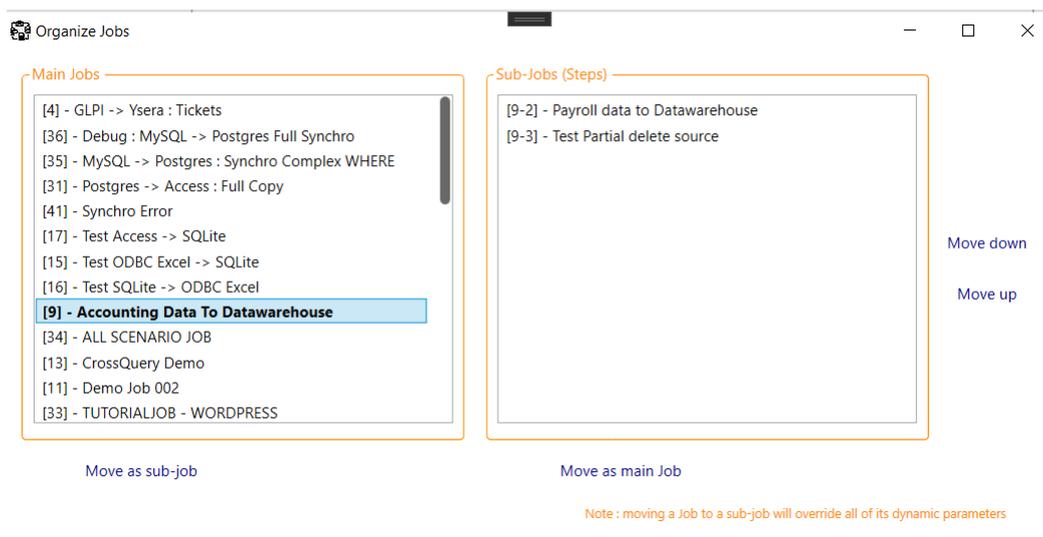
Ce menu vous permet d'importer un Job au format XML : L'intégration dans votre environnement inclut la création des connexions associées à ce Job, les paramètres et les requêtes. Le mot de passe du Job est requis pour réaliser l'importation.

## Reorganize Jobs

On peut réorganiser les jobs entre eux. En effet, il est possible de créer des jobs « multi-pas » (qui sont ni plus ni moins que plusieurs jobs lancés les uns à la suite des autres.

Ce menu permet :

- D'envoyer un job principal dans un autre job en tant que « pas »
- D'extraire un « pas » dans un job « multi-pas » pour le mettre en tant que job principal
- De réorganiser l'ordre des « pas » dans un job « multi-pas »



Tous les Jobs en gras servent à signifier qu'ils sont « multi-pas ». Quand on clique sur l'un d'eux, on voit la liste des sous-jobs.

## Move as Sub Job

Permet de déplacer un Job dans un autre. Il deviendra alors un sous-job. S'il est déplacé dans un Job possédant déjà des sous-jobs, il sera positionné en dernier.

## Move as Main Job

Permet d'extraire un sous-job pour en faire un job principal. Il vous sera demandé de lui définir un mot de passe.

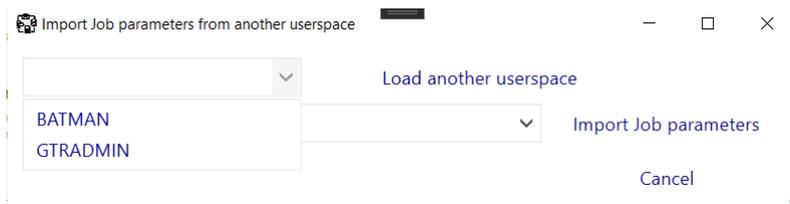
## Move Down/Up

Permet de déplacer un sous-job pour redéfinir l'ordre d'exécution d'un Job « multi-pas »

**Note :** Si vous déplacez un Job pour en faire un sous-Job, et que des planifications de lancement automatique lui sont associées, vous serez alerté et l'application vous proposera de supprimer cette(ces) planification(s) ou de les maintenir.

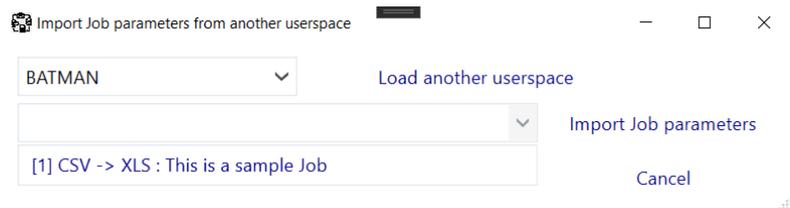
## Import External Job Parameters

On peut importer ici un job d'un autre utilisateur (à charge de connaître le mot de passe)



S'affiche la liste des utilisateurs de l'application.

...puis la liste des Jobs qui lui sont associés :



Ces informations sont lues dans le fichier de configuration de Fuzible. Vous pouvez également en charger un autre « Load Another Database file » pour récupérer les Jobs issus d'autres environnements.

Note : l'import de Jobs « multi-pas » n'est pas possible. Ils sont affichés comme ses Jobs normaux, à la différence qu'ils apparaissent en noir et non pas en bleu.

Après avoir saisi le mot de passe, le job est importé (remplissage des champs de paramétrage) et les chaînes de connexion qui y sont associées, importées automatiquement si toutefois vous ne les possédez pas dans votre liste.

Dans le cas où un ID de chaîne de connexion est mentionné dans les requêtes du Job (requêtes jointes, multi-cible), celles-ci sont également importées et transcodées automatiquement. Vous n'avez rien à faire pour reparamétrer le Job.

Après import, il faut alors cliquer sur « Save configuration » pour créer un nouveau job, basé sur ces nouveaux paramètres.

## Load another Userspace (read-only)

On peut charger toute la configuration d'un autre utilisateur pour par exemple, lancer l'un de ses jobs.

Il est en revanche impossible de modifier le moindre paramètre de sa configuration et de ses jobs. Seul le champ des paramètres dynamiques est modifiable. Ça vous permet de bénéficier du paramétrage dynamique, fonction qui peut s'avérer nécessaire.

Vous pourrez également modifier la planification de ses Jobs, fonction cruciale car si vous devez d'urgence intervenir sur la planification et que le propriétaire du compte n'est pas présent, vous devrez être en mesure de modifier le paramétrage pour lui.

# Création de Jobs

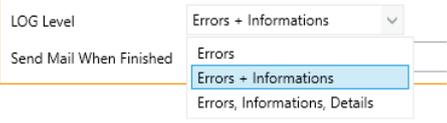
## Onglet Job Configuration

Il est temps de créer des Jobs.

Configurez ici les caractéristiques générales de votre nouveau Job.

### Options

Job description	Champ de texte qui vous permet de décrire l'usage du Job. Ce champ sert également d'objet pour un Job dont la Cible est une boîte mail.
Job Type	Data Replication (copie) ou Data Synchronization (synchro d'environnements) → Voir notion « Job Type »
Dynamic Parameters	Valeur des variables dynamiques : permet de variabiliser le comportement du job. Dynamic Parameters <code>%YYYY%MM&lt;1M,[4]</code> Chaque variable est séparée par un « ; ». Jusqu'à 9 variables supportées → Voir notion « Script language »
Visible in client app	Indique si l'on souhaite que le Job soit déclenchable par une tierce personne via l'application client fournie
Abort next steps on errors	Sur un job multi-steps, permet ou pas de continuer de dérouler les étapes si des erreurs ont été détectées durant l'exécution
Bypass post-commands if job has errors	Permet de ne pas jouer les commandes post-job si celui-ci a rencontré des erreurs → Voir notion « post-commands »
LOG level	Niveau de LOG souhaité

	 <p>- ERRORS : Vous n'aurez que les messages informatifs signifiant des erreurs et avertissements durant le déroulement d'un Job</p> <p>- ERRORS + INFORMATIONS : Vous aurez en plus les messages informatifs qui indiquent l'état d'avancement : les étapes, et ce que fait le programme</p> <p>- ERRORS, INFORMATIONS, DETAILS : Vous aurez le détail de tout ce qui se passe en interne dans le programme, des calculs d'avancement, des informations relative à l'analyse des données...</p>
Log in SQL	Si la chaîne de connexion SQL pour le LOG est paramétrée (Configuration du programme), permet de définir ici si on souhaite que ce Job fasse l'objet d'un LOG en SQL ou si celui-ci ne doit exister qu'au format « FICHIER »
Send mail when finished	Si on rentre des adresses mail ici (séparées par une virgule), un compte-rendu du job sera envoyé à ces personnes (compte-rendu contenant le rapport d'exécution, ainsi que les fichiers de LOG
Command Line	Code qui permet de lancer le job en ligne de commande MSDOS <p>Command Line <input \"%dd%mm%yyyy\""="" \"88burgfn+vn8ovvp0qcebq='=\"' \"[34]\"="" guizm\"="" type="text" value="Fuzible.exe \"/></p> <p>Arguments :</p> <ul style="list-style-type: none"> <li>- Utilisateur</li> <li>- Numéro du Job</li> <li>- Mot de passe du Job</li> <li>- Arguments dynamiques (ceux qu'on voit dans le champ « Dynamic Parameters »)</li> </ul>

## Job Type

### Data Replication

Main Parameters

Job Type

Will copy a source data into a target without any comparison

Dynamic Parameters

You can write any text or any available command. Each parameter must be separated by a semicolon (;)  
 You can use them anywhere (queries, text fields, connections) by referencing them like this: {?1}, {?2}...

[View Job With Replaced Values](#)

Ce mode est disponible pour tout type de cible. Il consiste à simplement copier des données d'un point A à un point B sans se soucier du contenu du point B.

### Data Synchronisation

Main Parameters

Job Type    Historize UPDATED and DELETED rows

Will compare source and target data, and update target according to

Dynamic Parameters

You can write any text or any available command. Each parameter must be separated by a semicolon (;)  
 You can use them anywhere (queries, text fields, connections) by referencing them like this: {?1}, {?2}...

[View Job With Replaced Values](#)

Ce mode est disponible uniquement dans le cas où la cible est un fichier, une base de données ou un domaine AD.

Il permet de comparer ce qui est envoyé du point A à ce qui existe déjà au point B. Pour ce faire, Fuzible va transcoder dynamiquement la requête source pour qu'elle soit compréhensible par la cible (partant du postulat que la source et la cible peuvent être ou différentes ou/et partager des noms de colonnes différents)

La comparaison se base sur la recherche de clé unique.

Fuzible utilise plusieurs méthodes pour trouver une clé primaire et procède du plus simple au plus complexe :

### **En mode SQL**

- 1/ Recherche de la clé primaire par des requêtes sur le schéma BDD (d'abord cible, puis source)
- 2/ Recherche de clés uniques par des requêtes sur le schéma BDD (d'abord cible, puis source)
- 3/ Détection automatique par analyse de l'ensemble de toutes les combinaisons de colonnes et de valeurs qui existent dans la source (cette méthode peut prendre énormément de temps si la quantité de lignes est important)

### **En mode FICHER**

C'est la méthode d'auto-détection qui est utilisée.

### **En mode ACTIVE DIRECTORY**

Le champ faisant office de clé se définit dans l'onglet « Cible »

On peut alors définir le comportement à adopter sur la cible :

- Insérer les nouvelles lignes présentes dans la source mais absentes dans la cible
- Mise à jour des lignes déjà existantes dans la cible, mais dont le contenu n'est plus le même dans la source
- Suppression des lignes qui sont dans la cible, mais plus dans la source
- Mise en « TAG » : si des données doivent être supprimées dans la cible (plus présentes dans la source) on peut décider de les garder mais de créer un champ « SYNCHRO\_TAG » prenant la valeur « D » (comme deleted) sur les lignes qui auraient dû être effacées

De la même manière, les lignes à mettre à jour peuvent être taguées « U » (comme updated) – elles seront toutefois mises à jour.

*L'option « Historize Updated and Deleted Rows » crée ou remplit une table à côté de la table cible (avec l'extension \_hist) qui contient toutes les lignes qui ont été soit supprimées, soit mises à jour au fil des synchros.*

*En substance, une nouvelle table est créée, utilisant le même schéma que l'originale, Fuzible lui ajoute simplement une colonne d'index complémentaire, ainsi qu'une date d'historisation.*

## Script Langage

Tous les champs « texte » de Fuzible peuvent contenir du langage de script qui sera interprété et remplacé par les valeurs associées.

On peut ainsi rendre dynamique :

- Une chaîne de connexion (par exemple, changer le serveur, l'utilisateur, la base de données, le chemin d'un fichier...)
- Les requêtes (par exemple, rendre des filtres dynamiques)
- Ajouter des colonnes additionnelles et personnalisées à un contenu

Par conception, toute zone « texte » de Fuzible comprend et interprète le langage de script. Les zones de script s'encadrent par des accolades.

Voici les variables qu'il est possible d'utiliser :

- ?1, ?2, ?3... : paramètres issus du champ "Dynamic Parameters"
- %MM : numéro de mois sur 2 caractères
- %YY : numéro d'année sur 2 caractères
- %YYYY : numéro d'année sur 4 caractères
- %DD : numéro de jour du mois sur 2 caractères
- %WW : numéro de semaine sur 2 caractères
- %HH : heure du jour sur 2 caractères
- %mm : minute sur 2 caractères
- %SS : seconde sur 2 caractères
- %DTTS / DTSMILLI : date actuelle au format Unix Timestamp
- <XM : retire X mois à MM (si existant) (ex : %MM<2M)
- <XY : retire X année à YY (ou YYYY) (si existant) (ex : %YY<2Y)
- <XD : retire X jours à DD (si existant) (ex : %DD<2D)
- <XW : retire X semaines à WW (si existant) (ex : %WW<2W)
- >XM : ajoute X mois à MM (si existant) (ex : %MM>2M)
- >XY : ajoute X année à YY (ou YYYY) (si existant) (ex : %YY>2Y)
- >XD : ajoute X jours à DD (si existant) (ex : %DD>2D)
- >XW : ajoute X semaines à WW (si existant) (ex : %WW>2W)
- %USER : le nom de l'utilisateur connecté à l'application
- %QUERYTARGETNAME : Le nom de la cible.
- %-CT1, %CS1, %CT2, %CS2... : si spécifié dans la liste des paramètres dynamique, ce paramètre va se remplacer par la valeur « retour » d'une pré-commande « source » (%CS1) ou une pré-commande « target » (%CT1)

Exemples (base : 20 mars 2018, considérons que ?1 = 100 et 2 = 'TEST')

SELECT * FROM MONFICHER_{ %MM%YYYY}.csv	SELECT * FROM MONFICHER_032018.csv
SELECT * FROM MONFICHER{%MM_%DD_%YYYY}.csv	SELECT * FROM MONFICHER03_20_2018.csv
SELECT * FROM MONFICHER_{%MM%YYYY>3M}.csv	SELECT * FROM MONFICHER_062018.csv
SELECT * FROM MONFICHER_{%MM%YYYY>3M>1Y}.csv	SELECT * FROM MONFICHER_062019.csv
SELECT * FROM MONFICHER_{%DD<10D}.csv	SELECT * FROM MONFICHER_10.csv
SELECT * FROM MATABLE WHERE ID_TEST={?1}	SELECT * FROM MATABLE WHERE ID_TEST=100
SELECT * FROM MATABLE WHERE ID_TEST={%YYYY01<2Y_?1-?2}	SELECT * FROM MATABLE WHERE ID_TEST= 201601_100-'TEST'

Notes additionnelles sur l'utilisation du langage de script :

- Dans le champ d'interface prévu pour recevoir les paramètres dynamiques, chaque paramètre doit être séparé par un « ; », ce caractère réservé à la séparation ne peut donc pas être utilisé dans une valeur dynamique !
- Les « codes » de date peuvent être utilisés comme paramètres dynamiques :  
Si vous affectez à { ?1 } la valeur « MM », { ?1 } va se remplacer par 03 (mois de mars si on est en mars)  
Ainsi, vous pouvez programmer des choses plus complexes, par exemple (nous sommes le 20 mars 2018) :  
{%YYYY01<2Y-?1} deviendra **201601-20TEST** si on ?1 a été défini avec la valeur %DDTEST

## Résumé du Job

En plus de ce paramétrage, lorsqu'un Job a été créé, vous avez accès à quelques informations sommaires (en haut de l'écran) : Date de création, dernière exécution, et état de la dernière exécution.

Job Description

Creation Date : 19/12/2020 09:28:54  
Last Modified : 14/01/2021 09:47:38  
Last Execution : 14/01/2021 09:47:23 - RUNNING TIME : 00:00:09 - ERRORS : 1 - WARNINGS : 79

Rename Job    Change Password    Create New Job  
Delete Job    Planification

Quand vous sauvegardez un Job, il vous sera demandé de définir un mot de passe. Celui-ci permet à ce que quelqu'un ne puisse pas l'importer dans sa session sans vous demander votre avis, mais il sert également à ce que les utilisateurs de l'application « Client » puissent le lancer.

Ce mot de passe est modifiable, tout comme le nom du Job.

## Planifications

Si l'application « Service » est configurée correctement, vous pouvez alors avoir accès au planificateur de Jobs. Celui-ci a pour fonction de lancer les Jobs à des dates/intervalles prédéfinis.

En ouvrant un Job, vous cliquez sur « Planification », un nouvel écran s'ouvre alors et vous propose de saisir, supprimer, modifier vos planifications pour ce Job.

La création d'un modèle de planification fonctionne en 2 modes :

### - La gestion sous forme de « jours de la semaine »

Minutes    Hours    Days    Weeks    Months

00	01 AM	Sunday	Week 02	January
05	02 AM	Monday	Week 03	February
10	03 AM	Tuesday	Week 04	March
15	04 AM	Wednesday	Week 05	April
20	05 AM	Thursday		May
25	06 AM	Friday	<input checked="" type="radio"/> Day of Week Model	June

Dynamic parameters

### - La gestion sous forme de « jours du mois »

Minutes    Hours    Days    Weeks    Months

00	01 AM	Day 01		January
05	02 AM	Day 02		February
10	03 AM	Day 03		March
15	04 AM	Day 04		April
20	05 AM	Day 05		May
25	06 AM	Day 06	<input type="radio"/> Day of Week Model	June

Dynamic parameters

Vous devrez nécessairement choisir un élément de chaque colonne pour réaliser une planification.

- Si par exemple, vous voulez lancer un Job le premier lundi de chaque mois, à 07h00, vous allez définir la planification ainsi :

Description Premier lundi de chaque mois, 7h

Minutes	Hours	Days	Weeks	Months
00	04 AM	Sunday	Week 01	janvier
05	05 AM	Monday	Week 02	février
10	06 AM	Tuesday	Week 03	mars
15	07 AM	Wednesday	Week 04	avril
20	08 AM	Thursday	● Day of Week	mai
25	09 AM	Friday	■ ■ ■ ■ ■	juin

- Si vous voulez lancer un Job toutes les 2 heures, passées de 15 minutes, le 20<sup>ème</sup> jour des 3 premiers mois de l'année, vous allez définir la planification ainsi :

Description TT les 2 heures, passées de 15 minutes, le 20e jour des 3 premiers mois de l'année

Minutes	Hours	Days	Weeks	Months
00	02 AM	Day 18		janvier
05	03 AM	Day 19		février
10	04 AM	Day 20		mars
15	05 AM	Day 21		avril
20	06 AM	Day 22	● Day of Week	mai
25	07 AM	Day 23	■ ■ ■ ■ ■	juin

La planification doit comporter une description, et éventuellement des paramètres optionnels : Ceux-ci sont, par défaut, reportés à partir du Job mais vous pourriez par exemple définir de multiples planifications pour un même job avec un paramétrage différent à chaque fois (exemple : lancer une réplication de données sur une BDD de préprod à 07h00, puis lancer cette même réplication sur une BDD de prod, à 08h00)

Orchestration

Set Planification Model

Description Sunday, connections 12 and 1

Minutes	Hours	Days	Weeks	Months
00	01 AM	Day 01		January
05	02 AM	Day 02		February
10	03 AM	Day 03		March
15	04 AM	Day 04		April
20	05 AM	Day 05	● Day of Week Model	May
25	06 AM	Day 06	■ ■ ■ ■ ■	June

Dynamic parameters [12];[1]

Sunday, connections 12 and 1 ● Active Save planification Delete planification

(new)

- Sunday, connections 12 and 1
- Monday, connections 14 and 2

Bien entendu, vous pouvez éditer ou supprimer chacune des planifications définies. Vous pouvez également choisir de les rendre actives ou non.



# Onglet Source

Ici, vous choisissez la source de connexion. Dans le cas d'une base de données, vous pouvez choisir d'afficher la liste des BDD disponibles et d'en utiliser une qui soit différente de celle qui est dans la chaîne de connexion.

Try Connection	<p>Permet de vérifier le bon fonctionnement de la connexion</p> <ul style="list-style-type: none"> <li>- BDD : teste à la connexion à l'instance SQL et ramène la liste des bases de données qu'elle contient</li> <li>- FICHER : teste l'existence du répertoire interrogé (ou (s)ftp) et affiche la quantité de fichiers présents</li> <li>- WEBSERVICE/MAIL : ping du serveur interrogé</li> <li>- ACTIVE DIRECTORY : tester l'accès au domaine</li> </ul>
Parallel Queries Execution	<p>Si le Job possède plusieurs requêtes, vous pouvez choisir d'en exécuter plusieurs en parallèle. Attention à la consommation de ressources associée à ces exécutions simultanées.</p>
Intelligent Data Analyzer	<p>Il s'agit ici du moteur d'analyse des données, grâce à lui, Fuzible est par exemple capable, entre autres, de créer automatiquement des tables SQL qui n'existent pas en mettant les types de champs les plus « justes », résultant de l'analyse des données.</p> <p>Si vous traitez de grandes masses de données, le moteur d'analyse peut s'avérer consommateur de ressources, et il n'est pas toujours utile d'analyser toutes les données. Vous pouvez donc définir, soit de scanner toutes les lignes, soit toutes les X lignes</p> <p>➔ Voir notion « Data Analyzer »</p>
Data Transformation	<p>Fuzible peut transformer les données source selon quelques scénarios prédéfinis</p> <p>➔ Voir notion « Data Transformation »</p>

## BDD

### Driver Parameters

- DataReader Mode (slower but uses less RAM. Can also avoid errors with some ODBC drivers)
- Direct Stream Copy (parallelized read & write data stream)
- Remove data from Source after having been inserted in Target

Will only be executed for queries with 1 table and will be bypassed in case of errors during Job execution.

DataReader Mode	C'est une méthode alternative de récupération des données, particulièrement utile si la Source utilise un pilote ODBC capricieux. Par ailleurs, cette méthode consomme moins de ressources mémoire.
Direct Stream Copy	En mode DataReader, les données sont lues ligne à ligne, ce qui signifie qu'elles peuvent être transférées à la Cible au fil de l'eau, par « morceaux » (batches de 100 lignes, se paramètre dans la configuration de Fuzible / Onglet SQL). Cette méthode présente l'avantage de pouvoir paralléliser la lecture et l'écriture des données, de consommer peu de RAM, et bien souvent, d'obtenir des performances accrues quand il s'agit de traiter des tables très lourdes. L'inconvénient se situe plutôt dans le cas où la connexion Cible est une BDD, et que la table Cible doit être créée par Fuzible : travailler avec un petit échantillon de données ne permettra peut-être pas de créer un schéma de données pertinent. Il est donc conseillé d'utiliser cette fonctionnalité avec une table Cible ayant déjà été créée.
Remove Data from Source after having been inserted in Target	Si vous voulez simplement transférer les données Source vers la Cible, cette option permet de supprimer dans la Source les données qui ont été récupérées. Si toutefois le Job contient des erreurs à l'exécution, cette étape sera évitée. Par ailleurs, si la requête Source contient plusieurs tables (jointures), elle sera également évitée.

## MONGODB

### Driver Parameters

- Remove internal MongoDB ID column from retrieved data

Remove internal MongoDB ID column from retrieved data	Chaque collection MongoDB identifie ses enregistrements par un ID. Celui-ci est récupéré en tant que colonne additionnelle lorsque la BDD est interrogée. Cette option permet de ne pas récupérer cette colonne
---	---

## Fichier CSV

### Driver Parameters

Are the files to be processed zipped? (Enter the ZIP filename)   Post-process (Source Files)  ▾

Raw Output (won't create a dataset, will only extract data in a single column)

Row Offset - Read files starting at row :

Read Multiple Files At Once - Name has to contain :  ?

Source File(s) are zipped in	Si les fichiers à requêter sont dans un fichier ZIP, on définit ici ce fichier (son nom peut être ou contenir des variables dynamiques)
Post-Process	On choisit ce qu'on veut faire des fichiers source une fois traités : 1 – Nothing : On les laisse où ils sont 2 – Move : On les déplace dans un sous-répertoire 3 – Zip : On les compacte dans un fichier ZIP 4- Delete : on les supprime Ces 4 options s'appliquent également dans le cas d'une récupération réseau et (S)FTP
Raw Output	Au lieu de créer un tableau de données du contenu du fichier, celui-ci est simplement extrait « brut », tel que présenté si vous l'ouvrez dans le bloc-notes par exemple
Row offset	Indique à Fuzible à partir de quelle ligne il doit lire le fichier ➔ Voir notion « détection d'entête »
Read multiple Files at once	Permet d'indiquer un "pattern" du nom de fichier : Fuzible récupèrera alors toutes les données de tous les fichiers possédant ce pattern

## Fichier Excel

### Driver Parameters

Are the files to be processed zipped? (Enter the ZIP filename)  [Browse](#) Post-process (Source Files)

Raw Output (won't create a dataset, will only extract data in a single column)

Sheet to Read (0=all)

Row Offset - Read files starting at row :

Password

Source File(s) are zipped in	Si les fichiers à requêter sont dans un fichier ZIP, on définit ici ce fichier (son nom peut être ou contenir des variables dynamiques)
Post-Process	On choisit ce qu'on veut faire des fichiers source une fois traités : 1 – Nothing : On les laisse ou ils sont 2 – Move : On les déplace dans un sous-répertoire 3 – Zip : On les compacte dans un fichier ZIP 4- Delete : on les supprime Ces 4 options s'appliquent également dans le cas d'une récupération réseau et (S)FTP
Raw Output	Au lieu de créer un tableau de données du contenu du fichier, celui-ci est simplement extrait « brut », tel que présenté si vous l'ouvrez dans le bloc-notes par exemple
Row offset	Indique à Fuzible à partir de quelle ligne il doit lire le fichier ➔ Voir notion « détection d'entête »
Sheet to read	Indique à Fuzible le numéro de la ou les feuilles qu'il doit récupérer
Password	Si le fichier Excel est protégé par mot de passe, c'est ici qu'il convient de l'indiquer. En revanche, si vous essayer d'écrire plusieurs requêtes faisant appel à plusieurs fichiers Excel, lesquels n'ont pas tous le même mot de passe, vous serez obligés de créer plusieurs Jobs.

## Fichier XML et JSON

### Source Driver

XML File  [Browse](#) Parallel Queries Execution  ?

### Driver Parameters

Are the files to be processed zipped? (Enter the ZIP filename)  [Browse](#) Post-process (Source Files)

Raw Output (will keep data as it was retrieved)

Source File(s) are zipped in	Si les fichiers à requêter sont dans un fichier ZIP, on définit ici ce fichier (son nom peut être ou contenir des variables dynamiques)
Post-Process	On choisit ce qu'on veut faire des fichiers source une fois traités : 1 – Nothing : On les laisse ou ils sont 2 – Move : On les déplace dans un sous-répertoire 3 – Zip : On les compacte dans un fichier ZIP 4- Delete : on les supprime Ces 4 options s'appliquent également dans le cas d'une récupération réseau et (S)FTP
Raw Output	Au lieu de créer un tableau de données du contenu du fichier, celui-ci est simplement extrait « brut », tel que présenté si vous l'ouvrez dans le bloc-notes par exemple

### Note sur la détection d'entête

Le mode par défaut (« Raw Output » décoché), Fuzible analyse de lui-même le contenu des fichiers CSV et EXCEL et détermine de lui-même la présence d'un entête. Son analyse est basée sur un ensemble de tests qui le rendent fiable dans 99% des cas.

## Webservice REST

### Driver Parameters

Raw Output (won't create a dataset, will only extract data in a single column)

SQL Language

Query Method

Body Sent As

API Endpoint (Nuxeo only)

SQL Language	Certaines API peuvent utiliser un simili-langage SQL qu'il est possible d'utiliser en lieu et place de « Fuzible SQL », qui est le moteur par défaut. Par exemple, le CRM Salesforce utilise SoQL
Method	Méthode de récupération des données : GET ou POST
Body Sent As	Vous pouvez envoyer du corps dans vos requêtes vers une API. Vous précisez ici de quel type de corps il s'agit : Form-Data, JSON, XML
Raw Output	Les réponses renvoyées par l'API sont extraites totalement brutes, leur contenu n'est pas réorganisé sous forme de tableau(x) de données.

Constatez que selon le « template » de connexion choisi (configuration des connexions), vous pouvez avoir un paramétrage un peu différent ; par exemple, si vous utilisez le « template » Salesforce SoQL, les paramètres sont automatiquement réglés pour vous :

### Source Driver

Parallel Queries Execution

[?](#)

### Driver Parameters

SQL Language

Raw Output (will keep data as it was retrieved)

## Mailbox

### Driver Parameters

Get Unread Messages Only (IMAP)

Post-Operation

Limit to retrieve

Get unread messages only	Choisit de ne récupérer que les mails qui n'ont pas encore été lus (IMAP uniquement)
Post-Operation	Définit ce qu'on fait du mail après l'avoir récupéré : <ul style="list-style-type: none"> <li>- Le laisser tel quel</li> <li>- Le déplacer dans un dossier « Fuzible »</li> <li>- Le supprimer</li> <li>- Le marquer comme lu</li> </ul> <p>Note : le protocole POP ne supporte que la suppression.</p>
Limit to retrieve	Par défaut, le programme récupère tous les messages qu'il trouve dans la boîte de réception, ce qui peut s'avérer extrêmement long. Il est préférable de définir une limite ici si toutefois on sait que la boîte interrogée possède un grand nombre de mails

## Active Directory

Driver Parameters

Search Scope

Base

Search scope

Profondeur de recherche des objets AD (base, one level, all levels)

## Data Analyzer

### Intelligent Data Analyzer

Data Scanning Level

Analyze each row

Analyze each row

Analyze every 5 rows

Analyze every 10 rows

Analyze every 20 rows

Advanced Parameters

Fuzible peut être amené à analyser les données qu'il collecte de façon à en deviner le type le plus juste. Cette fonctionnalité est particulièrement utile dans 2 cas :

- La cible est une BDD et la ou les tables CIBLES n'existent pas ou sont mal définies : Le programme peut soit les créer, soit les « améliorer » si demandé (changer les types de champs)
- Mode Synchronisation : Pour permettre au mieux de comparer au mieux les données CIBLE et SOURCE qui ont parfois des provenances très différentes, l'analyse permet de mieux cibler les modifications apportées à une valeur
- Filtrage et agrégation de données issues de Sources autres qu'une BDD

A ce jour, Fuzible est capable de créer et/ou transcoder les champs les plus standards : CHAR, NVARCHAR, VARCHAR, TEXT, DATE, DATETIME, TIMESTAMP, INTEGER, DECIMAL(X,X), FLOAT, BIGINT, SMALLINT, BIT. Tout autre type de champ sera vu comme un champ de type « Texte »

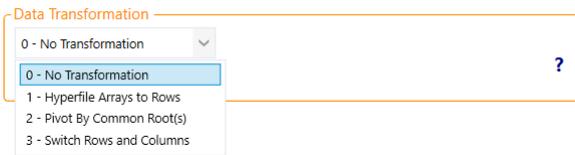
Plus la précision demandée par l'analyseur est faible (scan toutes les X lignes), plus il pondèrera son analyse pour assurer que l'import se fasse correctement.

*Exemple : un champ identifié VARCHAR(5) sera créé en VARCHAR(10) dans la cible si le mode d'analyse ne scanne que toutes les 50 lignes)*

A noter : pour toutes les requêtes de – de 100 000 lignes, le programme scanne toutes les lignes, peu importe le paramètre choisi.

## Data Transformation

On peut transformer totalement les données collectées avant de les intégrer dans une Cible :



### Hyperfile Arrays Transformation :



C'est un type de transformation propre aux bases de données Hyperfile : certains champs de ce SGBD sont en fait des tableaux, et lorsque ces bases sont requêtées par le driver ODBC, celui-ci produit autant de colonnes qu'il y a de colonnes dans ces champs de type « tableau » (exemple : PPRUBP\_01, PPRUBP\_02, PPRUBP\_03...) La transformation analyse et compte automatiquement toutes les colonnes de ce type (elles ont toujours comme dénomination le nom du champ, suivi de \_XX) et les renverse pour qu'il n'y en ait qu'une, et que les données soient ainsi transformées en lignes et non plus en colonnes.

*Exemple : Si vous avez une colonne de type tableau de 50 colonnes dans HYPERFILE, l'ODBC vous envoie 50 colonnes différentes ! L'outil de transformation n'en conserve qu'une pour privilégier l'affichage en ligne de ces données (soit 50 lignes pour une seule à l'origine)*

**Exemple :** à droite, la source initiale, à gauche, le résultat produit par la transformation :

1	id_sample	data_01	data_02	data_03	data_04	li_sample
2	1	hello1	hello2	hello3	hello4	firstrow
3	2	world1	world2	world3	world4	secondrow
4	3	wolf1	wolf2	wolf3	wolf4	thirdrow
5						

id_sample	data	li_sample	IDX_COL
1	hello1	firstrow	1
1	hello2	firstrow	2
1	hello3	firstrow	3
1	hello4	firstrow	4
2	world1	secondrow	1
2	world2	secondrow	2
2	world3	secondrow	3
2	world4	secondrow	4
3	wolf1	thirdrow	1
3	wolf2	thirdrow	2
3	wolf3	thirdrow	3
3	wolf4	thirdrow	4

Comme on peut le constater, les colonnes data\_01, data\_02,... suivent une logique de « libelle + nombre », ce qui permet de prendre « data » comme libellé, et d'afficher le nombre dans « IDX\_COL »

## Pivot by Common Root(s) :

Data Transformation

2 - Pivot By Common Root(s) ▾

\_

Avoid Transformation if Variable Arrays Sizes are Detected

Ce type de transformation permet d'utiliser une racine de colonne commune pour renverser les données. Par exemple, si vous mettez « x », le programme renversera toutes les colonnes qui commencent par « x ». A cet effet, la source se verra augmentée de 3 colonnes :

- Une colonne « x », contenant la valeur
- Une colonne « x\_lbl », contenant le reste du nom de la colonne (ex : colonne « x\_001 » -> 001)
- Une colonne « x\_idx » contenant l'index de la ligne

Si 7 colonnes avec la racine « x » ont été détectées, 7 lignes seront alors produites. Ces colonnes seront bien entendu supprimées de la source et remplacées par les 3 colonnes indiquées précédemment.

**Exemple :** à droite, la source initiale, au milieu, la requête, à gauche, le résultat produit par la transformation :

Data Transformation

2 - Pivot By Common Root(s) ▾

split

```
1 id_sample;dataW;dataX;dataY;dataZ;li_sample
2 1;hello1;hello2;hello3;hello4;firstrow
3 2;world1;world2;world3;world4;secondrow
4 3;wolf1;wolf2;wolf3;wolf4;thirdrow
5
```

```
TRANSFORMED.CSV:select id_sample, li_sample,
dataW as split01,
dataX as split02,
dataY as split03,
dataZ as split04
from TRANSFORM_CR.CSV
```

id_sample	li_sample	split	split_lbl	split_idx
1	firstrow	hello1	01	1
1	firstrow	hello2	02	2
1	firstrow	hello3	03	3
1	firstrow	hello4	04	4
2	secondrow	world1	01	1
2	secondrow	world2	02	2
2	secondrow	world3	03	3
2	secondrow	world4	04	4
3	thirdrow	wolf1	01	1
3	thirdrow	wolf2	02	2
3	thirdrow	wolf3	03	3
3	thirdrow	wolf4	04	4

On voit que les colonnes « W,X,Y,Z » ont été volontairement renommées dans la requête avec une racine commune, de façon à transformer cette racine en 3 colonnes :

- « split » : le nom de la racine
- « split\_lbl » : la donnée qui était contenue dans la colonne
- « split\_idx » : l'index de colonne original

## Switch Rows and Columns :

Data Transformation

3 - Switch Rows And Columns

Add a Column With Label (PROPERTIES) ?

Reverse "simplement" les colonnes en lignes et vice-versa.

L'option « Add a column with label (PROPERTIES) » permet d'ajouter une colonne avec le nom original de la colonne d'origine associée à la donnée renversée

Exemple : à droite, la source initiale, au milieu, la requête, à gauche, le résultat produit par la transformation :

```
1 id_sample;dataW;dataX;dataY;dataZ;li_sample
2 1;hello1;hello2;hello3;hello4;firstrow
3 2;world1;world2;world3;world4;secondrow
4 3;wolf1;wolf2;wolf3;wolf4;thirdrow
5
```

```
TRANSFORMED.CSV:select id_sample, dataW, dataX
from TRANSFORM_RC.CSV
```

R_001	R_002	R_003
1	2	3
hello1	world1	wolf1
hello2	world2	wolf2

...et avec l'option « PROPERTIES »

Data Transformation

3 - Switch Rows And Columns

Add a Column With Label (PROPERTIES) ?

```
1 id_sample;dataW;dataX;dataY;dataZ;li_sample
2 1;hello1;hello2;hello3;hello4;firstrow
3 2;world1;world2;world3;world4;secondrow
4 3;wolf1;wolf2;wolf3;wolf4;thirdrow
5
```

```
TRANSFORMED.CSV:select id_sample, dataW, dataX
from TRANSFORM_RC.CSV
```

PROPERTIES	R_001	R_002	R_003
id_sample	1	2	3
dataW	hello1	world1	wolf1
dataX	hello2	world2	wolf2

## Si existantes, transformer aussi les requêtes croisées

Si l'une des requêtes contient des requêtes croisées (données venant d'autres sources), la transformation ne s'opérera que sur les données de la requête principale : toutes les données issues des requêtes croisées seront jointes aux données principales sans transformation.

Dans le cas inverse, tous les jeux de données collectés sont transformés, puis croisés selon le mode de croisement déterminé dans la requête.

Data Transformation

2 - Pivot By Common Root(s)

?

Avoid Transformation if Variable Arrays are Detected

If Any, also Transform Cross-Queries

## Pre and Post-Job Commands

Valables dans l'onglet « Source » et l'onglet « Cible », pour les connexions de type « Base de Données » ou « Fichier ». Cette fonctionnalité répond à 2 besoins identifiés :

- Parfois, la simple copie de données ne suffit pas, on a besoin de lancer quelque chose avant ou après cette copie, et on veut éviter de devoir programmer ces scénarios dans un logiciel d'orchestration
- On veut récupérer une donnée avant de démarre le Job afin d'exploiter celle-ci (conditionner le comportement du job en fonction de cette donnée, par exemple)

Cette fonctionnalité permet donc de lancer, avant ou après l'exécution du Job, une ou plusieurs commandes.

### Fichier

Pre/Post Job Command(s) : any CMD raw command (ex : DATE /T)

Perform Post-Job Commands ▼  ?

N'importe quelle commande que vous pourriez lancer via l'interpréteur de commandes Windows est supportée. Si plusieurs commandes doivent être lancées, elles doivent être séparées par un « ; »

Ex : c:\Tools\macommande.bat

Ex2 : DATE /T (renvoie la date du jour)

Fuzible ajoute à son LOG toutes les infos ou erreurs que l'exécution de ces commandes externes pourraient générer.

### BDD

Pre/Post Job Command(s) : any SQL raw command (ex : SELECT MAX(field) FROM table)

Perform Post-Job Commands ▼  ?

N'importe quelle commande SQL compatible avec le SGBD sélectionné. L'exécution d'une procédure stockée, un UPDATE... Si plusieurs commandes doivent être lancées, elles doivent être séparées par un « ; »

Ex : UPDATE myTable SET envoye = 1 WHERE mois = {%MM}

Ex2 : EXECUTE myProcedure('1')

Dans l'exemple 1, vous constatez, pour rappel, que les paramètres dynamiques du Job peuvent être utilisés.

Ces commandes peuvent renvoyer une valeur. Fuzible peut exploiter ces valeurs en tant que paramètres dynamiques. Par exemple, si j'écris dans les paramètres dynamiques du Job %CS1, cela signifie que ce paramètre dynamique sera remplacé par la première valeur de la première commande de la connexion « Source » :

C = Command

S = Source

1 = Commande N°1

On peut donc écrire également %CT1 (Command Target N°1), ou encore %CT2...

## Boucles des commandes Pré-Job

Vous remarquerez peut-être que si une seule commande Pré-Job a été saisie, une option apparaît. « Loop Job for each Result »

Cette option permet de complexifier le scénario du Job en associant un paramètre dynamique au résultat d'une commande, qui, si celle-ci retourne plusieurs lignes, permet de boucler le Job autant de fois qu'il y a de résultats, en attribuant à chaque fois un paramètre dynamique différent.

### Explication :

Vous appelez une procédure stockée « getMails » (**CALL** est la syntaxe MySQL pour l'appel à une procédure stockée)

Pre/Post Job Command(s) : any SQL raw command or Stored Proc. (ex : SELECT MAX(field) FROM table)

Perform Pre-Job Commands   Loop Job for each result ?

... qui retourne ce jeu d'essai :

mailField	IdPeople	Comment
<a href="mailto:Leon@mymail.com">Leon@mymail.com</a>	1	Our beautiful CEO
<a href="mailto:Arthur@mymail.com">Arthur@mymail.com</a>	2	Our incredible COO
<a href="mailto:Samantha@mymail.com">Samantha@mymail.com</a>	3	Our amazing CTO

Vous pourriez avoir envie d'envoyer un mail à chacune de ces personnes, avec un mail contenant par exemple son récapitulatif d'informations.

Cette option permet de réaliser ce scénario, à condition de programmer le Job en conséquence.

En l'occurrence, il faut attribuer un ou plusieurs paramètres dynamiques aux résultats produits par l'appel à la procédure stockée (ou toute autre commande retournant un jeu de résultats).

**On peut associer par exemple une seule variable en procédant ainsi (onglet « Job Configuration ») :**

Dynamic Parameters

`%SC1;`

You can write any text or any available command. Each parameter must be separated by a semicolon (;)

You can use them anywhere (queries, text fields, connections) by referencing them like this : {?1}, {?2}...

**Ou bien plusieurs, en spécifiant le numéro de colonne (en base 1) auquel le paramètre est associé :**

Dynamic Parameters

`%SC1[1]; %SC1[2]`

You can write any text or any available command. Each parameter must be separated by a semicolon (;)

You can use them anywhere (queries, text fields, connections) by referencing them like this : {?1}, {?2}...

Vous pouvez alors programmer une requête qui, pour chaque mail retourné par la procédure stockée, enverra un mail à la personne en question, avec ses informations personnelles :

## Job Queries (MySQL SQL) Sandbox

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

```
{?1}:select * from peopleData where idPeople = {?2}
```

**{ ?1 }** -> Sera remplacé par le paramètre dynamique N°1, lequel se remplit des données de la première colonne de la procédure stockée, soit le mail de la personne.

**{ ?2 }** -> Sera remplacé par le paramètre dynamique N°2, lequel se remplit des données de la deuxième colonne de la procédure stockée, soit l'ID de la personne.

Ainsi, le Job s'exécute en boucle jusqu'à ce que le jeu de résultats de la procédure stockée appelée en Pré-Job ait été consommé.

Dans notre exemple, le Job « tournera » 3 fois, avec les valeurs suivantes

Itération	Param. Dynamique 1	Param. Dynamique 2
Première itération	{ ?1 } => %CS1[1] => <a href="mailto:Leon@mymail.com">Leon@mymail.com</a>	{ ?2 } => %CS1[2] => 1
Deuxième itération	{ ?1 } => %CS1[1] => <a href="mailto:Arthur@mymail.com">Arthur@mymail.com</a>	{ ?2 } => %CS1[2] => 2
Troisième itération	{ ?1 } => %CS1[1] => <a href="mailto:Samantha@mymail.com">Samantha@mymail.com</a>	{ ?2 } => %CS1[2] => 3

Avec cette option, il est donc possible de scénariser un peu plus votre Job, et le rendre plus dynamique.

**Restriction** : Une seule commande pré-job doit avoir été saisie pour que l'option soit disponible

Par ailleurs, si elle a été activée sur l'onglet « Source », elle ne pourra pas être activée sur l'onglet « Cible » (et inversement), quand bien même il y aurait une commande pré-Job saisie, et ce, pour éviter de rendre le Job trop confus et de tomber dans des scénarios qui nécessitent une présentation plus visuelle du comportement du Job (sous forme de diagramme, par exemple)

## Onglet Cible

Ici, vous choisissez la Cible des données. Dans le cas d'une base de données, vous pouvez choisir d'afficher la liste des BDD disponibles et d'en utiliser une qui soit différente de celle qui est dans la chaîne de connexion.

Fuzible - Data Replication & Synchronization

File Configuration Tools Help

Job Selection

GUIZM [65] BIG Data Perf Tests [Create/manage a Multi-steps Job](#) [Create new Step](#)

Job Configuration Source : CSV File **Target : SQLite Database** Queries Log Viewer

Choose a Target [17] -> Local SQLite File BIG [Edit Connection](#)

Databases [Try Connection and Get Databases](#)

Target Driver

SQLite  Trim Data (Left, Right) Parallel insertion 1 ?

Driver Parameters

Target table behavior Drop + Recreate  If non-existent, create table(s) automatically

If Target Table has to be created, try to find and add a Primary Key ? (can analyze up to 5 fields - may take a lot of time)

Allow Schema Alteration in Target (ie : when trying to insert a 10-char value in a VARCHAR(5) column)

Disable Constraints when inserting/updating/deleting data (faster but requires sufficient privileges)

Insert NULL instead of empty values

Pre/Post Job Command(s) : any SQL raw command (ex : SELECT MAX(field) FROM table)

Perform Post-Job Commands ?

[Save Job](#) [Abort Job](#)  Simulation Mode [Start Job](#)

## Options communes à toutes les cibles :

Additional Columns

Add a Row Count Column in Target : ROWNUM  Add a Column with Source DB/Path in Target : DBNAME

Add a Timestamp Column in Target : DTLOAD

Add one or more column(s) with dynamic param(s) ex : MYCOLUMN={?1} MYCOLUMN={?1}

Toutes les colonnes spéciales ont un nom par défaut qui peut être changé par l'utilisateur.

Add a row count column in target	Ajoute aux données récupérées dans la source une colonne "ROWNUM" qui est simplement un compteur de ligne
Add a timestamp column in target	Ajoute aux données récupérées dans la source une colonne « DTLOAD » qui contient la date de récupération des données
Add a column with source database/path	Ajoute aux données récupérées dans la source une colonne « DBNAME » qui contient la provenance des données
Add a column with a dynamic param	Ajoute aux données récupérées dans la source une ou plusieurs colonnes optionnelles (ex : MYCOLUMN={ ?1} : va rajouter une colonne « MYCOLUMN » et remplir ses données par le paramètre dynamique { ?1} Il est possible d'ajouter plusieurs colonnes en les séparant comme ceci : MYCOLUMN1={ ?1};MYCOLUMN2='test'

Target Driver Parallel insertion

Postgres   Trim Data (Left, Right)

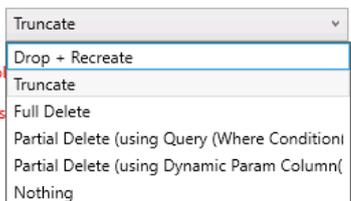
Trim data	Autorise ou non la suppression des espaces (avant/après) de toutes les données récupérées
Parallel Insertion	Option disponible uniquement en mode « CIBLE = BDD » : Elle permet de réaliser les opérations d'INSERT, UPDATE, DELETE en multi-thread : Nécessite une BDD Cible performante, surtout si vous cumulez l'insertion parallèle avec le chargement parallèle (onglet Source) !

## BDD

### Driver Parameters

Target table behavior   If non-existent, create table(s) automatically

- Bulk Insert (very fast, but less reliable if Source Data needs some transcoding to be properly inserted in Target)
- If Target Table has to be created, try to find and add a Primary Key ? (can analyze up to 5 fields - may take a lot of time)
- Allow Schema Alteration in Target (ie : when trying to insert a 10-char value in a VARCHAR(5) column)
- Disable Constraints when inserting/updating/deleting data (faster but requires sufficient privileges)
- Insert NULL instead of empty values

Target Tables Behavior	<p>Disponible en mode « Data Réplication » : Permet de définir le comportement à adopter sur la table cible lorsqu'on va la remplir</p> <p>Target Tables Behavior <input type="text" value="Truncate"/></p> <ul style="list-style-type: none"> <li><input type="radio"/> Allow Add+Change Type Columns</li> <li><input type="radio"/> Disable Constraints (requires sufficient privileges)</li> <li><input checked="" type="radio"/> Set NULL for Empty Values</li> </ul>  <p>1 – Drop + Recreate : la détruit puis la reconstruit          2 – Truncate : La vide intégralement selon la méthode « TRUNCATE »          3 – Full Delete : La vide intégralement selon la méthode « DELETE »          4 - Partial Delete using Query (Where) : va utiliser le "WHERE" de la requête source pour supprimer l'équivalent dans la cible          Ex : MYTARGET :SELECT * FROM MYSOURCE WHERE id &gt; 50          → Fuzible supprimera tous les « id &gt; 50 » dans « MYTARGET » avant d'insérer de nouvelles données          5 – Partial Delete using Dynamic Param Column : va utiliser la colonne dynamique comme champ de filtrage          Ex : Si la cible utilise une colonne dynamique MYCOLUMN={ ?1} avec { ?1} prenant comme valeur 'TEST', la requête DELETE FROM maTableCible WHERE MYCOLUMN = 'TEST' sera jouée avant l'insertion des données          6 – Nothing : Ne fait rien</p>
Bulk Insert	C'est une méthode de copie des données très rapide. Malgré le fait qu'elle offre des performances incroyables, cette méthode est parfois moins fiable que celle par défaut (les transactions), particulièrement si les données doivent être converties entre la Source et la Cible.
If non-existent, create table(s) automatically	Par défaut, si la table Cible n'existe pas, Fuzible va la créer automatiquement en utilisant le meilleur schéma de données possible. Vous pouvez empêcher ce comportement, si la table Cible n'existe pas, un message d'erreur sera inscrit dans le LOG
Allow add+change type columns in target	Autorise Fuzible, grâce à son moteur d'analyse des données, à modifier le schéma cible si besoin (modification des types de colonnes). Nécessite des privilèges importants sur la BDD cible. Dépend donc de l'utilisateur employé dans la chaîne de connexion. C'est très utile si les données Sources sont souvent changeantes et que les tables Cible doivent être adaptées en conséquence
Disable constraints	Permet de réaliser les insertions en désactivant les contraintes de clés. Nécessite des privilèges importants sur la BDD cible. Dépend donc de l'utilisateur employé dans la chaîne de connexion
Try to add primary key	Si la table cible ne possède pas de clé primaire, Fuzible peut la créer de lui-même en analysant toutes les combinaisons de colonnes et de valeurs possibles. L'analyse est limitée à 5 colonnes maximum. Si la quantité de colonnes et de données est grande, l'analyse peut durer extrêmement longtemps et il n'est pas conseillé d'utiliser cette fonctionnalité. Nécessite par ailleurs des privilèges importants sur la BDD cible. Dépend donc de l'utilisateur employé dans la chaîne de connexion
Set NULL for empty values	Toutes les données "vides" dans la Source peuvent être intégrées en tant que « NULL » dans la Cible

## Fichier CSV

### Driver Parameters

#### File Creation Behavior

Rows / created file :  Help for scripting multiple output filenames ?  If exists, append output file(s)

CSV separator :

- Add header row (using Source query fields -or aliases- names)
- Embrace values with double-quotes (ie : "Value1";"Value2";"Value3")

Rows / File	Quantité de lignes à envoyer dans le fichier cible avant d'en créer un nouveau. Si la quantité de lignes de la source dépasse cette valeur, il convient de déterminer un « pattern » pour nommer les fichiers créés afin de ne pas les écraser au fur-et-à-mesure ➔ Voir notion « Multiple files naming pattern »
Append Existing File	Si le fichier cible possède déjà des lignes, on peut décider de ne pas le vider mais plutôt le compléter
CSV separator	Permet de définir le caractère séparateur du fichier cible
Add header row	Ajout ou non d'une ligne d'entête dans le fichier produit (construit à partir des champs « Source »)
Embrace values with double quotes	Ajoute des guillemets avant et après chaque valeur (ex : "test";"125";"hello")

## Fichier Excel

### Driver Parameters

#### File Creation Behavior

Rows / created file :  Help for scripting multiple output filenames ?  If exists, append output file(s)

Set password  Visual Style :

- Add header row (using Source query fields -or aliases- names)
- Add a Title Row (using 'Job Description')

Rows / File	Quantité de lignes à envoyer dans le fichier cible avant d'en créer un nouveau. Si la quantité de lignes de la source dépasse cette valeur, il convient de déterminer un « pattern » pour nommer les fichiers créés afin de ne pas les écraser au fur-et-à-mesure ➔ Voir notion « Multiple files naming pattern »
Append Existing File	Si le fichier cible possède déjà des lignes, on peut décider de ne pas le vider mais plutôt le compléter
Set password	Définit un mot de passe sur le fichier EXCEL produit
Add header row	Ajoute ou un entête (nom des champs)
Add a title Row	Ajoute une ligne d'entête général, la valeur sera la description du Job
Style	Permet de choisir un style graphique parmi ceux qui sont proposés

## Fichier XML

### Driver Parameters

#### File Creation Behavior

Rows / created file :  Help for scripting multiple output filenames ?  If exists, append output file(s)

Header Row :  Row Tag script builder  ?

Write Mode :

- Don't create Tag for empty values
- Add CDATA for all values (ie : <![CDATA[<sender>John Smith</sender>]]>

Rows / File	Quantité de lignes à envoyer dans le fichier cible avant d'en créer un nouveau. Si la quantité de lignes de la source dépasse cette valeur, il convient de déterminer un « pattern » pour nommer les fichiers créés afin de ne pas les écraser au fur-et-à-mesure ➔ Voir notion « Multiple files naming pattern »
Append Existing File	Si le fichier cible possède déjà des lignes, on peut décider de ne pas le vider mais plutôt le compléter
Header	Choix de l'entête du fichier XML (généralement : xml version='1.0')
Write Mode	Permet de choisir la façon dont les données seront écrites dans le fichier XML : ➔ Mode 1 : Chaque ligne de la Source est encadrée par une balise (Row Tag Script Builder), et chaque champ est une sous-balise, prenant comme valeur la donnée de cette colonne dans la ligne Source. Ex : <Row><MyField>myValue</MyField><MyField2>myValue2</MyField2></Row> ➔ Mode 2 : Chaque ligne correspond à une ligne des données Source, chaque valeur est associée à un attribut (le nom de colonne). Ex : <Row MyField= "myValue" MyField2= "myValue2" />
Add CDATA tag for each value	(Mode 1 uniquement) Si la source contient des valeurs exotiques, le tag standard "CDATA" permet d'encadrer les données de façon à ce qu'un moteur d'interprétation XML comprenne que les valeurs encadrées par ce tag contiennent des caractères spéciaux
Don't create tag for empty value	Si une valeur est vide, permet de dire si on veut malgré tout générer un tag vide dans le fichier de sortie
Row Tag script builder	Permet de définir un script de comportement pour la balise de chaque ligne ➔ Voir notion « Tag Builder »

## Fichier JSON

Driver Parameters

File Creation Behavior

Rows / File

10000



Append existing file

Help for multiple files output scripting



Row Tag script builder

[JOBNAME]



Rows / File

Quantité de lignes à envoyer dans le fichier cible avant d'en créer un nouveau.

Si la quantité de lignes de la source dépasse cette valeur, il convient de déterminer un « pattern » pour nommer les fichiers créés afin de ne pas les écraser au fur-et-à-mesure

➔ Voir notion « Multiple files naming pattern »

Append Existing File

Si le fichier cible possède déjà des lignes, on peut décider de ne pas le vider mais plutôt le compléter

Row Tag script builder

Permet de définir un script de comportement pour la balise de chaque ligne

➔ Voir notion « Tag Builder »

## TAG BUILDER

Par défaut, la structure d'un fichier XML et JSON produits par Fuzible :

Job Queries Sandbox

A Query must start with the filename in which data will be written in.  
ex : MyFile.CSV:SELECT (...)

```
[3]myfile.xml[4]myfile.json:select * from sample_table_1 limit 3
```

Le fichier JSON se présentera ainsi :

```
[
  {
    "id_sample": 5,
    "li_sample": "five",
    "dt_random_date": "14/09/2020 13:10:49",
    "nb_random_number": 25,
    "li_random_string": "iidbknotshqzizjf",
    "id_group": "005",
    "id_ssgroup": "000"
  },
  {
    "id_sample": 28,
    "li_sample": "two-eight",
    "dt_random_date": "18/10/2020 13:10:49",
    "nb_random_number": 13,
    "li_random_string": "ysisszfpzpcmohdx",
    "id_group": "008",
    "id_ssgroup": "003"
  },
  {
    "id_sample": 66,
    "li_sample": "six-six",
    "dt_random_date": "15/01/2021 13:10:49",
    "nb_random_number": 56,
    "li_random_string": "inpljrzhwvkmvmfb",
    "id_group": "006",
    "id_ssgroup": "001"
  }
]
```

...et le fichier XML :

```
<?xml version='1.0'?>
<sample_table_1>
  <Row>
    <id_sample>5</id_sample>
    <li_sample>five</li_sample>
    <dt_random_date>14/09/2020 13:10:49</dt_random_date>
    <nb_random_number>25</nb_random_number>
    <li_random_string>iidbknotshqzizjf</li_random_string>
    <id_group>005</id_group>
    <id_ssgroup>000</id_ssgroup>
  </Row>
  <Row>
    <id_sample>28</id_sample>
    <li_sample>two-eight</li_sample>
    <dt_random_date>18/10/2020 13:10:49</dt_random_date>
    <nb_random_number>13</nb_random_number>
    <li_random_string>ysisszfpzpcmohdx</li_random_string>
    <id_group>008</id_group>
    <id_ssgroup>003</id_ssgroup>
  </Row>
  <Row>
    <id_sample>66</id_sample>
    <li_sample>six-six</li_sample>
    <dt_random_date>15/01/2021 13:10:49</dt_random_date>
    <nb_random_number>56</nb_random_number>
    <li_random_string>inpljrzhwvkmvmfb</li_random_string>
    <id_group>006</id_group>
    <id_ssgroup>001</id_ssgroup>
  </Row>
</sample_table_1>
```

La balise principale utilise le nom de la table d'entrée (le premier si la requête contient des jointures)

A présent, si je veux changer la balise XML principale, il me suffit de poser un alias dans ma requête :

#### Job Queries Sandbox

A Query must start with the filename in which data will be written.  
ex : MyFile.CSV/SELECT (-)

```
[3]myfile.xml[4]myfile.json:select * from sample_table_1 as sampleTag limit 3
```

“SampleTag” est l'alias qui sera utilisé dans le corps XML

```
<?xml version='1.0'?>
<sampleTag>
  <Row>
    <id_sample>5</id_sample>
    <li_sample>five</li_sample>
    <dt_random_date>14/09/2020 13:10:49</dt_random_date>
    <nb_random_number>25</nb_random_number>
    <li_random_string>iidbknotshqzjif</li_random_string>
    <id_group>005</id_group>
    <id_ssgroup>000</id_ssgroup>
  </Row>

```

#### Mots-clé supportés :

- [JOBNAME]
- [DATETIME]
- [ROWCOUNT]
- [FILECOUNT]
- [USER]
- [nImporteQuelChamp]

Je vais définir le script dans le menu "Cible" :

The screenshot shows the 'Driver Parameters' dialog box, specifically the 'File Creation Behavior' section. It contains several settings:

- Rows / File:** A text box containing '1000'.
- Append existing file:** A radio button that is currently selected.
- Header:** A text box containing 'xml version='1.0''. To its right is a radio button labeled 'Don't create Tag for empty values', which is unselected.
- Row Tag script builder:** A text box containing 'CurrentID=[id\_sample]'. To its right is a radio button labeled 'Add CDATA Tag for each value', which is unselected.
- There are two question mark icons: one next to 'Help for multiple files output scripting' and another next to the 'Row Tag script builder' text box.

Le

Le fichier JSON créé :

```
[
  {
    "CurrentID": "5",
    "sampleTag": [
      {
        "id_sample": 5,
        "li_sample": "five",
        "dt_random_date": "14/09/2020 13:10:49",
        "nb_random_number": 25,
        "li_random_string": "iidbknotshqzjif",
        "id_group": "005",
        "id_ssgroup": "000"
      },
      {
        "id_sample": 28,
        "li_sample": "two-eight",
        "dt_random_date": "18/10/2020 13:10:49",
        "nb_random_number": 13,
        "li_random_string": "ysisszfpzpcmhdx",
        "id_group": "008",
        "id_ssgroup": "003"
      },
      {
        "id_sample": 66,
        "li_sample": "six-six",
        "dt_random_date": "15/01/2021 13:10:49",
        "nb_random_number": 56,
        "li_random_string": "impljrzzhvkvvmfb",
        "id_group": "006",
        "id_ssgroup": "001"
      }
    ]
  }
]
```

...et le fichier XML :

```

<?xml version="1.0" ?>
<sampleTag>
  <CurrentID=5>
    <id_sample>5</id_sample>
    <li_sample>five</li_sample>
    <dt_random_date>14/09/2020 13:10:49</dt_random_date>
    <nb_random_number>25</nb_random_number>
    <li_random_string>iidbknotshqzjif</li_random_string>
    <id_group>005</id_group>
    <id_ssgroup>000</id_ssgroup>
  </CurrentID=5>
  <CurrentID=28>
    <id_sample>28</id_sample>
    <li_sample>two-eight</li_sample>
    <dt_random_date>18/10/2020 13:10:49</dt_random_date>
    <nb_random_number>13</nb_random_number>
    <li_random_string>ysisszfpzpcmohtdx</li_random_string>
    <id_group>008</id_group>
    <id_ssgroup>003</id_ssgroup>
  </CurrentID=28>
  <CurrentID=66>
    <id_sample>66</id_sample>
    <li_sample>six-six</li_sample>
    <dt_random_date>15/01/2021 13:10:49</dt_random_date>
    <nb_random_number>56</nb_random_number>
    <li_random_string>inpljrzhwvkvmvfb</li_random_string>
    <id_group>006</id_group>
    <id_ssgroup>001</id_ssgroup>
  </CurrentID=66>
</sampleTag>

```

### MULTIPLE FILES NAMING PATTERN

En mode CIBLE = FICHER, vous pouvez décider de découper le résultat en plusieurs fichiers à partir d'un certain nombre de lignes. Par exemple, si la source renvoie 1000 lignes, vous pouvez découper le fichier final en 1 seul fichier de 1000 lignes, ou bien 5 fichiers de 200 lignes chacun, ou bien 1000 fichiers de 1 ligne chacun.

Ce découpage multi-fichier peut être intelligemment nomenclaturé. Il est saisi dans l'onglet des requêtes (voir plus loin)

[QUERYALIAS]	Alias de la première table source de la requête <b>Ex</b> : MONFICHER_[QUERYALIAS].CSV : SELECT * FROM MATABLE AS MYQUERY <b>Donne</b> : MONFICHER_MYQUERY.CSV
[FILECOUNT]	Compteur de fichiers produits : renvoie le nombre de fichiers créés Si le programme a déjà créé 3 fichiers, il va renvoyer « 4 » au prochain fichier produit <b>Ex</b> : MONFICHER_[FILECOUNT].CSV : SELECT * FROM MATABLE <b>Donne</b> : MONFICHER_1.CSV, MONFICHER_2.CSV...
[ROWCOUNT]	Compteur de lignes extraites : renvoie le numéro de ligne de la requête source au départ du fichier Si on définit un changement de fichier toutes les 1000 lignes, à la création du 2 <sup>ème</sup> fichier, le programme va renvoyer 1001 (au 3 <sup>ème</sup> fichier, 2001) <b>Ex</b> : MONFICHER_[ROWCOUNT].CSV : SELECT * FROM MATABLE <b>Donne</b> : MONFICHER_1.CSV, MONFICHER_1001.CSV...
[COLONNE]	Renvoie la valeur d'une colonne en particulier au moment de la création du nouveau fichier Si la valeur de la colonne « ma_colonne » est « Hello » au moment de la création du nouveau fichier, puis « World » à la création du suivant, le programme va renvoyer « Hello » et ainsi de suite. <b>Ex</b> : MONFICHER_[myField].CSV : SELECT myField FROM MATABLE <b>Donne</b> : MONFICHER_Hello.CSV, MONFICHER_World.CSV...

Ce qu'il est possible de faire :

- Les utiliser plusieurs fois
- L'ordre n'a pas d'importance
- [COLONNE] peut être utilisé plusieurs fois, avec plusieurs colonnes différentes (veillez à ce que la colonne existe bien dans la source, le cas échéant, le nom de la colonne est retourné et non sa valeur !)
- Intercaler des caractères entre chaque paramètre (ex : [FILECOUNT]-\_[ROWCOUNT])

Restriction :

- Les caractères spéciaux seront automatiquement remplacés par un « \_ »

Quelques exemples :

Postulat : la source fait 2000 lignes et le découpage est demandé toutes les 1000 lignes. Filename = TEST, Le fichier de sortie est un CSV.

2 fichiers seront donc créés :

hello[ROWCOUNT]world[NOM_CLIENT].CSV : <i>SELECT ...</i>	- hello0worldFNAC.CSV - hello1001worldAUCHAN.CSV
Hello[FILECOUNT]_[ID_CLIENT] .CSV : <i>SELECT ...</i>	- Hello1_283.CSV - Hello2_81036.CSV
[QUERYTARGETNAME][ROWCOUNT][FILECOUNT] .CSV : <i>SELECT ...</i>	- TEST11.CSV - TEST10012.CSV
[QUERYTARGETNAME]%%\$£[FILECOUNT] .CSV : <i>SELECT ...</i>	- TEST__1.CSV - TEST__2.CSV

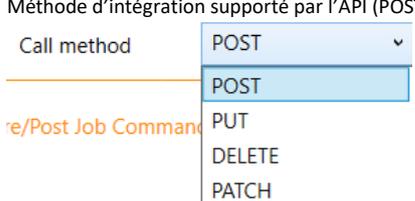
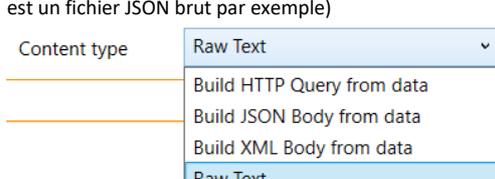
### WebService REST

**Driver Parameters**

**Server Responses**

- Save Server Responses in Source -> Table/File name (As per Source, auto-create) : myWSanswers
- String pattern in API responses that can be interpreted as a success (ie : <result>OK</result>) : <result>OK</result>
- Add some Source column(s) to server responses (ie : myField1;myField2) : myIDcolumn

Call method : POST  
 Source data will be processed as : Build JSON Body from data  
 Format URL with upper chars  Don't send empty values  
 Source Data offset - Process data starting at field : 0

Save HTTP responses in source	Les webservices génèrent usuellement des réponses (XML, JSON) que Fuzible peut récupérer et intégrer dans la connexion qui a servi de Source (si c'est une BDD, dans une table, si c'est une autre connexion, dans des fichiers)
Log Table/File Name	(Optionnel) nom de la table (ou du fichier) qui va recevoir les réponses des requêtes faites à l'API
Track source column(s) in responses	(Optionnel) Si votre requête Source contient 100 lignes, cela signifie qu'il y aura 100 appels au webservice, et il n'est pas facile de se repérer dans la liste des réponses que celui-ci va délivrer. Vous pouvez ici définir une ou plusieurs colonnes issues de la source à recopier dans la table des réponses afin de savoir quelle ligne d'origine a créé quel message.
String that says success in WS Answers	Si vous connaissez le format de réponse d'un webservice, vous pouvez ici définir un « bout de contenu » de ces réponses qui identifie l'envoi comme ayant été un succès. Si ce morceau de chaîne n'est pas dans la réponse, Fuzible affichera une ligne de « WARNING » dans le LOG
Format URL with upper chars	Mode de compatibilité : certains webservices pour lesquels les données sont envoyées sous forme de paramètres HTTP n'acceptent que des URL formatées en majuscule
Don't Send Empty Values	A la construction de la requête HTTP, Fuzible évitera d'intégrer les champs dont la donnée est vide
Call Method	Méthode d'intégration supporté par l'API (POST, PUT, DELETE, PATCH) 
Content Type	Détermine la façon dont le contenu des données source va être envoyé dans le Webservice. Soit sous forme de donnée JSON ou XML dans le body, soit en tant que paramètres http, soit en mode « raw » (quand votre fichier source est un fichier JSON brut par exemple) 
Columns send offset	Une requête vers un webservice HTTP construit une chaîne concaténée des colonnes et de leurs valeurs. Toutefois, si la requête source renvoie par exemple 10 colonnes, on peut décider de n'envoyer que les 8 dernières dans le webservice si on définit un offset de 2

## Mailbox

### Driver Parameters

Assemble queries with same recipient in a single mail

Data Presentation:

Use an HTML template file:  

Keyword Identifier that will be replaced by Query Results (formatted as an HTML table)

?

Assemble queries with same recipient in one mail	Si le Job possède plusieurs requêtes destinées au(x) même(s) adresse(s) mail, on peut décider de grouper tous les résultats dans un seul mail plutôt que d'en envoyer 1 par requête
Data Presentation	<p>Ici, on choisit la façon dont les données Source vont être présentées dans le mail :</p> <ul style="list-style-type: none"> <li>- HTML table in mail body : un tableau HTML dans le corps du mail</li> <li>- Excel file with (or without) an header : une pièce jointe au format Excel</li> <li>- CSV file with (or without) an header : une pièce jointe au format CSV</li> </ul> <p>Note : si le tableau contient trop de lignes (à paramétrer dans la configuration du programme), et qu'on a choisi « HTML table », c'est une pièce jointe CSV qui sera attachée au mail plutôt qu'un tableau dans le corps</p> <p>Note 2 : le nom du tableau en mode « HTML table » correspondra à l'alias de la première table de la requête. Ex : SELECT * FROM MYTABLE -&gt; MyTable sera l'entête du tableau Ex2 : SELECT * FROM MYTABLE as My_Reporting -&gt; My Reporting sera l'entête du tableau (l'underscore est remplacé par un espace)</p>
Use an HTML Template file	<p>S'affiche uniquement si vous avez choisi « HTML table in mail body »</p> <p>Fuzible crée du contenu HTML à partir des données des Requetes Source. Le Template par défaut est très simple, c'est pourquoi vous pouvez en choisir un autre.</p> <p>Dans ce cas, Fuzible doit savoir ou inclure les données de la Requete Source dans ce Template. C'est là que le mot-clé a son utilité :</p> <p>1/ Si aucun mot-clé n'a été saisi, Fuzible va procéder comme ceci : Il va prendre l'alias de la requête (ex : mymail@mail.com:select * from myCustomers as MonAlias) et tâcher de le trouver dans le Template HTML.</p> <ul style="list-style-type: none"> <li>- Si trouvé (dans l'exemple, MonAlias), celui-ci sera remplacé par le code HTML qui a été produit par les résultats de la Requete.</li> <li>- Si non trouvé, les résultats seront simplement concaténés au code HTML du Template.</li> </ul> <p>2/ Si un mot-clé a été spécifié :</p> <ul style="list-style-type: none"> <li>- Fuzible va remplacer ce mot-clé par le code HTML qui a été produit par les résultats de la Requete.</li> <li>- Si le mot-clé n'a pas été trouvé dans le Template, les résultats seront simplement concaténés au code HTML du Template.</li> </ul> <p>3/ Cas spécial en utilisant la fonction "Assembler les requêtes qui ont le même destinataire en un seul mail"</p> <p>Si vous avez plusieurs requêtes qui doivent être assemblées en un seul mail, le Template peut être peuplé intelligemment.</p> <ul style="list-style-type: none"> <li>- Le Template utilise un mot-clé (ex : MYTABLE01) qui doit servir à recevoir les résultats de la première Requete.</li> <li>- Le Template utilise un autre mot-clé (ex : MYTABLE02) qui doit servir à recevoir les résultats de la seconde Requete.</li> </ul> <p>-&gt; Vous pouvez alors définir vos Requetes ainsi :</p> <ul style="list-style-type: none"> <li>- MyMail@mail.com:SELECT * FROM MyCustomers as MYTABLE01 WHERE last_transaction = CURRENT_TIMESTAMP</li> <li>- MyMail@mail.com:SELECT * FROM MySuppliers as MYTABLE02 WHERE last_transaction = CURRENT_TIMESTAMP</li> </ul> <p>Les résultats de la Requete 'MyCustomers' seront injectés dans le Template en lieu et place du mot 'MYTABLE01' Les résultats de la Requete 'MySuppliers' seront injectés dans le Template en lieu et place du mot 'MYTABLE02'</p> <p><b>Note Pratique : si vous avez injecté des paramètres dynamiques reconnus par Fuzible dans votre Template HTML (ex : {?1}), ceux-ci seront remplacés par les valeurs dynamiques du Job !</b></p>
Keyword Identifier	Le mot-clé contenu dans le Template qui détermine l'endroit où le tableau de données sera injecté

Note : L'objet du mail sera la description du Job

## Active Directory

### Driver Parameters

Key Attribute (Primary Key):

Existing Objects:

Activate created Entries

Key Property	Définit la propriété unique qui permet d'identifier un objet dans l'AD (par exemple, « name » est la propriété unique par défaut pour un compte utilisateur)
Existing Objects	<p>Choisit le mode de comportement à adopter quand on tente d'écrire sur un nouvel objet AD :</p> <ul style="list-style-type: none"> <li>- Remove : On le supprime pour le recréer</li> <li>- Ignore : On laisse l'objet tel quel, sans intervenir dessus</li> </ul> <p>Note : En mode « Data Synchronization », cette option est désactivée car on compare les données : les entrées seront donc mises à jour, ajoutées ou supprimées selon le mode de synchronisation choisi dans les paramètres principaux du Job</p>
Activate New Entries	Lorsqu'une nouvelle entrée est créée, elle ne sera pas activée par défaut. On peut forcer son activation en cochant cette case

## Onglet Queries

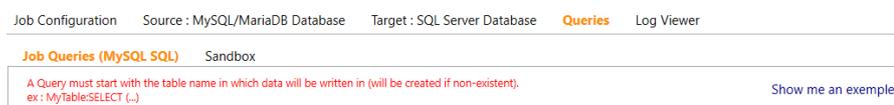
Le coeur de Fuzible se trouve ici. Le logiciel étant un interfaceur, une Source de données n'est ni plus ni moins qu'un ensemble de colonnes dans lesquelles il y a des valeurs.

Le logiciel a pour but de simplifier considérablement le travail fastidieux de mapping et de transformation des données. Il utilise le principe des requêtes SQL pour fonctionner.

Dans le cas où la source est un SGBD, pas de difficulté, c'est le langage SQL du SGBD qui va fonctionner, vous pouvez saisir n'importe quelle requête (simple ou complexe) compatible avec celui-ci pour extraire des données.

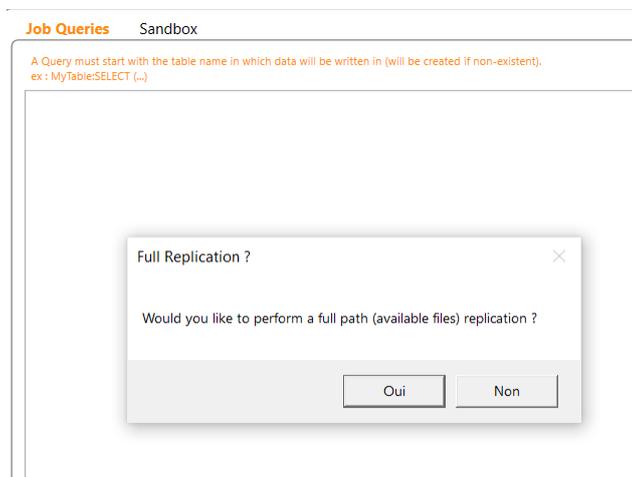
Dans tous les autres cas, Fuzible s'appuie sur le langage SQL standard (compatible SQL92) : Les requêtes saisies sont traduites et appliquées au type de source requêté.

Le sous-menu vous rappelle le langage SQL que vous manipulez lorsque vous constituez vos requêtes.

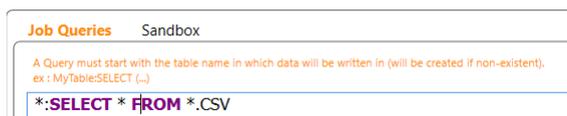


Chaque requête d'un Job s'écrit comme telle : **OUTPUT :SELECT [...]**

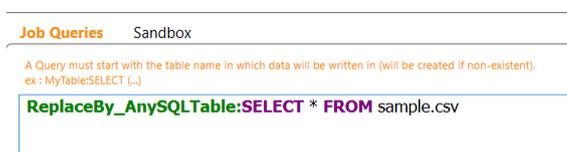
Si vous faites un clic-droit sur la zone des requêtes lorsque celle-ci est vide, une requête de démonstration vous sera proposée si vous n'êtes pas familier avec ce mode de fonctionnement. Si la Source est une base de données ou bien un fichier, on vous proposera de réaliser une réplification de tout ce qui est disponible :



Ce qui produira le résultat suivant (dans l'exemple, ma connexion Source est un répertoire local contenant des fichiers CSV) :



Si vous refusez la réplification complète, une requête de démonstration sera ajoutée :



A présent, voyons comment écrire une requête manuellement, puisque tout l'intérêt est là !

Output	
BDD	<p>Nom de la table de destination</p> <p><small>A Query must start with the table name in which data will be written in (will be created if non-existent). ex : MyTable:SELECT (...)</small></p> <pre>AX_f_fae_avaoires:select * from (SELECT T1.RECID AS id_ecriture, CAST(T1.ACCOUNTINGCURRENCYAMOUNT AS numeric(12,3)) AS nb_montant, CAST(LEFT(CONVERT(varchar, T2.ACCOUNTINGDATE,112),6) AS INTEGER) AS id_anneemois,</pre>
FICHER	<p>Nom du fichier de destination</p> <p><small>A Query must start with the filename in which data will be written in. ex : MyFile.CSV:SELECT (...)</small></p> <pre>CLI_[CLI_code]_{%YYYY%MM%DD%HH%mm%SS}.XML:select * from cli_viewcli_export_comete as clients SIT_[SIT_code]_{%YYYY%MM%DD%HH%mm%SS}.XML:select * from cli_viewsit_export_comete as sites</pre> <p><i>Note : dans l'exemple, le nom du fichier de sortie est dynamique !</i></p>
MAIL	<p>Adresse mail du destinataire. Il est possible d'en définir plusieurs en mettant un « ; » entre chaque !</p> <p><small>A Query must start with the mail address(es) for which you want to send data to (addresses have to be separated by a ';'). ex : mymail@gmail.com,anymail@yahoo.fr:SELECT (...)</small></p> <pre>mymail@hotmail.com:select * from (select snp.id_matricule as Matricule, concat(snp.li_nom, ' ', snp.li_prenom) as Nom_Prenom, ptc.li_typecontrat, pmfc.li_motifcontrat as Motif_Fin_Contrat, pqc.li_qualifcontrat as Qualification_Contrat, pa.li_agence as Agence,</pre>
WEBSERVICE	<p>Nom de l'objet API qui va recevoir les données (basiquement, la queue de l'URL)</p> <p><small>A Query must start with the target Webservice Object (URL queue) in which the source data will be sent in. ex : /ws/mywebservice:SELECT (...)</small></p> <pre>subjects/Account/:select * from Account.csv</pre>
ACTIVE DIRECTORY	<p>Nom de l'objet AD sur lequel on va intervenir. Le nom des champs requêtés de la Source doit correspondre à un champ existant dans l'Active Directory. On peut utiliser un alias sur ces champs s'ils n'ont pas le nom attendu (ex : SELECT myname as name, account as sAMAccountName FROM...)</p> <pre>users:select addata.sAMAccountName, addata.name, addata.description from ad_users.csv as addata</pre>

## Fonctionnement du mapping de champs

Par simplification, une requête SELECT s'écrit ainsi :

```
SELECT * FROM [SOURCE]
```

Ou

```
SELECT champ1, champ2 [...] FROM [SOURCE]
```

Ou

```
SELECT champ1 as dest1, champ2 as dest2 [...] FROM [SOURCE]
```

Ou

```
SELECT CONCAT(champ1, champ2) as dest1, TRIM(champ2) as dest2 [...] FROM [SOURCE]
```

L'alias de champ sert de référence à Fuzible pour construire ou alimenter la connexion Cible. S'il n'est pas présent, le nom de la colonne est utilisé, sinon, c'est l'alias :

Sur un « SELECT \* », les noms de colonne de la cible seront les mêmes que celles de la source

Sur un « SELECT champ1 », le nom de colonne dans la cible sera « champ1 »

Sur un « SELECT champ1 as dest1 », c'est « dest1 » qui sera utilisé comme nom de colonne dans la cible

...et ainsi de suite.

D'où l'utilité de bien préparer sa requête source.

**Note importante :** Si vous souhaitez mettre un alias sur les colonnes requêtées, vous devez impérativement utiliser « AS »

En effet, si un fichier CSV comprend des noms de colonnes avec des espaces, Fuzible sera incapable de séparer l'alias et les champs. L'encadrement n'est pas supporté (guillemets ou crochets autour des champs)

Exemple :

<b>SELECT</b> mon champ, mon deuxième champ <b>FROM</b> monfichier.CSV	OK
<b>SELECT</b> mon champ 1erchamp, mon deuxième champ 2emechamp <b>FROM</b> monfichier.CSV	NON-OK
<b>SELECT</b> mon champ <b>as</b> 1erchamp, mon deuxième champ <b>as</b> 2emechamp <b>FROM</b> monfichier.CSV	OK

Dans le cas où la cible est un SGBD, Fuzible compare les colonnes source et cible, il effectue ses opérations « INSERT, DELETE, UPDATE » en fonction de ce qui est disponible dans la cible uniquement. Si par exemple vous requêtez une source qui possède 50 colonnes et que la cible ne possède que 25 de ces colonnes, ça ne pose aucun problème. L'inverse est aussi exact.

Concernant l'inter-compatibilité (la source serait SQL server, la cible serait MySQL), Fuzible transcode les données à la volée pour les rendre compatibles entre la Source et la Cible, vous n'avez donc pas à vous préoccuper du type de données des champs.

**Notez que Fuzible SQL support l'encadrement des champs lorsque les colonnes/tables utilisent des caractères spéciaux :**

**ex : SELECT "my,weird field/with !strangechars" FROM myFile**

### Cas particulier des requêtes de synchronisation :

La synchronisation travaille en comparant les données source et les données cible. Ce mode de fonctionnement requiert d'exécuter la même requête dans les 2 environnements. Le programme sait transcoder la plupart des requêtes mais il y a certaines limites :

Exemple 1 :

**CIBLE** :SELECT champ1 as dest1, champ2 as dest2 FROM SOURCE WHERE champ1 = 'TEST'

➔ La requête qui sera jouée sur la cible sera :

SELECT dest1, dest2 FROM CIBLE WHERE dest1 = 'TEST'

Vous remarquez que la condition WHERE a été transcodée.

En revanche, prenons un autre cas :

**CIBLE** :SELECT champ1 as dest1, champ2 as dest2 FROM SOURCE WHERE champ3 = 'TEST'

➔ La requête qui sera jouée sur la cible sera :

SELECT dest1, dest2 FROM CIBLE WHERE champ3 = 'TEST'

Le problème, c'est que "champ3" n'existe pas dans la liste des champs requêtés. Fuzible ne peut pas savoir à quoi « champ3 » fait référence dans la table Cible. Si par chance ce champ existe (tables ISO-périmètre), ça ne lui posera pas de problème, mais si ce champ n'existe pas (parce que la requête source est complexe, que les conditions font références à des champs sur des tables en jointure, des conditions parfois très complexes (SELECT imbriqués...)) la requête ne pourra pas aboutir, et la synchronisation se soldera par un échec.

Une astuce SQL pour pallier cette particularité : faire exécuter au SGBD source une requête imbriquée :

Prenons cette requête complexe dans Dynamics AX. Elle contient un certain nombre de transformations et de conditions. En l'encadrant comme une sous-requête, Fuzible ne se préoccupera alors que de la requête principale pour faire sa transcodification.

```

AX_f_écritures:SELECT * FROM
(SELECT T1.RECID AS id_écriture,
T1.ACCOUNTINGCURRENCYAMOUNT AS nb_montant,
T2.ACCOUNTINGDATE AS dt_écriture,
CAST(LEFT(CONVERT(varchar, T2.ACCOUNTINGDATE,112),6) AS INTEGER) AS id_anneemois,
T6.DATAAREA AS id_societe_ax,
MA.MAINACCOUNTID AS id_comptecomptable,
T3.DISPLAYVALUE as li_analytique,
CASE WHEN SUBSTRING(T3.DISPLAYVALUE, 8, 3) = '-' OR CHARINDEX('-', T3.DISPLAYVALUE) = 0 THEN NULL ELSE
SUBSTRING(T3.DISPLAYVALUE, 8 + LEN(MA.MAINACCOUNTID) - 6, 3) END as id_bu_ax,
SUBSTRING(T3.DISPLAYVALUE, 12 + LEN(MA.MAINACCOUNTID) - 6, 3) as id_activite_ax,
SUBSTRING(T3.DISPLAYVALUE, 16 + LEN(MA.MAINACCOUNTID) - 6, 3) as id_agence_ax,
CASE WHEN LEN(LTRIM(SUBSTRING(T3.DISPLAYVALUE, 20 + LEN(MA.MAINACCOUNTID) - 6, 9))) IN (3,5) THEN '' ELSE
REPLACE(SUBSTRING(T3.DISPLAYVALUE, 20 + LEN(MA.MAINACCOUNTID) - 6, 9), '-', '') END as id_chantier_ax,
CASE WHEN LEN(LTRIM(SUBSTRING(T3.DISPLAYVALUE, 20 + LEN(MA.MAINACCOUNTID) - 6, 9))) = 3 THEN SUBSTRING(T3.DISPLAYVALUE, 20 +
LEN(MA.MAINACCOUNTID) - 6, 3) ELSE '' END as id_metier_ax,
CASE WHEN LEN(LTRIM(SUBSTRING(T3.DISPLAYVALUE, 20 + LEN(MA.MAINACCOUNTID) - 6, 9))) = 5 THEN SUBSTRING(T3.DISPLAYVALUE, 20 +
LEN(MA.MAINACCOUNTID) - 6, 5) ELSE SUBSTRING(T3.DISPLAYVALUE, CASE WHEN LEN(LTRIM(SUBSTRING(T3.DISPLAYVALUE, 30 +
LEN(MA.MAINACCOUNTID) - 6, 5))) = 4 THEN 29 + LEN(MA.MAINACCOUNTID) - 6 ELSE 30 + LEN(MA.MAINACCOUNTID) - 6 END, 5) END as
id_destination_ax,
t1.text AS li_écriture,
t2.SUBLEDGERVOUCHER AS li_numero_piece,
ljt.JOURNALNAME as id_code_journal,
T2.JOURNALNUMBER as id_journal,
T1.QUANTITY AS nb_quantite,
T2.CREATEDDATETIME as dt_saisie,
CONVERT(nvarchar(6), T2.CREATEDDATETIME, 112) as id_anneemois_saisie,
T2.CREATEDBY as li_utilisateur
FROM GENERALJOURNALACCOUNTENTRY T1
LEFT JOIN GENERALJOURNALENTY T2 ON (T1.GENERALJOURNALENTY=T2.RECID AND (T1.PARTITION = T2.PARTITION))
LEFT JOIN LEDGERENTRYJOURNAL lej on t2.LedgerEntryJournal=lej.Recid
LEFT JOIN LEDGERJOURNALTABLE ljt on lej.JournalNumber=ljt.JournalNum and ljt.DATAAREAID=T2.SUBLEDGERVOUCHERDATAAREAID
LEFT JOIN DIMENSIONATTRIBUTEVALUECOMBINATION T3 ON (T1.LEDGERDIMENSION=T3.RECID AND (T1.PARTITION = T3.PARTITION))
LEFT JOIN LEDGER T4 ON (T2.LEDGER=T4.RECID AND (T2.PARTITION = T4.PARTITION))
LEFT JOIN FISCALCALENDARPERIOD T5 ON (T2.FISCALCALENDARPERIOD=T5.RECID AND (T2.PARTITION = T5.PARTITION))
LEFT JOIN DIRPARTYTABLE T6 ON (((T6.PARTITION=T1.PARTITION) AND (T6.PARTITION=T1.PARTITION)) AND (T6.PARTITION=T1.PARTITION))
AND (T4.PRIMARYFORLEGALENTY=T6.RECID AND (T4.PARTITION = T6.PARTITION))) AND (T6.INSTANCERELATIONTYPE IN (41) ))
Left join MAINACCOUNT MA on MA.RECID = T3.MAINACCOUNT
LEFT JOIN DIMENSIONHIERARCHY H ON T3.ACCOUNTSTRUCTURE = H.RECID AND H.PARTITION=T3.PARTITION
WHERE 1 = (CASE WHEN t2.SUBLEDGERVOUCHER LIKE 'CLOTURE%' AND MONTH(T2.ACCOUNTINGDATE) = 12 AND MA.MAINACCOUNTID <
600000 THEN 0 ELSE 1 END)
AND 1 = (CASE WHEN (T6.DATAAREA IN ('AIR', 'MPY') AND T2.ACCOUNTINGDATE >= '01/01/2019') THEN 0 ELSE 1 END)
) as REQ
WHERE REQ.id_anneemois_saisie >= 202001

```

Sera transcodée pour la cible en tant que : SELECT \* FROM **AX\_f\_écritures** WHERE id\_anneemois\_saisie >= 202001

Voyons à présent ce qu'il est possible de faire avec une requête SQL :

## SELECT – A partir d’une base de données, vers une base de données

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

[Quick Help](#)

```
AX_dim_agences:select distinct value as id_agence_ax, name as li_agence from DimAttributeOMCostCenter
AX_dim_societes:select ID as id_societe_ax, NAME as li_societe from BICOMPANYVIEW
AX_dim_activites:select value as id_activite_ax, name as li_activite from DimAttributeOMBusinessUnit
AX_dim_metiers:select value as id_metier_ax, name as li_metier from DimAttributeOMDepartment
AX_dim_plancomptable:SELECT DISTINCT(A.[MAINACCOUNTID]) as id_comptecomptable,
LEFT(A.[MAINACCOUNTID], 1) as id_racine_comptecomptable,
A.[NAME] as li_comptecomptable,
A.[TYPE] as id_type,
CASE A.[TYPE] WHEN 0 THEN 'Bilan' WHEN 3 THEN 'Résultat' ELSE 'Inconnu' END as li_type
FROM [MAINACCOUNT] A
INNER JOIN (SELECT MIN(RECID) AS RECID, MAINACCOUNTID FROM MAINACCOUNT GROUP BY MAINACCOUNTID) B ON A.R
ORDER BY A.[MAINACCOUNTID]
```

N’importe quelle requête de type « SELECT » et compatible avec le SGBD source. Celle-ci peut être simple ou complexe.

## SELECT – A partir d’une base de données, vers un fichier

A Query must start with the filename in which data will be written in.  
ex : MyFile.CSV:SELECT (...)

[Show me an exemple](#)

```
MyTutorial.csv:SELECT user as userColumn, connstring_id as idColumn, connstring_name as nameColumn
from user_connstrings WHERE user LIKE 'GUIZ'
MyTutorial2.csv:SELECT COUNT(*) as nbConn
from user_parameters as up
inner join user_connstrings uc on up.user = uc.user
order by up.user ASC
```

N’importe quelle requête de type « SELECT » et compatible avec le SGBD source. Celle-ci peut être simple ou complexe.

## SELECT – A partir d’une base de données, vers une adresse mail

A Query must start with the mail address(es) for which you want to send data to (addresses have to be separated by a ',').  
ex : mymail@gmail.com,anymail@yahoo.fr:SELECT (...)

[Quick Help](#)

```
{?1}:select * from v_adm_metriques as Métriques_Principales;
{?1}:SELECT nspname || '.' || relname AS "Nom_table",pg_size_pretty(pg_relation_size(Top_10_des_tables_les_plus_lourdes.oid))
WHERE nspname NOT IN ('pg_catalog', 'information_schema') and relkind = 'r'
ORDER BY pg_relation_size(Top_10_des_tables_les_plus_lourdes.oid) DESC LIMIT 10
{?1}:SELECT Top_10_des_tables_avec_le_plus_de_lignes.relname as Nom_table, reltuples as Nombre_de_lignes
FROM pg_class as Top_10_des_tables_avec_le_plus_de_lignes
JOIN pg_stat_user_tables AS tabstat ON Top_10_des_tables_avec_le_plus_de_lignes.relname = tabstat.relname
ORDER BY reltuples DESC LIMIT 10
```

N’importe quelle requête de type « SELECT » et compatible avec le SGBD source. Celle-ci peut être simple ou complexe.

**Note :** Constatez que la requête source est variabilisée, dans cet exemple, on met en « OUTPUT » une adresse mail dynamique, définie dans l’onglet « Job Configuration »

En dehors d’une requête sur un SGBD, c’est le moteur SQL de Fuzible qui prend le relais :

## SELECT - A partir d'un fichier

A Query must start with the filename in which data will be written in.  
ex : MyFile.CSV:SELECT (...)

```
SAMPLE_OUTPUT_LEFTJOIN.XLSX:SELECT fileA.id_sample AS IdInitial, fileB.id_sample_join AS IdJoin,
fileB.li_sample_join as LiJoin, ISNULL(fileB.li_sample_join, 'Nothing') as IsNullField
FROM SAMPLE.CSV AS fileA
LEFT JOIN SAMPLE_JOIN.CSV AS fileB ON fileA.id_sample = fileB.id_sample_join
SAMPLE_OUTPUT_INNERJOIN.XLSX:SELECT fileA.id_sample AS IdInitial, fileB.id_sample_join AS IdJoin,
fileB.li_sample_join as LiJoin FROM SAMPLE.CSV AS fileA
INNER JOIN SAMPLE_JOIN.CSV AS fileB ON fileA.id_sample = fileB.id_sample_join
SAMPLE_INNERJOIN_SUBQUERY.XLSX:SELECT fileA.id_sample AS IdInitial, fileB.id_sample_join AS IdJoin,
fileB.li_sample_join as LiJoin FROM SAMPLE.CSV AS fileA
INNER JOIN (SELECT * FROM SAMPLE_JOIN.CSV) AS fileB ON fileA.id_sample = fileB.id_sample_join
```

Voyez ici un exemple un peu complexe de requêtes réalisées sur plusieurs fichiers, avec des jointures, et même une sous-requête. La syntaxe SQL est rigoureusement la même que celle d'un SGBD traditionnel.

## SELECT - A partir d'un webservice en mode Fuzible SQL (A)

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

[Quick Help](#)

```
gpi_tickets:select
Entity.id as id_entite,
Entity.name as li_entite,
Ticket.id as id_ticket,
Ticket.name as li_ticket,
Ticket.date as dt_ticket,
Ticket.closedate as dt_cloture,
Ticket.urgency as nb_urgency,
Ticket.impact as nb_impact,
Ticket.priority as nb_priority,
Ticket.status as id_status,
Cat.id as id_category,
Cat.completename as li_category,
CASE Ticket.status WHEN 1 THEN 'Nouveau' WHEN 3 THEN 'Planifié' WHEN 4 THEN 'En Attente' WHEN 6 THEN 'Cloturé' WHEN
Ticket.users_id_recipient as id_user,
User.name as li_user,
ISNULL(Item.id, 1) as id_item,
Item.itemtype as li_itemtype,
CASE Item.itemtype WHEN 'Software' THEN Soft.id ELSE " END as id_software,
CASE Item.itemtype WHEN 'Software' THEN Soft.name ELSE " END as li_software,
ISNULL(Grp.id, 0) as id_groupe,
Grp.name as li_groupe,
usrTicket.users_id as id_user_attribue,
UserAffecte.name as li_user_attribue,
Req.id as id_requesttype,
Req.name as li_requesttype,
CASE Ticket.type WHEN 1 THEN 'Incident' ELSE 'Demande' END as li_type
FROM Ticket/?order=desc&range={?1}&is_deleted=0 as Ticket
INNER JOIN RequestType/ AS Req ON Req.id = Ticket.requesttypes_id
INNER JOIN Entity/ AS Entity ON Entity.id = Ticket.entities_id
LEFT JOIN (SELECT tickets_id, MAX(groups_id) as groups_id FROM group_Ticket?order=desc&range=0-15000 GROUP BY ticket
LEFT JOIN (SELECT MAX(users_id) as users_id, tickets_id from Ticket_User/?range=0-20000&order=desc where type = 2 group
LEFT JOIN User/?range=0-5000 as UserAffecte ON usrTicket.users_id = UserAffecte.id
LEFT JOIN Group/ AS Grp ON Grp.id = GrpTicket.groups_id
LEFT JOIN User/?range=0-5000 as User ON User.id = Ticket.users_id_recipient
LEFT JOIN ITILCategory/ as Cat ON Cat.id = Ticket.itilcategories_id
LEFT JOIN Item_Ticket/?order=desc&range=0-20000 as Item ON Item.tickets_id = Ticket.id AND Item.itemtype <> 'PlugIn'
LEFT JOIN Software/?order=desc&range=0-20000 as Soft ON Soft.id = Item.items_id
```

L'exemple présenté est particulièrement pointu et permet de montrer à quel point on peut réaliser une requête telle qu'on la réaliserait dans un SGBD. En revanche, contrairement à un SGBD, l'entête n'est pas connu par avance, il est conseillé de faire quelques « SELECT \* » pour interroger les webservices avant de se lancer dans des jointures et autres transformations.

L'onglet « Sandbox » permet de faire des tests.

### Exemple du webservice du logiciel GLPI :

Requête simple :

```
SELECT * FROM Computer/?range=0-5000
```

→ Ramène la liste de tous les ordinateurs (5000 entrées)

Je peux bien entendu filter :

```
SELECT * FROM Computer/?range=0-5000 where states_id = 1
```

→ Je sais que le champ "states\_id" existe dans ce que me ramène le webservice donc je peux filtrer sur ce champ

De la même manière, si je connais la liste des champs, je peux requêter ainsi :

```
SELECT serial as Serial, name as Machine, users_id as Glpi FROM Computer/?range=0-5000 WHERE states_id=1
```

## SELECT – A partir d'un webservice en mode Fuzible SQL (B)

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

```
sample3:SELECT TABLE 1 id_sample,  
substring(li_random_string, 1, 5) as li_substr  
FROM /ws/public/sample_data?order=desc[{"id_ssgroup":"001"}]
```

Ce webservice renvoie par exemple 2 tables de données. Je peux décider de récupérer les 2 (donc créer 2 tables dans la cible), ou bien ne récupérer qu'une seule des 2

La fonction "TABLE x" détermine ici la table de données sur laquelle la requête s'applique. L'autre table sera retournée comme un simple « SELECT \* », le nom des champs dans la cible sera le même que celui de la source car il n'est pas possible d'intervenir sur plusieurs tables du résultat d'origine.

Il est possible de ne récupérer qu'une seule des 4 tables grâce à la syntaxe "ONLY":

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

```
sample3:SELECT TABLE 1 ONLY id_sample,  
substring(li_random_string, 1, 5) as li_substr  
FROM /ws/public/sample_data?order=desc[{"id_ssgroup":"001"}]
```

Concernant l'API requêtée, on peut intégrer du contenu "BODY" dans la requête, ici, la documentation du webservice (API) précise la manière de filtrer l'un des champs en utilisant un corps au format JSON

→ On utilise les crochets pour intégrer n'importe quel contenu body (au format XML, JSON ou Form-data)

Contrairement à un SGBD, l'entête n'est pas connu par avance, il est conseillé de faire quelques « SELECT \* » pour interroger le service mail avant de se lancer dans des jointures et autres transformations.

L'onglet « Sandbox » permet de faire des tests.

## SELECT – A partir de l'API Salesforce en mode SoQL

Lorsque vous avez paramétré la connexion Source avec un template qui possède un langage API particulier (graphql, NxQL, SoQL), vous pouvez utiliser ce langage pour requêter l'API plutôt que d'utiliser Fuzible SQL, ce qui vous facilitera nettement le travail de transcodage.

Par exemple, dans l'exemple suivant, on requête le CRM Salesforce en utilisant SoQL :

Job Queries (SOQL) Sandbox

A Query must start with the filename in which data will be written in.  
ex : MyFile.CSV:SELECT (...)

```
salesforce.csv:select name, BillingCity from Account WHERE isDeleted = false
```

Ainsi que la même requête, en mode Fuzible SQL, ressemble à ça, ce qui est nettement moins pratique :

Job Queries (Fuzible SQL) Sandbox

A Query must start with the filename in which data will be written in.  
ex : MyFile.CSV:SELECT (...)

```
salesforce.csv:select * from /query/?q=SELECT+name,BillingCity+from+Account+WHERE+isDeleted+=+false
```

## SELECT - A partir d'une boîte mail

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

```
myOutputTable:select TOP 3 DATE, FROM, SUBJECT, UID from mymail@gmail.com
myOutputTable2:select TOP 10 DATE as myDate, FROM as From, SUBJECT as mySubject, UID as IdMail
from myothermail@gmail.com[myPassword]
WHERE DATE >= '01/01/2020'
```

### Exemple d'une adresse GMAIL :

```
SELECT * FROM mymail@gmail.com[mypassword]
  → Va ramener la liste des mails de l'adresse « mymail@gmail.com » en utilisant le mot de passe saisi entre crochets
SELECT * FROM mymail@gmail.com
  → Va ramener la liste des mails de l'adresse « mymail@gmail.com » en utilisant le mot de passe saisi dans la chaîne de connexion
```

Je peux bien entendu filtrer et nommer les colonnes si je les connais (rappel : la SANDBOX est faite pour ça) :  
SELECT SUBJECT, SENDER, TEXTBODY, TO FROM mymail@myProvider.com WHERE DATE >= '01/01/2019'

## SELECT - A partir de l'Active Directory

A Query must start with the filename in which data will be written in.  
ex : MyFile.CSV:SELECT (...)

Quick Help

```
test_ad_grp.csv:select * from groups
```

Comme pour les webservices et les services mail, on ne connaît pas forcément le nom des colonnes que qu'un objet Active Directory renvoie. Il est conseillé, pour tester de faire un simple SELECT \* FROM USERS  
Le « FROM » fait référence à l'objet AD qu'on interroge (USERS ou GROUPS)  
L'exemple présenté ramène la liste des groupes AD qui existent.

## ASSISTANT DE SAISIE

A la manière des outils SQL, le programme propose des facilités de saisie de requête à travers un menu contextuel qui s'affiche en fonction des mots saisis :

Dans l'exemple suivant, je requête un SGBD et viens de saisir « FROM » : le menu me propose alors une liste des tables disponibles. Un simple « TAB » me permet d'accéder à ce menu flottant, les flèches permettent de choisir une table, tandis que « ENTER » permet d'insérer l'élément choisi dans la requête

Job Queries Sandbox

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

Show me an exemple

```
myTable:select * from |
user_constrings
user_constrings_params
job_queries
sqlite_sequence
app_stacklaunch
app_planifmodel
service_parameters
client_jobs
app_log_ent
app_log_lig
app_synchro_records
shs_replicator_synchro_records
user_parameters
sneakpeak
MyTutorial
sample_table_3
sample
sample_table_1
sample_table_2
sample_table_4
sample_table_5
prefix_sample_table_3
prefix_sample
prefix_sample_table_1
prefix_sample_table_2
```

Right-click on a query for advanced operations or press 'F5' to show Data

Quick Help

● Synchro : Bypass query filters in Target

Dans l'exemple suivant, j'ai déjà une table et je complète mon « SELECT » : la liste des champs est présentée, ainsi que les fonctions SQL classiques qui peuvent être utilisées.



The screenshot shows the 'Job Queries Sandbox' interface. At the top, it says 'Job Queries Sandbox' and 'Show me an exemple'. Below that, a note states: 'A Query must start with the table name in which data will be written in (will be created if non-existent). ex: MyTableSELECT (...)'. The main area contains a text input with the query 'myTable:select st1, | from sample\_table\_1 as st1'. A dropdown menu is open, listing the following fields: 'id\_sample', 'li\_sample', 'dt\_random\_date', 'nb\_random\_number', 'li\_random\_string' (which is highlighted), 'id\_group', and 'id\_ssgroup'. At the bottom, there are instructions: 'Right-click on a query for advanced operations or press 'F5' to show Data' and 'Quick Help'. A small icon and text 'Synchro : Bypass query filters in Target' are also visible.

L'assistant de saisie fonctionne pour tous les types de sources, il est beaucoup plus avancé pour les requêtes sur fichiers et SGBD car pour les autres cas, il n'existe pas de méthode pour, par exemple, exposer la liste des objets d'un webservice et encore moins la liste des champs disponibles. Dans ce cas, il sert simplement d'assistant à la création d'une fonction SQL (un CONCAT, un ISNULL...)

## SELECT AVANCE

Les exemples précédents sont relativement simples, ils montrent comment on peut basiquement interroger une source sous forme de « SELECT », quand celle-ci n'est pas une base de données.

Le langage SQL de Fuzible peut comprendre les syntaxes les plus complexes, et la saisie d'une requête SQL dans l'onglet est augmentée d'un assistant de saisie qui permet, à la volée, de voir et utiliser toutes les options disponibles pour requêter.

Si par exemple je tape « SELECT C », un petit menu contextuel va me proposer plusieurs choses :

- CONCAT
- CASE
- COALESCE
- CHARINDEX
- COUNT
- ...

Je peux alors utiliser l'un de ces éléments pour manipuler les données source. Exemples :

- SELECT CONCAT(champ1, champ2) as dest1 [...]
- SELECT CASE champ1 when 'OUI' then 1 WHEN 'NON' then 2 ELSE 0 END as dest2 [...]
- SELECT COUNT(champ1) as dest1, champ2 FROM MONFICHIER GROUP BY champ2

Pour mesurer toute l'étendue des possibilités offertes par le moteur SQL, vous disposez d'un Job "Sample" par défaut qui vous présente de nombreuses requêtes utilisant toutes ces fonctions avancées. Vous pouvez vous en inspirer.

**Il y a également une fonction exclusive à Fuzible SQL qui simule une anonymisation des données :**

- SELECT ANONYMIZE(myField) FROM myFile  
Mélange les données de "myField" entre elles
- SELECT ANONYMIZE(myField, "RANDOM") FROM myFile  
Rend aléatoire chaque valeur du champ "myField"
- SELECT ANONYMIZE(myField, "myRandomizationFile.csv") FROM myFile  
Charge le fichier spécifié et l'utilise comme liste aléatoire pour remplacer les valeurs de "myField"

## Fuzible SQL : Limitations techniques :

- 1) Les SELECT imbriqués ne fonctionnent que sur les « tables » et les conditions « where » (voir exemple « webservices A »), on ne peut pas faire un sous-select pour un champ :

### VALIDE :

A Query must start with the filename in which data will be written in.  
ex : MyFile.CSV:SELECT (...)

Quick Help

```
test:select *  
from monfichier as a  
inner join (select MAX(id) as id, monclient as id_client from unfichier where typeclient = 'local') as b ON a.id = b.id
```

### VALIDE :

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

Quick Help

```
test:select *  
from (select * from monfichier) as a
```

### INVALIDE :

A Query must start with the filename in which data will be written in.  
ex : MyFile.CSV:SELECT (...)

Quick Help

```
test:select a.*,  
(select MAX(id) as id, monclient as id_client from unfichier where typeclient = 'local') as b  
from monfichier as a
```

- 2) Vous ne pouvez pas imbriquer les sous-select. La requête n'aboutira pas.  
ex : `SELECT * FROM (SELECT * FROM (SELECT * FROM monfichier) as data) as data`

## Note additionnelle :

Certaines sources peuvent être susceptibles de renvoyer plusieurs tables de données (webservices, fichiers JSON ou XML).

Par défaut, le logiciel va créer autant de cibles que de tables restituées. Pour au temps, toutes les opérations de transformation SQL associées à votre requête (fonctions SQL, Group By, Where, Order By...) vont être appliquées à la première table uniquement.

Pour une raison ou pour une autre, si vous souhaitez que votre requête s'applique à une autre table, il est possible de l'indiquer en procédant ainsi :

OUTPUT :SELECT **TABLE x** champ1, ...

...en indiquant, grâce à "TABLE x", le numéro de table à laquelle s'applique la requête.

Si « x » est invalide (exemple : je mets TABLE 10 alors que la source ne renvoie qu'une table, toutes les opérations associées à la requête seront annulées.

*Bien sûr, vous ne pouvez pas deviner à l'avance ce que retourne la source de données interrogée. L'onglet « sandbox » est là pour ça. Faites-y un simple « SELECT \* FROM masource » et un **F5** pour connaître l'ensemble des résultats retournés par la requête. Si le résultat retourne plusieurs tables, vous le verrez, et pourrez alors consulter le contenu de chacune d'entre elles.*

Vous pouvez aussi indiquer « ONLY » si vous ne voulez récupérer que la table demandée (sinon, les autres seront également traitées).

Exemple : `SELECT TABLE x ONLY champ 1,...`

Regardez le job d'exemple pour comprendre ce qu'il est possible de faire et surtout, expérimentez en utilisant la SANDBOX. Le programme est suffisamment verbeux pour vous permettre de déboguer pas à pas une requête qui serait mal construite.

## SELECT Multi-tables, multi-fichiers

### BDD

Vous pouvez requêter plusieurs tables d'un coup, si par exemple vous voulez récupérer toutes les tables qui contiennent le mot « param », vous pouvez saisir la requête ainsi :

```
OUTPUT : SELECT * FROM %param%
```

Fuzible détecte l'usage du « % » et va rechercher toutes les tables qui correspondent à ce pattern. Il transforme alors une seule requête en plusieurs. Il va sans dire qu'à moins que le nom des champs de ces tables soient tous les mêmes, le « SELECT \* » est recommandé...

Il y a 3 scénarios pour gérer la destination :

- Si vous mettez « \* » (ex : \*:select \* from %param%), le « \* » sera remplacé par le nom de la table.
- Si vous mettez un nom forcé (ex : param:select \* from %param%), les données iront toutes dans le même OUTPUT, ici : « param ». Toutes les données Source de toutes les requêtes correspondant au filtre seront fusionnées.
- Si vous mettez « import\_\* » (ex : param\_\* :select \* from %param%), le « \* » sera remplacé par le nom de la table, mais la table finale aura un nom qui commencera par « import\_ »

### FICHIERS

De la même manière qu'avec une BDD, vous pouvez requêter plusieurs fichiers d'un coup, par exemple, si vous voulez récupérer tous les fichiers qui contiennent le mot « param », vous pouvez saisir la requête ainsi (à la manière d'un « DIR » dans l'invite de commande MSDOS)

```
OUTPUT :SELECT * from *.CSV
```

Fuzible détecte l'usage du « \* » et va récupérer tous les fichiers CSV du répertoire de connexion (chaîne de connexion).

On peut étendre l'usage du « \* » ainsi :

```
OUTPUT :SELECT * from FI*test*.*
```

- ➔ Cela signifie que les fichiers commençant par "FI" puis ayant autre chose ensuite, puis "test", puis autre chose, vont être récupérés.

Comme pour les bases de données, vous pouvez nommer les champs et travailler la requête de manière avancée (avec des fonctions de transformations...etc...) mais en ce cas, il faut que les fichiers soient de même structure et, s'ils ont un entête, que ce soit le même.

La gestion de la destination fonctionne de la même manière qu'expliqué précédemment.

## Requêtes Multi-Cible

Par défaut, un job alimente une seule cible.

On peut toutefois faire sauter cette condition pour en alimenter 2 en parallèle (dans le cas où par exemple on souhaite alimenter 2 BDD à l'identique, en simultané)

Prenons un exemple simple : construisez une requête de Job, puis faites un clic-droit dessus. Un menu contextuel apparaît et vous propose plusieurs options dont « Create Dual Target »

The screenshot shows the 'Job Queries' interface in 'Sandbox' mode. A SQL query is displayed in a text area:

```
mysynchrotable:select tb1.id_sample as idSample, tb1.li_sample as liSample,
tb1.dt_random_date as RandomDate, tb2.SecondColumn as AnotherColumn
from sample1 as tb1
inner join sample2 as tb2 ON tb1.id_sample = tb2.FirstColumn
where 100 > tb1.id_sample
and tb1.id_sample > 10
and (tb1.id_sample > 11 and tb1.id_sample < 99)
and tb1.li_sample not like 'weirdstring'
order by tb1.id_sample desc
```

A context menu is open over the query, titled 'Postgres Raspberry -> mysynchrotable'. The menu items are:

- Query Analyzer
  - > Source Infos
  - > Target Infos
  - > Query Details
- Synchro Query
  - > Transcoded for Target
  - > Validity check for Synchro Query
- Execute Query
  - > Load Source Data (F5)
  - > Run this individual query
- Scripting
  - > Get full header from query and copy/paste it
  - > Dynamic Parameters
  - > Basic Query Builder
- Advanced Query Scripting
  - > Add Cross-Connections Join
  - > Create Dual Target

Une invite va alors vous proposer de choisir vos 2 cibles, et l'output associé. Validez.

The 'Multi-Target' dialog box is shown. It contains the following information:

- Technical Restrictions:
  - You can't add more than 2 targets / query
- First connection in Query: [12] -> POSTGRE : Postgres Rasp (selected), mytargettable
- Second connection in Query: [11] -> MYSQL : MySQL Raspber (selected), myothertargettable
- Accept button

La requête sera donc augmentée d'un petit script :

The screenshot shows the 'Job Queries' interface with the updated SQL query:

```
[12]mytargettable[11]myothertargettable:select tb1.id_sample as idSample, tb1.li_sample as liSample,
tb1.dt_random_date as RandomDate, tb2.SecondColumn as AnotherColumn
from sample1 as tb1
inner join sample2 as tb2 ON tb1.id_sample = tb2.FirstColumn
where 100 > tb1.id_sample
and tb1.id_sample > 10
and (tb1.id_sample > 11 and tb1.id_sample < 99)
and tb1.li_sample not like 'weirdstring'
order by tb1.id_sample desc
```

A context menu is open over the query, titled 'Postgres Raspberry [A] + MySQL Raspberry [B] -> mytargettable'. The menu items are:

- Query Analyzer
  - > Source Infos
  - > Target Infos
  - > Target B Infos
  - > Query Details

Entre crochets, l'identifiant de la connexion, suivi du nom de la table cible. Un nouveau clic-droit pour consulter le menu contextuel, auquel un élément « Target B Infos » a été ajouté, et qui permet d'obtenir des informations à propos de la Cible choisie.

Mais on peut aussi imaginer d'alimenter une table de BDD en parallèle d'un FICHER. Tout est possible. On peut même variabiliser les connexions Cible en utilisant les paramètres dynamiques :

La requête :

```
Job Queries Sandbox
A Query must start with the table name in which data will be written in (will be created if non-existent).
ex : MyTableSELECT (...)
{?1}mysynchrotable{?2}mysynchrofile.csv:select tb1.id_sample as idSample, tb1.li_sample as liSample,
tb1.dt_random_date as RandomDate, tb2.SecondColumn as AnotherColumn
from sample1 as tb1
inner join sample2 as tb2 ON tb1.id_sample = tb2.FirstColumn
where 100 > tb1.id_sample
and tb1.id_sample > 10
and (tb1.id_sample > 11 and tb1.id_sample < 99)
and tb1.li_sample not like 'weirdstring'
order by tb1.id_sample desc
```

Les paramètres dynamiques :

Main Parameters

Job Type: Data Synchronization | Allow Delete, Update, Insert |  Historize UPDATED and DELETED rows

Will compare source and target data, and update target according to what's in the source

Dynamic Parameters: [?1][?2] ?

You can write any text or any available command. Each parameter must be separated by a semicolon (;)  
You can use them anywhere (queries, text fields, connections) by referencing them like this: (?1), (?2)...

[View Job With Replaced Values](#)

Cela présente un intérêt lorsque vous souhaitez basculer des données d'un environnement à un autre (pré-production, production, environnement de développement...) à la volée.

Par ailleurs, on peut choisir d'alimenter les 2 Cibles en parallèle ou bien l'une après l'autre, tout dépend des performances de la machine qui héberge Fuzible (voir Configuration Générale / SHS Analyzer)

### Limitation technique :

Le multi-Cible est limité à 2 cibles.

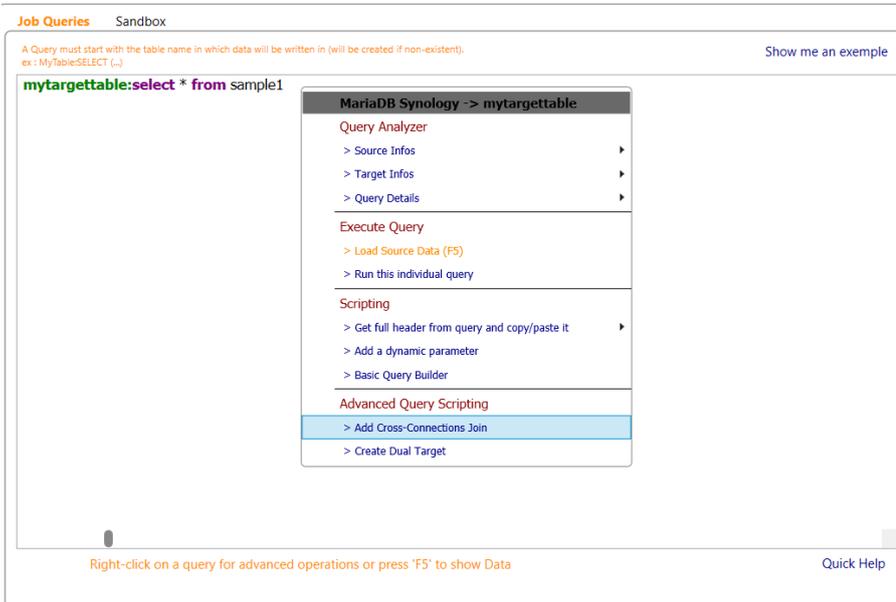
## Requêtes Croisées

Cette fonctionnalité est l'une des plus puissantes de Fuzible. Elle permet de joindre entre elles différentes sources de données : cela signifie que vous pourrez croiser à la volée des données issues d'une base de données avec un fichier, et compléter le tout avec des données issues d'un webservice. Tout est possible et se conçoit de manière assez simple.

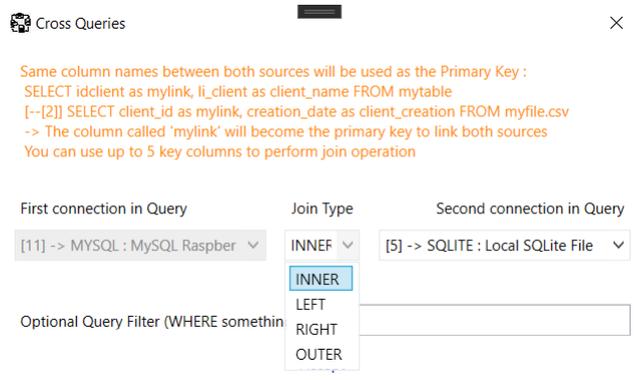
Pour que ce concept soit plus clair, imaginez qu'à la place de faire des jointures entre plusieurs tables SQL pour compléter un jeu de données, vous faites des jointures entre plusieurs connexions différentes.

Prenons l'exemple suivant. Ma connexion principale est une BDD MySQL.

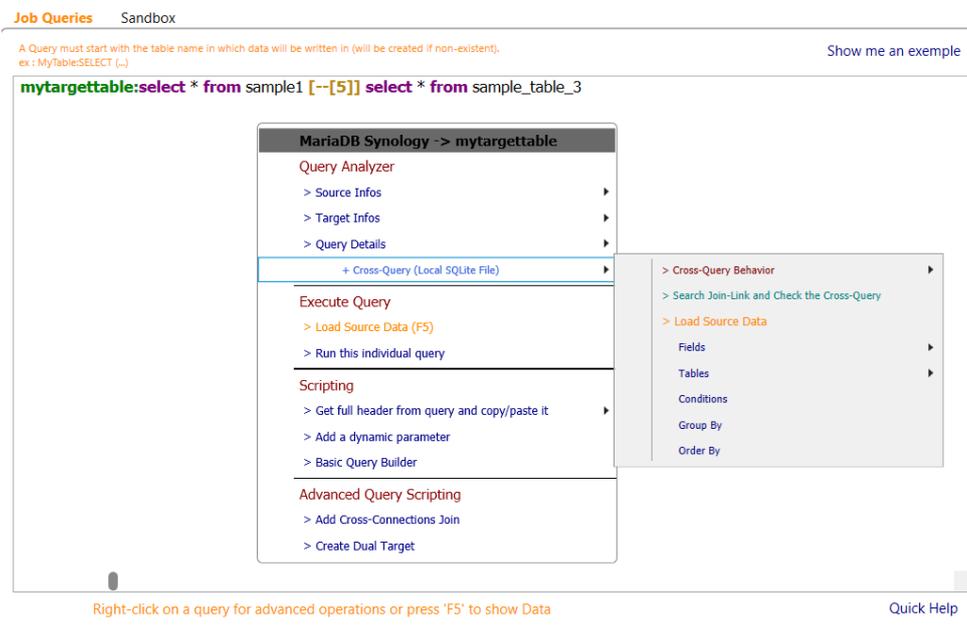
Construisez une requête simple, puis faites un clic-droit dessus. Le menu contextuel apparaît et vous propose plusieurs options dont « Add Cross-Connections Join ».



Une invite vous propose alors de choisir une nouvelle source de données, et le type de jointure qui va être réalisé entre les 2 sources. Je passe sur la fonction « Optional Query Filter » très avancée, qui permet de filtrer les résultats.



La requête va se voir augmentée d'une zone de script [--[5]] Vous pouvez alors à la suite réaliser une requête. Celle-ci sera alors associée à la deuxième connexion. Un clic-droit et vous constaterez que le menu contient un nouveau sous-menu associé à cette requête jointe.



Dans la zone de script, le type de jointure s'exprime ainsi :

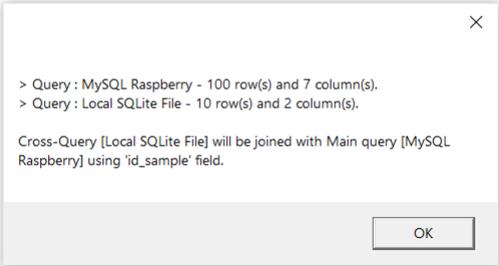
--	INNER JOIN
>-	RIGHT JOIN
<-	LEFT JOIN
<>	OUTER JOIN

La connexion est indiquée entre crochets.

Fuzible n'a qu'une seule manière de définir la façon de joindre les 2 sources entre elles : il utilise les champs au nom identique pour faire sa jointure. Dans l'exemple précédent, un « SELECT \* » est réalisé sur les 2 sources, mais je sais que le champ « id\_sample » est présent sur les 2 sources, donc Fuzible fera la jointure en utilisant ce champ.

En revanche, si les noms de colonne sont différents d'une connexion à l'autre, il faudra nommer spécifiquement les champs. Par exemple :

```
mytargettable:select * from sample1  
[--[5]] select FirstColumn as id_sample, SecondColumn from sample_table_2
```



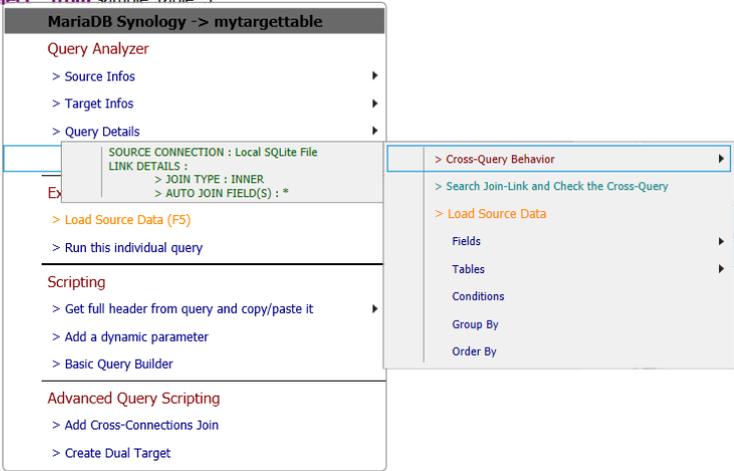
Grâce à l'alias, je force l'établissement du lien entre les 2 sources avec « id\_sample ».

### Comportement d'une requête jointe

Le menu contextuel permet de vous faciliter la compréhension du comportement adopté par Fuzible pour réaliser la requête jointe. Dans l'exemple ci-dessous, SELECT \* étant utilisé dans les 2 connexions, le programme ne pourra déterminer le lien qu'au moment de l'exécution des requêtes : Il vous faut être certain que le champ « id\_sample » est présent dans les 2 jeux de données.

Attention : Tout autre champ ayant le même nom serait également considéré comme partie intégrante de la clé !

```
mytargettable:select * from sample1 [--[5]] select * from sample_table_2
```



Une autre manière consiste à faire un SELECT qui spécifie le nom des champs. Le menu contextuel vous précise alors le ou les champ(s) de jointure associé(s). Inutile d'attendre l'exécution pour connaître le comportement qui sera opéré.

```
mytargettable:select id_sample, id_group from sample1  
[--[5]] select id_sample, li_substr from sample_table_3
```

MariaDB Synology -> mytargettable

- Query Analyzer
  - > Source Infos
  - > Target Infos
  - > Query Details
- + Cross-Query (Local SQLite File)
  - > Cross-Query Behavior
  - > Search Join-Link and Check the Cross-Query
  - > Load Source Data
    - Fields
    - Tables
    - Conditions
    - Group By
    - Order By
- Execute Query
  - > Load Source Data (F5)
  - > Run this individual query
- Scripting
  - > Get full header from query and copy/paste it
  - > Add a dynamic parameter
  - > Basic Query Builder
- Advanced Query Scripting
  - > Add Cross-Connections Join
  - > Create Dual Target

SOURCE CONNECTION : Local SQLite File  
LINK DETAILS :  
> JOIN TYPE : INNER  
> AUTO JOIN FIELD(S) : id\_sample

Enfin, la troisième manière consiste à modifier le script de requête jointe : on peut définir manuellement le ou les champs de jointure en les séparant par une virgule. Le menu contextuel précise que le type de jointure est désormais manuel.

```
mytargettable:select * from sample1  
[-id_sample-[5]] select * from sample_table_3
```

MariaDB Synology -> mytargettable

- Query Analyzer
  - > Source Infos
  - > Target Infos
  - > Query Details
- + Cross-Query (Local SQLite File)
  - > Cross-Query Behavior
  - > Search Join-Link and Check the Cross-Query
  - > Load Source Data
    - Fields
    - Tables
    - Conditions
    - Group By
    - Order By
- Execute Query
  - > Load Source Data (F5)
  - > Run this individual query
- Scripting
  - > Get full header from query and copy/paste it
  - > Add a dynamic parameter
  - > Basic Query Builder
- Advanced Query Scripting
  - > Add Cross-Connections Join
  - > Create Dual Target

SOURCE CONNECTION : Local SQLite File  
LINK DETAILS :  
> JOIN TYPE : INNER  
> MANUAL JOIN FIELD(S) : id\_sample

## Voir le mapping des colonnes

L'un des aspects de Fuzible étant d'éviter de gérer le mapping des champs (une tâche bien connue des utilisateurs d'ETL), une option vous permet de voir le mapping qui va être réalisé entre les colonnes Source et Cible. Si certaines colonnes sont orphelines, une suggestion de requête vous sera proposée pour relier ces colonnes si c'est le comportement souhaité.

Fuzible va trouver automatiquement les correspondances de noms les plus proches pour relier les colonnes orphelines.

Vous pourrez alors copier-coller la requête proposer en lieu et place de la vôtre.

**Job Queries (MySQL SQL)** Sandbox

A Query must start with the table name in which data will be written in (will be created if non-existent).  
ex : MyTable:SELECT (...)

```
mynsychrotable:select * from bulkinsert
```

**Postgres Raspberry -> mynsychrotable**

**Query Analyzer**

- > Source Infos
- > Target Infos
- > Query Details
- > Search and Show Columns Mappings

**Synchro Query**

- > Transcoded for Target
- > Validity check for Synchro Query

**Execute Query**

- > Load Source Data (F5)
- > Run this individual query

**Scripting**

- > Get full header from query and copy/paste it
- > Dynamic Parameters
- > Basic Query Builder

**Advanced Query Scripting**

- > Add Cross-Connections Join
- > Create Dual Target

Mapping Information

Keep in mind that unmapped columns will be avoided until you set column aliases that matches the Target column names on your Query.  
Check the bottom of that page to get a Query suggestion.

**Mynsychrotable :**  
Columns will be mapped as follows : 40 record(s)

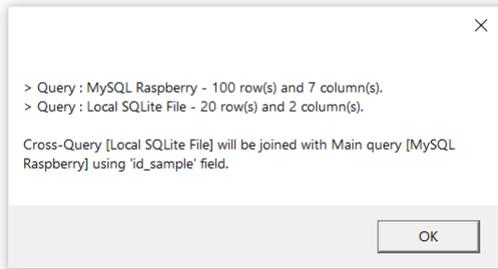
Source column	Link	Target column
id_bench_01	OK	id_bench_01
fi_bench_01	OK	fi_bench_01
nb_bench_01	OK	nb_bench_01
dt_bench_01	OK	dt_bench_01
id_bench_02	OK	id_bench_02
fi_bench_02	OK	fi_bench_02
nb_bench_02	OK	nb_bench_02
dt_bench_02	OK	dt_bench_02
id_bench_03	OK	id_bench_03
fi_bench_03	OK	fi_bench_03
nb_bench_03	OK	nb_bench_03
dt_bench_03	OK	dt_bench_03
id_bench_04	OK	id_bench_04
fi_bench_04	OK	fi_bench_04

Close

## Résultats

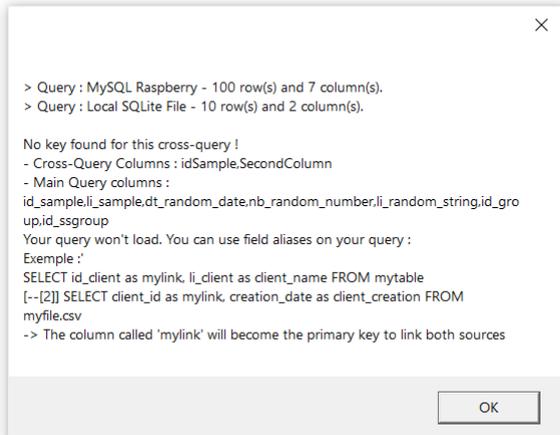
En vous rendant dans « Cross-Query/Search Join-Link and Check the Cross-Query », Fuzible va réaliser un test pour déterminer la viabilité de la requête jointe. Le résultat sera présenté ainsi :

```
mytargettable:select * from sample1  
[--[5]] select * from sample_table_3
```



En revanche, si aucun lien n'existe, voici ce qui sera affiché ; dans cet exemple, j'ai forcé le nom des colonnes de la deuxième Source de données, en mettant un nom « idSample » qui n'existe pas dans la première Source de données (id\_sample).

```
mytargettable:select * from sample1  
[->[5]] select FirstColumn as idSample, SecondColumn from sample_table_2
```



## Présentation des données

Le menu contextuel contient une option « Load Source Data » qui permet de consulter les résultats, que ce soit l'ensemble des données jointes, ou alors simplement les données de chaque requête jointe.

Dans le tableau ci-dessous, j'ai réalisé une requête croisée de type « LEFT JOIN », et chargé les données issues d'un croisement entre les 2 sources. Seuls les enregistrements « id\_sample » de 90 à 100 sont présents dans la deuxième connexion, ce qui explique que « SecondColumn » est vide dans les autres cas.

Show Source Data
— □ ×

Preview (rows) 
Data Successfully Loaded.
● With data analyzer

**Source Data**

DataTable Name : test  
 Namespace : sample1  
 Row Count : 100  
 Fields Count : 8

Additional Properties :  
 [test] CROSSJOIN\_LINK : id\_sample->id\_sample

id_sample	li_sample	dt_random_date	nb_random_number	li_random_string	id_group	id_ssgroup	SecondColumn
70	seven-zero	10/8/2020 1:10:49 PM	9.0625	piymoetghykoighe	001	000	
71	seven-one	10/19/2020 1:10:49 PM	31.0625	hsxnmgwrcwhwjnc	001	001	
72	seven-two	10/27/2020 1:10:49 PM	59.6250	kudzrucwhtiottx	002	002	
73	seven-three	12/19/2020 1:10:49 PM	33.6250	dbvsgxnmzmpxzgui	003	003	
74	seven-four	1/22/2021 1:10:49 PM	41.6250	ijfllkkoohnciam	004	004	
75	seven-five	11/13/2020 1:10:49 PM	29.1250	dsdfmlsimiywnfzc	005	000	
76	seven-six	8/31/2020 1:10:49 PM	16.4375	dxhkhbjeejdxnhj	006	001	
77	seven-seven	9/24/2020 1:10:49 PM	9.0625	mcvxnlymgsbozvw	007	002	
78	seven-eight	11/19/2020 1:10:49 PM	7.1250	cskbwhvhyiaoeay	008	003	
79	seven-nine	9/12/2020 1:10:49 PM	33.1250	ifkwmvqjpwkceh	009	004	
80	eight-zero	10/25/2020 1:10:49 PM	34.4375	qkgocfmpikdqbijk	000	000	
81	eight-one	9/17/2020 1:10:49 PM	12.4375	pfhogfkgkobyrgwn	001	001	
82	eight-two	12/2/2020 1:10:49 PM	16.8750	gudejxkntohaxpb	002	002	
83	eight-three	8/18/2020 1:10:49 PM	8.6875	hceaqwywnmswtthe	003	003	
84	eight-four	9/5/2020 1:10:49 PM	45.5000	zlwieuserdoejppv	004	004	
85	eight-five	8/3/2020 1:10:49 PM	18.3750	afegijjsqzavtx	005	000	
86	eight-six	8/26/2020 1:10:49 PM	20.3125	fyxklesjgdthyhc	006	001	
87	eight-seven	11/25/2020 1:10:49 PM	58.3125	nsbkycgcbfrdslmni	007	002	
88	eight-eight	9/28/2020 1:10:49 PM	22.0625	hqxcraatpibsqv	008	003	
89	eight-nine	12/23/2020 1:10:49 PM	25.5000	euaghggnorxryha	009	004	
90	nine-zero	8/19/2020 1:10:49 PM	38.0000	avqjiontdygcqce	000	000	
91	nine-one	12/26/2020 1:10:49 PM	54.8125	gqirlocjcbzkuscb	001	001	nine-one
92	nine-two	12/14/2020 1:10:49 PM	35.6250	rukcduncoiohokuk	002	002	nine-two
93	nine-three	10/27/2020 1:10:49 PM	13.8125	gjjbhaujifdbvp	003	003	nine-three
94	nine-four	1/1/2021 1:10:49 PM	26.8125	menocejauumpjizz	004	004	nine-four
95	nine-five	1/26/2021 1:10:49 PM	31.7500	dqmikhshesifnmzxl	005	000	nine-five
96	nine-six	10/14/2020 1:10:49 PM	38.1250	qseabnfytagdjj	006	001	nine-six
97	nine-seven	9/27/2020 1:10:49 PM	55.9375	wbwugaqrgpmsdkk	007	002	nine-seven
98	nine-eight	11/8/2020 1:10:49 PM	13.3125	hdozehujkschidboa	008	003	nine-eight
99	nine-nine	8/6/2020 1:10:49 PM	55.8750	nvlekjyrrfapvuc	009	004	nine-nine
100	one-zero-zero	12/12/2020 1:10:49 PM	23.3125	ioxpaxpxpnyfyrrpn	000	000	one-zero-zero

[Export content as CSV](#)

A l'inverse, un « INNER JOIN » aurait donné ceci :

Show Source Data
— □ ×

Preview (rows) 
Data Successfully Loaded.
● With data analyzer

**Source Data**

DataTable Name : test  
 Namespace : sample1  
 Row Count : 10  
 Fields Count : 8

Additional Properties :  
 [test] CROSSJOIN\_LINK : id\_sample->id\_sample

id_sample	li_sample	dt_random_date	nb_random_number	li_random_string	id_group	id_ssgroup	SecondColumn
91	nine-one	12/26/2020 1:10:49 PM	54.8125	gqirlocjcbzkuscb	001	001	nine-one
92	nine-two	12/14/2020 1:10:49 PM	35.6250	rukcduncoiohokuk	002	002	nine-two
93	nine-three	10/27/2020 1:10:49 PM	13.8125	gjjbhaujifdbvp	003	003	nine-three
94	nine-four	1/1/2021 1:10:49 PM	26.8125	menocejauumpjizz	004	004	nine-four
95	nine-five	1/26/2021 1:10:49 PM	31.7500	dqmikhshesifnmzxl	005	000	nine-five
96	nine-six	10/14/2020 1:10:49 PM	38.1250	qseabnfytagdjj	006	001	nine-six
97	nine-seven	9/27/2020 1:10:49 PM	55.9375	wbwugaqrgpmsdkk	007	002	nine-seven
98	nine-eight	11/8/2020 1:10:49 PM	13.3125	hdozehujkschidboa	008	003	nine-eight
99	nine-nine	8/6/2020 1:10:49 PM	55.8750	nvlekjyrrfapvuc	009	004	nine-nine
100	one-zero-zero	12/12/2020 1:10:49 PM	23.3125	ioxpaxpxpnyfyrrpn	000	000	one-zero-zero

## Les données de la requête jointe :

Show Source Data

Preview (rows)  Data Successfully Loaded.  With data analyzer

**Source Data**

DataTable Name : sample\_table\_2  
Namespace : sample\_table\_2  
Row Count : 10  
Fields Count : 2

id_sample	SecondColumn
100	one-zero-zero
99	nine-nine
98	nine-eight
97	nine-seven
96	nine-six
95	nine-five
94	nine-four
93	nine-three
92	nine-two
91	nine-one

### Requêtes jointes, mode avancé

Ce mode permet par exemple de comparer le schéma de 2 bases de données (pour établir un rapport des différences entre l'une et l'autre, par exemple)

Par défaut, la fonctionnalité « Requêtes-Jointes » est simple à utiliser. Mais comme toute fonctionnalité automatique, elle est rigide. On peut néanmoins la rendre plus souple en filtrant les résultats.

Cet exemple compare les tables présentes dans une BDD de production et une BDD de pré-production (Postgres)

A Query must start with the mail address(es) for which you want to send data to (addresses have to be separated by a ',').  
ex : mymail@gmail.com,anymail@yahoo.frSELECT (...)

```
mail@mail.com:select table_name, table_type,  
'Oui' as presente_en_prod  
from information_schema.tables As Tables_absentes_en_prod_ou_en_preprod  
where table_name not like 'pg%' and table_type <> 'VIEW'  
order by table_type, table_name  
[<>[75]WHERE presente_en_prod IS NULL OR presente_en_preprod IS NULL]  
select table_name, table_type,  
'Oui' as presente_en_preprod  
from information_schema.tables  
where table_name not like 'pg%' and table_type <> 'VIEW'  
order by table_type, table_name
```

Constatez que la zone encadrée de script [<>[75]...] est augmentée d'une condition qui permet de réaliser la comparaison entre les 2 jeux de données.

- ➔ Cette façon de scripter permet, une fois les données rassemblées, de filtrer les résultats selon le filtre indiqué dans la zone de script

Si la cible est une adresse mail, le résultat se présentera ainsi :

#### Tables absentes en prod ou en préprod : 21 record(s)

Table name	Table type	Présente en prod	Présente en préprod
param_webservices	BASE TABLE		Oui
cometepreprod_cii_agents	FOREIGN TABLE		Oui
cometepreprod_cii_mois_agent	FOREIGN TABLE		Oui
cometepreprod_planning_cloture	FOREIGN TABLE		Oui
cometepreprod_ressources	FOREIGN TABLE		Oui
dbg_absence_comete	BASE TABLE	Oui	
dbg_absence_rhpi	BASE TABLE	Oui	
dbg_calendar	BASE TABLE	Oui	
dbg_contrat_comete	BASE TABLE	Oui	
dbg_contrats_rhpi	BASE TABLE	Oui	
dbg_generateur_documents	BASE TABLE	Oui	

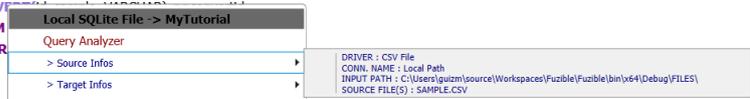
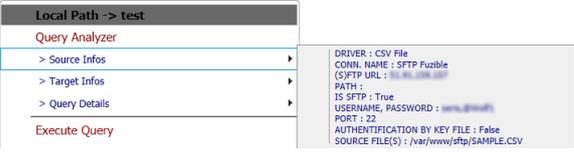
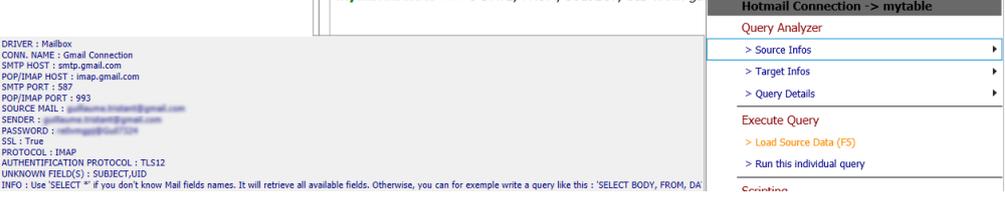
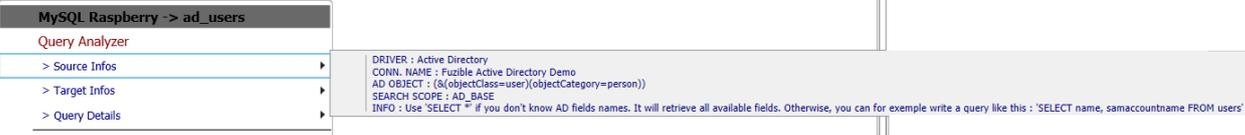
## Menu Contextuel des requêtes

Le principe de la requête SQL pour faire une synchronisation ou une réplication de données peut représenter un niveau d'abstraction assez important. C'est pourquoi Fuzible possède un menu contextuel très utile, accessible lorsque la souris est sur une requête. Un clic-droit et il apparaît pour vous permettre :

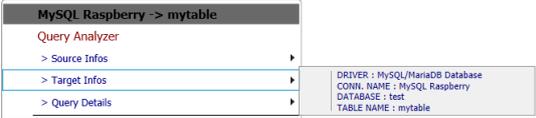
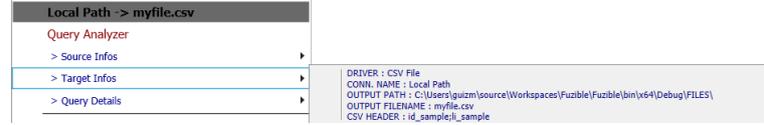
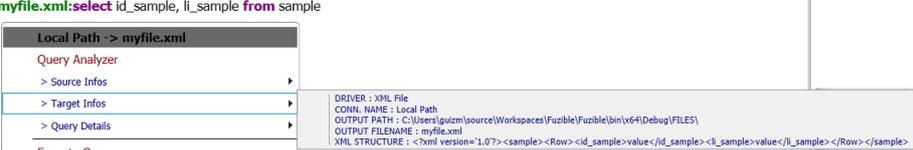
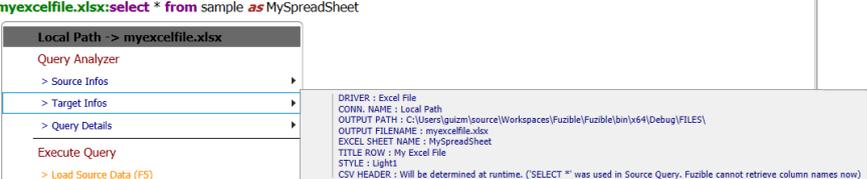
- D'obtenir des informations avancées sur la requête
- De tester la validité d'un Job paramétré en mode de synchronisation
- De jouer l'une ou l'autre des requêtes de du Job sans nécessairement toutes les jouer
- De bénéficier d'un assistant de script pour les fonctions avancées

# QUERY ANALYZER - Source Infos

## Résumé de la connexion source

<b>FILE</b>	<p>Pour un fichier local, le chemin et le type de connexion</p> <pre>MyTutorial:SELECT id_sample, SUBSTRING(li_sample, 0, 5) as li_sample, id_group as RenommedColumn, dt_random_date, li_sample as li_test,</pre>  <p>Pour un (S)FTP, la configuration du (S)FTP et le chemin sur le serveur</p> <pre>test:select * from /var/www/sftp/SAMPLE.CSV</pre> 
<b>BDD</b>	<p>Pour une connexion SQL, le pilote, et le nom de la base de données interrogée</p> <pre>MyTutorial.csv:SELECT user as userColumn, connstring_id as idColumn, connstring_name as nameColumn from user_connstrings WHERE user LIKE '{?}'</pre> 
<b>WS</b>	<p>Indique tout le cheminement réalisé par Fuzible pour interroger un webservice. Permet de contrôler l'URL, le type d'autorisation</p> <pre>sample1:SELECT * FROM /ws/public/sample_data</pre> 
<b>MAIL</b>	<p>Résumé des paramètres utilisés pour se connecter au serveur mail</p> <pre>mytable:select TOP 3 DATE, FROM, SUBJECT, UID from gui</pre> 
<b>AD</b>	<p>La requête de recherche, le périmètre</p> <pre>ad_users:SELECT * FROM users</pre> 

# QUERY ANALYZER - Target Infos

<p>BDD</p>	<ul style="list-style-type: none"> <li>- Le nom de la base de données « cible »</li> <li>- La table de destination</li> <li>- Les éventuelles commandes qui ont été définies (pré-post job)</li> </ul> <pre>mytable:select * from sample</pre> 
<p>FILE</p>	<ul style="list-style-type: none"> <li>- Toutes les informations relatives à la cible : le répertoire, et, en cas d' « OUTPUT » avec pattern (création de plusieurs fichiers), le comportement qui sera adopté.</li> <li>- Les éventuelles commandes qui ont été définies (pré-post job)</li> </ul> <p>CSV :</p> <pre>myfile.csv:select id_sample, li_sample from sample</pre>  <p>XML :</p> <pre>myfile.xml:select id_sample, li_sample from sample</pre>  <p>EXCEL :</p> <pre>myexcelfile.xlsx:select * from sample as MySpreadSheet</pre> 
<p>WS</p>	 <p>Indique tout le cheminement réalisé par Fuzible pour envoyer des données dans un Webservice. Permet de contrôler l'URL et le comportement de l'appel à l'API.</p>
<p>MAIL</p>	<p>A Query must start with the mail address(es) for which you want to send data to (addresses have to be separated by a '); ex: mymail@gmail.com,anymail@yahoo.fr:SELECT (...)</p> <pre>fuzible@fuzible-app.com:select id_sample,li_sample,dt_random_date,nb_random_number,li_random_string,id_group,id_ssgroup from sample1 as MyChart WHERE id_sample &gt; {?1}</pre>  <p>Indique :</p> <ul style="list-style-type: none"> <li>- Le sujet du mail (récupéré à partir de la description du Job)</li> <li>- Les destinataires (OUTPUT de la requête)</li> <li>- Le nom du tableau produit (récupéré à partir de l'alias de la requête : <i>SELECT * from matable AS My_Chart -&gt; Affichera My Chart</i> (Les underscores sont systématiquement remplacés par un espace)</li> </ul>

AD

```
users:select user as name, param1 as description from user_parameters
```



```
DRIVER : Active Directory  
CONN. NAME : Fuzible Active Directory Demo  
AD QUERY : (&{objectClass=user}{objectCategory=person}(name=[DATA FROM : name])  
AD OBJECT TARGET : users
```

Indique :

- L'objet AD concerné
- La requête de recherche qui sera effectuée

## QUERY ANALYZER - View Data

Cette option ouvre une nouvelle fenêtre qui vous permettra plusieurs choses :

- Charger les données source pour les prévisualiser
- Avoir des informations sur chaque champ source
- Tester le mode de synchronisation

En cliquant sur « Load Source Data », le programme chargera les données source et affichera une preview de 500 lignes (paramétrable).

Si le Job est un Job de synchronisation, chaque onglet (Target, Insert, Update, Delete) vous affichera tout ce que la synchronisation va faire.

Additional Properties :  
[AX\_dim\_plancomptable] SYNCHRO\_PRIMARY\_KEY : id\_comptecomptable

id_comptecomptable	id_racine_comptecomptable	il_comptecomptable	id_type	il_type	SYNCHRO_TAG	DBNAME	DTLOAD
101000	1	CAPITAL APPELE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
10100000	1	Capital souscrit (Sociétés de capitaux -	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
101100	1	CAPITAL SOUSCRIT NON APPELE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
104100	1	PRIME D EMISSION	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
104200	1	PRIME DE FUSION	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
104300	1	PRIME D APPORT	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
106100	1	RESERVE LEGALE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
106110	1	RESERVE LEGALE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
106200	1	Reserves indisponibles.	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
106300	1	RESERVES CONTRACTUELLES	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
106400	1	Reserves réglementées	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
106480	1	RESERVE SPEC ART 238B CGI	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
106800	1	AUTRES RESERVES	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
109000	1	ACTIONNAIRE CAPITAL SOUSCRIT NON AP	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
110000	1	REPORT A NOUVEAU BENEFICE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
119000	1	REPORT A NOUVEAU PERTE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
120000	1	RESULTAT EXERCICE BENEFICE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
121000	1	RESULTAT PROVISOIRE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
129000	1	RESULTAT EXERCICE PERTE	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
129100	1	ACOMPTES SUR DIVIDENDES EN ATTENTE D'AFFECTATION	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
13100000	1	Reserve légale	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
13810000	1	Autres réserves disponibles	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
13821000	1	Reserve pour l'impôt sur la fortune	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
14110000	1	Résultats reportés en instance d'affectation	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
14120000	1	Résultats reportés (affectés)	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
145000	1	AMORTISSEMENTS DEROGATOIRES	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
151100	1	PROVISION POUR LITIGES	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
151110	1	LABOR TERMINATION COSTS	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
151120	1	SOC - Divers provisions litige	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
151130	1	SOC - Divers provisions litige	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM
151800	1	AUTRES PROVISIONS RISQUES	3	Résultat	I	AXPROD	5/22/2020 11:44:59 AM

Export Content as CSV

Il est possible de définir le nombre de lignes à afficher dans la fenêtre de prévisualisation, mais également de faire un export simple et rapide des données au format CSV (utile pour faire des comparaisons rapides de données)

Par ailleurs, en cliquant sur « With Data Analyzer », vous pourrez consulter le détail de chaque champ de la source. Utile pour comprendre comment Fuzible interprète les types de données.

**Data Preview** Data Successfully Loaded. ● With Data Analyzer

Source Data Target Data To Insert To Update To Delete

DataTable Name : AXPROD  
Namespace : A  
Row Count : 2502  
Preview : 500  
Fields Count : 8

id_comptecomptable	id_racine_comptecomptable	il_comptecomptable	id_type	li_type	SYNCHRO_TAG	DBNAME	DTLOAD
101000	1	Name : id_racine_comptecomptable - Original Definition : LEFT(A.MAINACCOUNTID,1) as id_racine_comptecomptable - SQL Type : INT - Linq Type : Int64 - Allow Null : False - Is Unique : False - Primary Key : [id_comptecomptable]				PROD	5/19/2020 7:22:03 PM
10100000	1					PROD	5/19/2020 7:22:03 PM
101100	1					PROD	5/19/2020 7:22:03 PM
104100	1					PROD	5/19/2020 7:22:03 PM
104200	1					PROD	5/19/2020 7:22:03 PM
104300	1					PROD	5/19/2020 7:22:03 PM
106100	1					PROD	5/19/2020 7:22:03 PM
106110	1					PROD	5/19/2020 7:22:03 PM
106200	1				PROD	5/19/2020 7:22:03 PM	
106300	1	RESERVES CONTRACTUELLES	3	Résultat	I	AXPROD	5/19/2020 7:22:03 PM

A noter que chaque sous-requête, chaque requête jointe apparaissent dans le menu contextuel et les données de chacune d'entre elles, chargées indépendamment du reste. Dans l'exemple ci-dessous, la requête Source contient une sous-requête, un UNION, ainsi qu'une requête jointe.

```
myTarget:select *
from sample_table_1 as a
inner join sample_table_2 as b ON a.id_sample = b.id_sample
inner join (select * from sample_table_3) as subQ on a.id_sample = subQ.FirstColumn
UNION
select * from sample_table_4
[--[11]]
select * from sample as crossQuery
```

Local Path -> myTarget

Query Analyzer

- > Source Infos
- > Target Infos
- > Query Details
- + Sub-Query : subQ
- + Union Query 1
- + Cross-Query (MySQL Raspberry)

Execute Query

- > Load Source Data (F5)
- > Run this individual query

> Load Source Data

- Fields
- Tables
- Conditions
- Group By
- Order By

## QUERY ANALYZER - Query Details

Permet de déconstruire intégralement une requête pour vérifier qu'elle est bien conforme, et qu'elle n'a pas d'erreurs de syntaxe.

*Par exemple, on peut parcourir tous les champs...*

BD7 (bleu\_ro) -> AX\_dim\_plancomptable

Query Analyzer

- > Source Infos
- > Target Infos
- > Load Source Data (F5)
- > Query Details
- + Sub-Query : B

Synchro Query

- > Transcoded for Target
- > Validity check for Synchro Query

Execute Query

- Fields
- Tables
- Conditions
- Group By
- Order By
- Additional Informations

- > id\_comptecomptable
- > id\_racine\_comptecomptable
- > il\_comptecomptable
- > id\_type
- > li\_type

```
> RAW : (A.MAINACCOUNTID) as id_comptecomptable
> FIELD : (A.MAINACCOUNTID)
> TABLE : MAINACCOUNT
> TABLE ALIAS : A
> INDEX : 0
```

## Toutes les tables et comprendre leurs jointures...

The screenshot shows the 'Query Analyzer' interface for a query named 'BD7 (bleu\_ro) -> AX\_dim\_plancomptable'. The left pane shows a tree view with 'Query Details' expanded to '+ Sub-Query : B'. A right pane shows a tree view with 'Tables' expanded to '> A', which is further expanded to '> B'. A tooltip is visible over the '> B' item, displaying the following SQL snippet:

```
> NAME : SUBQUERY  
> JOIN-TYPE : INNER JOIN  
> JOIN-LINK : A.RECID = B.RECID  
> SUBQUERY : True
```

Below the tree view, there are sections for 'Synchro Query' and 'Execute Query'. At the bottom, there is a button labeled '> Load Source Data'.

## Consulter les sous-requêtes et les tester...

The screenshot shows the 'Query Analyzer' interface for the same query. The left pane shows the tree view with '+ Sub-Query : B' selected. The right pane shows the 'Load Source Data' menu, which includes options for 'Fields', 'Tables', 'Conditions', 'Group By', and 'Order By'.

## Vérifier les erreurs de syntaxe...

Dans cet exemple, la requête contient une transformation inconnue : ERROR(id\_sample)

The screenshot shows the 'Query Analyzer' interface with a SQL query in the main pane. The query is:

```
MyTutorial:SELECT id_sample, SUBSTRING(li_sample, 0, 5) as li_sample, id_group as RenommedColumn, dt_random_date,  
li_sample as li_test,  
CONVERT(id_sample, VARCHAR) as convertid,  
ERROR(id_sample) as ErrorDemo  
FROM SAMPLE.CSV  
WHERE id_sample >= 5
```

The left pane shows the 'Query Analyzer' tree view with 'Query Details' expanded to '> Query Details'. The right pane shows the 'Load Source Data' menu. A tooltip is visible over the 'ERROR : ERROR(id\_sample)' item, displaying the following error message:

```
DESCRIPTION : Unrecognized SQL Function.  
> ERROR : ERROR(id_sample)
```

La détection des erreurs de syntaxe est particulièrement utile pour les requêtes « non-SGBD », cela vous permet, si la requête échoue, de comprendre pourquoi elle n'a pas fonctionné.

Dans le cas d'une requête sur un SGBD, la détection est essentiellement informative, dans la mesure où Fuzible ne connaît pas tous les cas de figure tordus qu'une requête peut comporter.

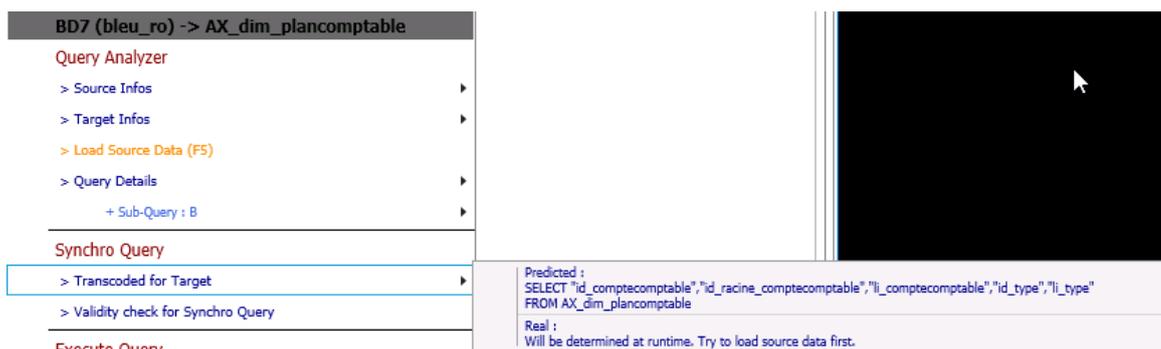
### EXECUTE QUERY - Run this individual Query

Vous pouvez jouer uniquement une requête spécifique plutôt que tout le Job. En choisissant cette option, seule la requête sur laquelle vous êtes positionné sera jouée.

Dans ce mode, l'onglet « LOG » ne se remplit pas et l'interface graphique est « bloquée » le temps de l'exécution. Une fois le traitement effectué, un écran de LOG s'affiche et vous présente le résultat.

### SYNCHRO-QUERY – Transcoded for Target

Lorsque vous avez saisi une requête pour un job de synchro, vous pouvez avoir envie de tester le bon fonctionnement du transcodage de la requête source appliquée à la cible. Ce menu vous propose de voir la requête telle qu'elle sera réalisée dans la cible.

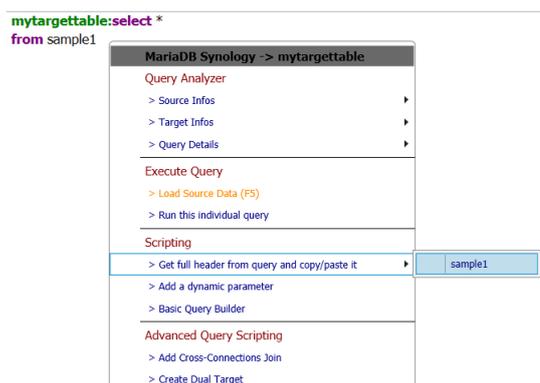


### SYNCHRO-QUERY – Validity check for Synchro Query

Cette option permet tout simplement de tester la validité de la requête de synchro.

### SCRIPTING - Get Full Header Query and copy/paste it

Il peut être fastidieux de saisir manuellement tout l'entête d'une table SQL ou d'un fichier (si on veut par exemple ne pas faire un SELECT \* mais un SELECT avec le nom des champs). En cliquant ici, Fuzible va récupérer tous les champs d'entête de la requête si celle-ci possède des jointures, vous pouvez déterminer à partir de quelle table/fichier/webservice... vous souhaitez récupérer l'entête :



Une fois la récupération terminée, le programme copie l'entête dans le presse-papier, vous êtes alors invité à le coller dans votre requête :

```
mytargettable:select id_sample,li_sample,dt_random_date,nb_random_number,li_random_string,id_group,id_ssgroup
from sample1
```

## SCRIPTING – Transformations

Dans le cas où votre connexion Source n'est pas une base de données, le menu « Transformations » vous rappelle toute la syntaxe SQL disponible pour manipuler les données. Celle-ci est évidemment disponible également lorsque vous saisissez une requête : l'assistant vous fait des suggestions en fonction de ce que vous tapez.

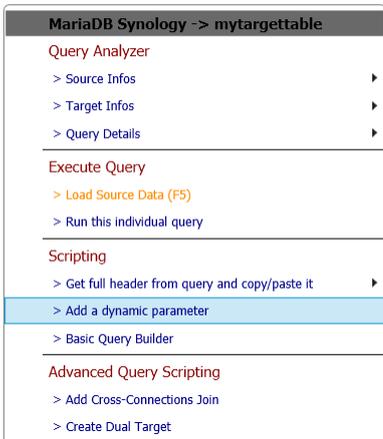
The screenshot displays the SHS Fuzible Data Replicator, Synchronizer interface. The main window shows a 'Job Selection' section with 'GUIZM' and '[10] Demo Job 001'. Below this, the 'Job Configuration' section indicates 'Source : CSV File' and 'Target : SQLite Database'. The 'Job Queries' section is active, showing a query editor with the following text: 'MyTutorial:SELECT id\_sample, SUBSTRING(li\_sample, 0, 5) as li\_sample as li\_test, CONVERT(id\_sample, VARCHAR(255)) as converted\_id FROM SAMPLE.C WHERE id\_sam...'. A context menu is open over the query, listing various SQL functions and operations. The menu items include: 'Local SQLite File -> MyTutorial', 'Query Analyzer', 'Source Infos', 'Target Infos', 'Query Details', 'Execute Query', 'Load Source Data (F5)', 'Run this individual query', 'Scripting', 'Get full header from query and copy/paste it', 'Add a dynamic parameter', 'Transformations', 'Basic Query Builder', 'Advanced Query Scripting', 'Add Cross-Connections Join', and 'Create Dual Target'. The 'Transformations' menu item is highlighted. To the right of the menu, a list of SQL functions is displayed, including: CASE field WHEN 'value1' THEN 'value2' (...) ELSE 'value3' END, COALESCE(field, 'replacementValue'), CHARINDEX(field, 'expToFind'), LENGTH(field), CONCAT(field1, field2,...), CONVERT(field, SQL type), DISTINCT (ex : SELECT DISTINCT \* FROM [...]), TOP (ex : SELECT TOP 100 \* FROM), LIMIT (ex : SELECT \* FROM myTable LIMIT 100), AVG(field), SUM(field), MIN(field), MAX(field), COUNT(field), GROUP BY (ex : SELECT id\_client, li\_client, SUM(nb\_amount) FROM [...] GROUP BY id\_client, li\_client), ISNULL(field, 'replacementValue'), INNER JOIN, LEFT JOIN, RIGHT JOIN, JOIN, OUTER JOIN, LPAD(field, paddedLength, 'padString'), LTRIM(field), LOWER(field), ORDER BY (ex : SELECT \* FROM [...] ORDER BY field ASC), RPAD(field, paddedLength, 'padString'), RTRIM(field), REPLACE(field, 'ValueToReplace', 'ReplacementValue'), SUBSTRING(field, startIndex, length), UPPER(field), WHERE (ex : SELECT \* FROM [...] WHERE field = 'value'), AS (ex : SELECT id\_client AS myClient [...]), UNION (ex : SELECT field FROM table1 UNION SELECT field FROM table2), and \* (ex : SELECT \* FROM [...]).

Right-click on a query for advanced operations or press 'F5' to

## SCRIPTING – Add a Dynamic Parameter

Vous pouvez ajouter un paramètre dynamique dans votre requête. Par exemple, ici, je veux variabiliser la condition « WHERE id\_sample » :

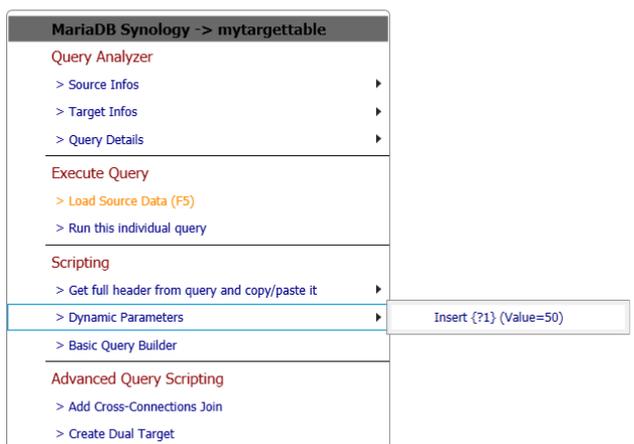
```
mytargettable:select id_sample,li_sample,dt_random_date,nb_random_number,li_random_string,id_group,id_ssgroup
from sample1 WHERE id_sample >
```



The screenshot shows the MariaDB Synology interface for a query named 'mytargettable'. The 'Scripting' menu is open, and the option '> Add a dynamic parameter' is highlighted in blue. Other options in the menu include 'Query Analyzer', 'Execute Query', 'Basic Query Builder', and 'Advanced Query Scripting'.

Après avoir entré la valeur souhaitée, Fuzible va coller le paramètre dynamique dans la requête :

```
mytargettable:select id_sample,li_sample,dt_random_date,nb_random_number,li_random_string,id_group,id_ssgroup
from sample1 WHERE id_sample > {?1}
```



The screenshot shows the same interface as before, but now the 'Dynamic Parameters' option in the 'Scripting' menu is highlighted. A tooltip is visible next to it, displaying 'Insert {?1} (Value=50)'. The query in the background now includes the dynamic parameter '{?1}' in the WHERE clause.

...puis compléter le champ dans l'onglet de paramétrage général :

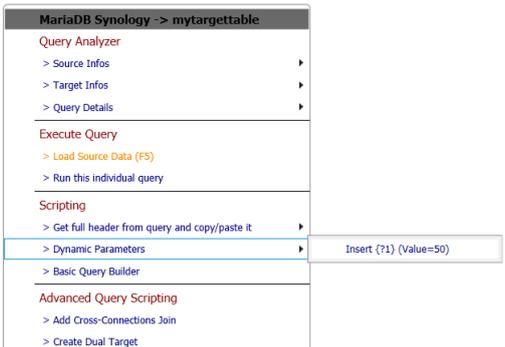
Dynamic Parameters  ?

You can write any text or any available command. Each parameter must be separated by a semicolon (;)  
You can use them anywhere (queries, text fields, connections) by referencing them like this : {?1}, {?2}...

[View Job With Replaced Values](#)

Si des paramètres dynamiques ont déjà été saisis, le menu se présentera ainsi :

```
mytargettable:select id_sample,li_sample,dt_random_date,nb_random_number,li_random_string,id_group,id_ssgroup
from sample1 WHERE id_sample > {?1}
```



The screenshot shows the same interface as before, but now the 'Dynamic Parameters' option in the 'Scripting' menu is highlighted. A tooltip is visible next to it, displaying 'Insert {?1} (Value=50)'. The query in the background now includes the dynamic parameter '{?1}' in the WHERE clause.

### **SCRIPTING – Basic Query Builder**

Un simple assistant à la création d'une requête. Il est entendu que ce mode ne permet absolument pas de réaliser des requêtes avancées.

### **ADVANCED QUERY SCRIPTING – Add Cross-Connections Join**

C'est ici que vous pourrez ouvrir le menu de construction d'une requête jointe.

### **ADVANCED QUERY SCRIPTING – Create Dual Target**

C'est ici que vous pourrez ouvrir le menu de construction d'une requête multi-cibles

## Onglet Log Viewer

En fonction du niveau de LOG choisi dans l'onglet « Job Configuration » vous verrez plus ou moins de détails dans le Viewer.

Ce que vous y voyez se reporte systématiquement dans les fichiers de LOG créés sur le disque dur.

A la fin de l'exécution d'un Job, un message apparaîtra sur l'écran indiquant son statut.

Tour d'horizon des possibilités de debug :

- Le Query Analyzer et le chargement des données « Source » (menu contextuel des requêtes)
- Le mode « SIMULATION », qui exécute le Job sans réaliser la moindre opération dans la cible, il se contente d'afficher tout ce qu'il va y faire (voir un peu plus bas)
- Le LOG général : assez verbeux, il peut vous aider à comprendre ce qui pose un problème dans l'exécution d'un Job, vous pouvez d'ailleurs le paramétrer en « Détaillé » pour avoir un maximum d'informations.

Consultez les fichiers produits par le Job, vous en avez 3 :

- 1) Le fichier LOG, qui affiche tout ce qu'on voit dans l'onglet « LOG Viewer »
- 2) Le fichier « QUERIES » qui affiche toutes les requêtes qui ont planté
- 3) Le fichier « DEBUG » qui offre un niveau d'information supérieur à chaque erreur rencontrée

Exemple d'une exécution réussie :

The screenshot shows the SHS Fuzible Data Replicator interface. At the top, the title bar reads "SHS Fuzible Data Replicator, Synchronizer - Threads Running : 0 / CPU : 0 % / RAM : 147 Mb / 17 Mb". Below the title bar is a menu bar with "File", "Configuration", "Tools", and "Help". The main area is divided into several sections:

- Job Selection:** A dropdown menu shows "[10] Demo Job 001". To the right is a button "Create/manage a Multi-steps Job" with a sub-button "Create new Step".
- Job Configuration:** A row of tabs includes "Source : CSV File", "Target : SQLite Database", "Queries", and "Log Viewer" (which is selected).
- Log Viewer:** A large text area displays the following log output:

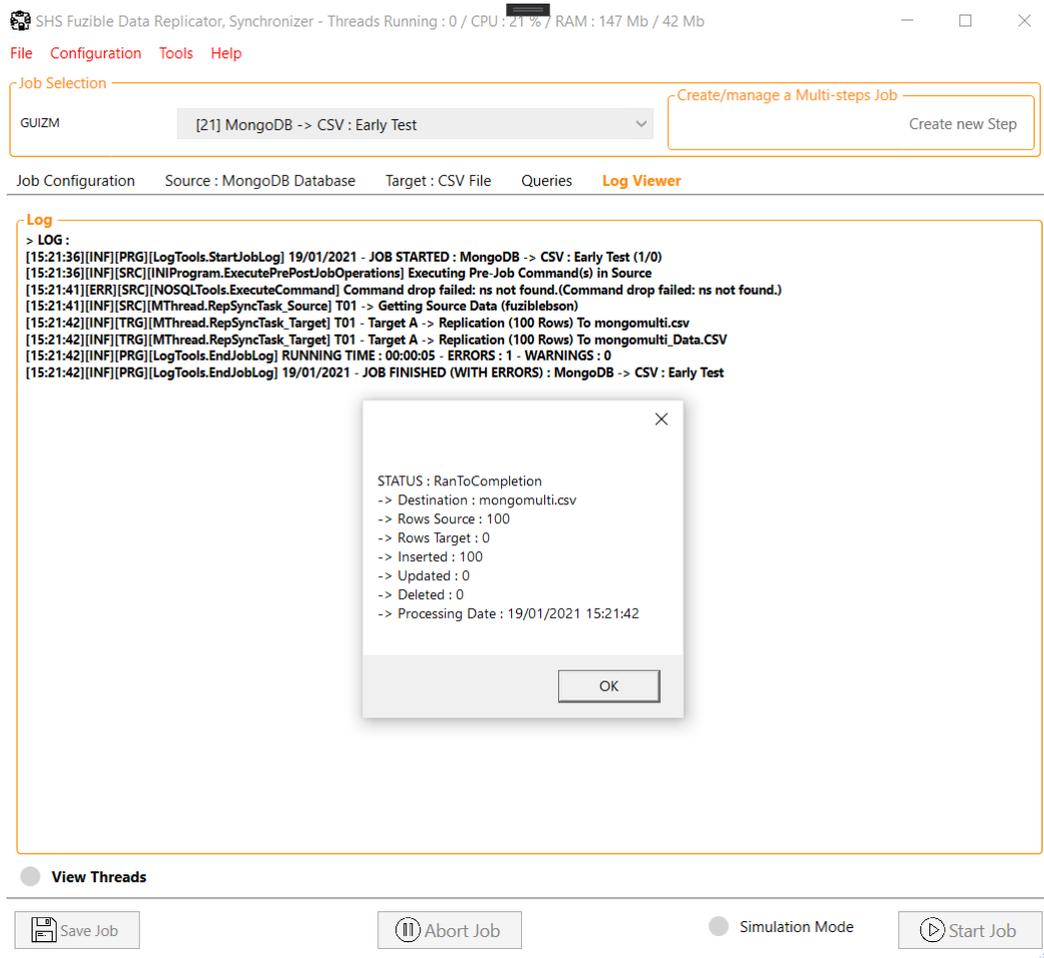
```
> LOG:
[15:19:46][INF][PRG][LogTools.StartJobLog] 19/01/2021 - JOB STARTED : Demo Job 001 (1/0)
[15:19:46][INF][SRC][MThread.RepSyncTask_Source] T01 -> Getting Source Data (SAMPLE.CSV)
[15:19:46][INF][TRG][MThread.RepSyncTask_Target] T01 - Target A -> Replication (51 Rows) To MyTutorial
[15:19:46][INF][TRG][SQLTools.InsertDataInBDD] insert data from dataset on MyTutorial(51)
[15:19:47][INF][PRG][LogTools.EndJobLog] RUNNING TIME : 00:00:00 - ERRORS : 0 - WARNINGS : 0
[15:19:47][INF][PRG][LogTools.EndJobLog] 19/01/2021 - JOB FINISHED (NO ERRORS) : Demo Job 001
```

A modal dialog box is overlaid on the log viewer, displaying the following status information:

```
STATUS : RanToCompletion
-> Destination : MyTutorial
-> Rows Source : 51
-> Rows Target : 0
-> Inserted : 51
-> Updated : 0
-> Deleted : 0
-> Processing Date : 19/01/2021 15:19:46
```

At the bottom of the interface, there are several buttons: "View Threads", "Save Job", "Abort Job", "Simulation Mode" (with a radio button), and "Start Job".

## Exemple d'une exécution ratée ; le LOG s'affiche en gras :



SHS Fuzible Data Replicator, Synchronizer - Threads Running : 0 / CPU : 21 % / RAM : 147 Mb / 42 Mb

File Configuration Tools Help

Job Selection

GUIMZM [21] MongoDB -> CSV : Early Test Create/manage a Multi-steps Job Create new Step

Job Configuration Source : MongoDB Database Target : CSV File Queries Log Viewer

Log

```
> LOG :
[15:21:36][INF][PRG][LogTools.StartJobLog] 19/01/2021 - JOB STARTED : MongoDB -> CSV : Early Test (1/0)
[15:21:36][INF][SRC][INIProgram.ExecutePrePostJobOperations] Executing Pre-Job Command(s) in Source
[15:21:41][ERR][SRC][NOSQLTools.ExecuteCommand] Command drop failed: ns not found.(Command drop failed: ns not found.)
[15:21:41][INF][SRC][MThread.RepSyncTask_Source] T01 -> Getting Source Data (fuziblebson)
[15:21:42][INF][TRG][MThread.RepSyncTask_Target] T01 - Target A -> Replication (100 Rows) To mongomulti.csv
[15:21:42][INF][TRG][MThread.RepSyncTask_Target] T01 - Target A -> Replication (100 Rows) To mongomulti_Data.CSV
[15:21:42][INF][PRG][LogTools.EndJobLog] RUNNING TIME : 00:00:05 - ERRORS : 1 - WARNINGS : 0
[15:21:42][INF][PRG][LogTools.EndJobLog] 19/01/2021 - JOB FINISHED (WITH ERRORS) : MongoDB -> CSV : Early Test
```

STATUS : RanToCompletion  
-> Destination : mongomulti.csv  
-> Rows Source : 100  
-> Rows Target : 0  
-> Inserted : 100  
-> Updated : 0  
-> Deleted : 0  
-> Processing Date : 19/01/2021 15:21:42

View Threads

Save Job Abort Job Simulation Mode Start Job

## Exécution d'un Job

Une fois le Job paramétré, sauvegardé et testé, vous pouvez le jouer et l'interrompre si besoin.

Ces boutons se trouvent en bas de l'interface. Par ailleurs, vous constaterez que sur l'onglet « Job Configuration », vous avez un petit bouton « Simulation Mode » juste au-dessus. Celui-ci permet d'inscrire dans le fichier de LOG l'intégralité de ce qui va être réalisé, mais aucune donnée ne sera écrite dans la cible pour autant.

Ce LOG a l'avantage de ne pas réellement jouer le Job, donc de ne pas prendre de risque et d'éventuellement compromettre la cible.

Par ailleurs, ce mode affiche l'intégralité des requêtes SQL passées, ce qui peut être très utile dans le cas de réplication de données de SGBD à SGBD : permet de récupérer tout le code SQL « INSERT », « DELETE », « UPDATE ».

● Simulation Mode Start Job

## Exemple d'output du mode « Simulation » (Job qui copie des données d'un fichier vers une base de données)

```
new 2 x WHATSNEW TXT x HELP_ENVM L x new 1 x GJZM_10_Queries_20210119152750.txt x
1 [SIMULATION MODE] WILL POST-WORK INPUT FILE(S) -RIEN : C:\Users\guizm\source\Workspaces\Fuzible\Fuzible\bin\x64\Debug\FILES\SAMPLE.CSV
2 [SIMULATION MODE] WILL POST-WORK INPUT FILE(S) ON (S) FTP - RIEN : SAMPLE.CSV
3 [SIMULATION MODE] SELECT COUNT(*) FROM sqlite_master WHERE type = 'table' AND tbl_name = 'MyTutorial';
4 [SIMULATION MODE] SELECT il.cid, il.name, il.name, il.type, '0', il."notnull", il.name, il.pk FROM sqlite_master AS m, pragma_table_info(m.name) AS il WHERE m.type='table' AND m.name = 'MyTutorial' AND pk =
5 [SIMULATION MODE] SELECT tbl.cid, tbl.name, il."from" || ' ' || il."to" as name_constraint, 0, tbl."notnull", il.seq, il.seq, il."table", il."to" FROM sqlite_master AS m INNER JOIN pragma_foreign_key_list('
6 [SIMULATION MODE] SELECT il.name, il.type, '0', il."notnull", il."notnull", il.type, '0', il.dflt_value, 'NO' as UniqueKey, 'NO' as IsKey FROM sqlite_master AS m, pragma_table_info(m.name) AS il WHERE m.type
7 [SIMULATION MODE] DELETE FROM "MyTutorial";
8 [SIMULATION MODE] SELECT COUNT(*) FROM "MyTutorial";
9 [SIMULATION MODE] INSERT INTO "MyTutorial" ("id_sample","li_sample","RenommedColumn","li_test") VALUES (50,'five-', '000','five-');INSERT INTO "MyTutorial" ("id_sample","li_sample","RenommedColumn","li_test")
10 [SIMULATION MODE] INSERT INTO "MyTutorial" ("id_sample","li_sample","RenommedColumn","li_test") VALUES (82,'eight', '002','eight');INSERT INTO "MyTutorial" ("id_sample","li_sample","RenommedColumn","li_test")
11
```

Par ailleurs, si vous consultez le fichier de LOG « Queries » associé, vous aurez toutes les requêtes en clair, notamment les INSERT. Ce scénario est très utile si vous avez installé l'application en local, et que vous souhaitez envoyer des données dans une BDD inaccessible en local.

Vous pouvez donc uniquement générer les requêtes via le mode « simulation », et vous connecter ensuite au serveur distant pour intégrer les données via le code INSERT produit par Fuzible.

## Application « Service »

Le programme est fourni avec une application type « service » : Ce service fonctionne en harmonie avec l'application client (en mode SQL), ainsi que le planificateur de Jobs.

L'application crée et purge automatiquement la table SQL utile à son fonctionnement. S'agissant d'une application « console », son LOG est inscrit dans un fichier « SERVICE\_YYYYMMDD\_LOG.TXT ».

L'application ne conserve qu'un fichier, elle efface systématiquement celui de la veille.

Elle récupère la liste des Jobs invoqués (par l'application « client » ou par le planificateur) et les exécute les uns à la suite des autres (elle peut en exécuter plusieurs en parallèle, ce paramètre étant géré dans l'application principale, menu de configuration)

## Configuration du planificateur de tâches Windows

Note : Ce paramétrage peut être effectué automatiquement par le menu de configuration de Fuzible. Toutefois, vous pourriez avoir besoin de la modifier/créer manuellement sur des points un peu délicats, tels que le compte d'exécution.

The image displays four screenshots of Windows Task Scheduler configuration windows for the Replicator Service (Ordinateur local).

- Modifier le déclencheur:** Shows the trigger configuration. The task is set to launch at a specific time (18/04/2018, 07:00:00). The frequency is set to "Chaque jour" (Every day) with a repetition interval of 1 day. The task is active.
- Propriétés de Replicator Service (Ordinateur local) - Déclencheurs:** Shows the conditions for task execution. It includes settings for inactivity (10 minutes), power (do not start if on battery), and network (do not start if no network connection).
- Modifier une action:** Shows the action configuration. The action is "Démarrer un programme" (Start a program) with the program path set to "C:\Tools\FuzibleService.exe".
- Propriétés de Replicator Service (Ordinateur local) - Actions:** Shows the action settings. It includes options to allow manual execution, execute on missed start, and stop the task if it runs for more than 4 hours.

A chaque démarrage, elle consulte la pile de Jobs à exécuter et :

- Vérifie que le nombre de Jobs en cours de traitement n'excède pas la valeur configurée
- Trie la liste des Jobs à lancer selon la priorité affectée à celui-ci (entre 1 et 3)
- Démarre l'exécution du ou des Job(s) demandés
- Suit en temps réel l'avancement du job invoqué par la récupération de son « output » et met à jour la table SQL en conséquence afin que l'utilisateur, à partir de l'application client, puisse connaître l'avancement (en cliquant sur « Job Status »)

**Le service doit être installé sur la même machine et le même répertoire que l'application Fuzible.**

## Configuration d'un Job externe (hors Fuzible) avec l'application « Service »

L'idée de cette option permet de tirer profit du système « client/service » pour exécuter n'importe quelle autre tâche.

C'est tout à fait réalisable, et nécessite simplement d'inscrire manuellement dans la table SQL « client\_jobs » (qui est utilisée par l'application de « Service » et qui se situe sur l'instance SQL sur laquelle elle tourne) les informations relatives au Job externe qu'on souhaite pouvoir déclencher (basiquement, l'exécution d'un fichier BAT exécutant certaines opérations est tout indiqué)

CHAMP	DESCRIPTION
User_jobs	Utiliser n'importe quel nom d'utilisateur (par exemple, la personne qui crée ce job)
Job_id	Un numéro de Job, par exemple « 001 »
Job_name	Le nom du Job : c'est ce qui sera affiché à l'utilisateur
Job_description	La description : Quelques mots de plus pour décrire le Job
Job_params	Les paramètres dynamiques par défaut (optionnels)
Job_queries	Ce que vous voulez, peut-être un descriptif des différentes étapes de l'opération ?
Job_haschildren	0
Job_password	Le mot de passe qui permet à l'utilisateur de lancer le Job. Au temps les mots de passe des Jobs Fuzible sont chiffrés, au temps pour ces commandes externes, il vous faut le saisir en « clair » dans la table
Job_priority	La priorité d'exécution (1,2,3)
Application_name	Le nom de l'application à lancer (ex : c:\Tools\Monfichierbat.bat)
Job_category	Catégorisation du Job pour optimiser l'affichage à l'utilisateur

Pour l'utilisateur, voilà la représentation d'un Job externe dans la liste des Jobs qui lui seront proposés dans l'application « Client » :

```
[DATABASE -> FILE]
ReplicatorApp (SVCSCO)
DWH -> FICHIER : Extraction du Grand Livre Comptable
MOSAIC -> BLEUSHARE : Extraction du fichier Acomptes
MOSAIC -> RHPI : Préparation aux modifications de contrats + LOG

[DATABASE -> WEBSERVICE]
ReplicatorApp (SVCSCO)
MOSAIC-PP -> RHPI : Webservice Primes V2
MOSAIC -> RHPI : Webservice Modifier Contrat V2
MOSAIC -> RHPI : Webservices Personnel/Contrat/RIC V2

[EXTERNAL BATCH]
omnis_comete_cloture (CMD)
Clôture Comete

[FILE -> DATABASE]
ReplicatorApp (SVCSCO)
BLEUSHARE -> BLEU : Véhicules : Import Factures Leaseplan (Carburant)
BLEUSHARE -> BLEU : Véhicules : Import Factures Leaseplan
BLEUSHARE -> BLEU : Véhicules : Import Factures Total
RIFLISHARF -> PANAMA : Import RIRPAI RhPI
```

Et voilà la représentation en base de données :

	user_jobs	job_id	job_name	job_description	job_params	job_queries	job_haschildren	job_password	job_priority	application_name	job_category
1	CMD	001	Clôture Comete	Préparation à la cl...	2020001	NULL	0	wolf	1	c:\Tools\omnis_comete_cloture.bat	External Batch
2	SVCSCO	[14]	BLEUSHARE -> BLEU : Véhicules : Imp...	Traitement des fic...	150320	IMPORT_LEASEPLAN_C...	0	3sdiqA8lNKu0...	1	ReplicatorApp.exe	File -> Database
3	SVCSCO	[15]	BLEUSHARE -> BLEU : Véhicules : Imp...	Traitement des fic...	2020004	IMPORT_LEASEPLAN:s...	0	y/95rn5oe2Dy...	1	ReplicatorApp.exe	File -> Database
4	SVCSCO	[16]	BLEUSHARE -> BLEU : Véhicules : Imp...	Traitement des fic...	20200331	IMPORT_TOTAL_SERIS...	0	r+CKpDwXSFm...	1	ReplicatorApp.exe	File -> Database
5	SVCSCO	[53]	DWH -> FICHIER : Extraction du Grand ...	Extraction du Gra...	>=%YYYY%MM<...	v_grand_livre_comptable...	0	lyMQTIDIK0+u...	1	ReplicatorApp.exe	Database -> File
6	SVCSCO	[62]	MOSAIC -> BLEUSHARE : Extraction d...	Réalise l'extractio...	20200515	INT_acomptes.CSV:selec...	0	57P6RHhg8K...	1	ReplicatorApp.exe	Database -> File
7	SVCSCO	[63]	BLEUSHARE -> PANAMA : Import RUB...	Import de l'OD de ...	120520_130125...	RUBPAI_RHPI_import.sel...	0	PAEzQGzkzVD...	1	ReplicatorApp.exe	File -> Database
8	SVCSCO	[65]	BLEUSHARE -> PANAMA : Import des ...	Import des heures ...	2020Avril_202005...	COMETE_heures:select *	0	4w/nb5iyvudz...	1	ReplicatorApp.exe	File -> Database
9	SVCSCO	[72]	MOSAIC -> RHPI : Webservice Modifier...	Exécution des We...	id_maticule	webService=dossier.contr...	1	S5M08Gtp0/G...	1	ReplicatorApp.exe	Database -> Webservice
10	SVCSCO	[75]	MOSAIC -> RHPI : Webservices Person...	Exécution des We...	modifier.ric.id_matr...	webService=dossier.(?)...	0	HNq7kAZlyKn...	1	ReplicatorApp.exe	Database -> Webservice
11	SVCSCO	[81]	MOSAIC -> RHPI : Préparation aux modi...	Extractions à partir...		01_CONTRAT_MODIFIE...	0	07JbyuFDrbq1...	1	ReplicatorApp.exe	Database -> File
12	SVCSCO	[85]	MOSAIC-PP -> RHPI : Webservice Prim...	Exécution du We...		program=SPS010RB:SEL...	0	pms5BF4cxE6...	1	ReplicatorApp.exe	Database -> Webservice

## Application « Client »

Un client léger existe et permet d'exécuter les jobs à distance à partir d'un poste client.

L'intérêt de mettre à disposition d'un utilisateur « novice » la possibilité de déclencher un Job est crucial, par exemple :

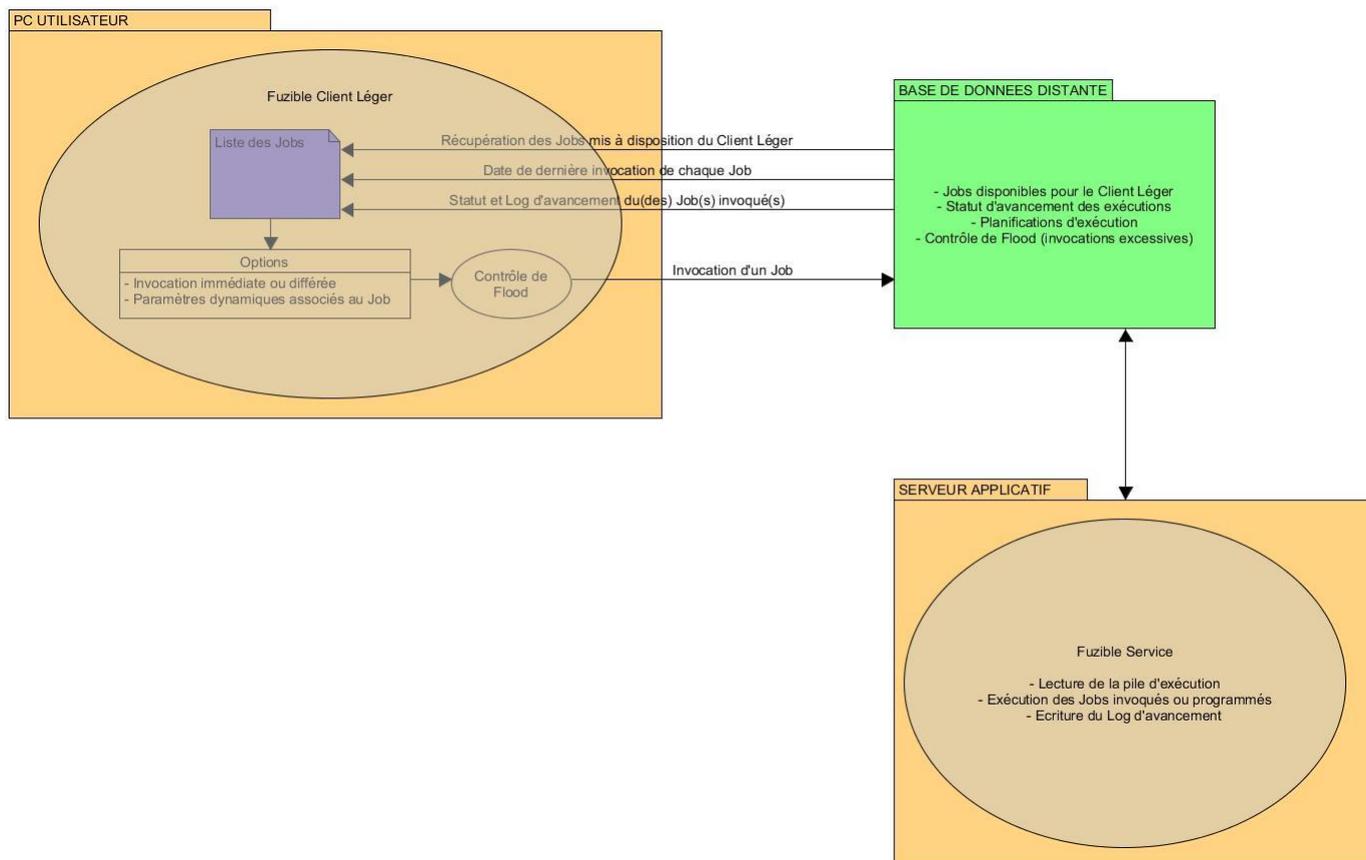
- Déclencher des actions précises à des dates non planifiables
- Produire des reportings à la volée

### Pré-requis sur poste client :

- Windows OS (7+)
- .NET Core 3.1
- Fichier INI « CLIENTAPP.INI » (disponible dans répertoire d'installation de Fuzible)

**Attention : Si vous changez la chaîne de connexion du couple Client/Service dans les paramètres généraux de l'application, vous devrez fournir le nouveau « CLIENTAPP.INI » aux utilisateurs car ce fichier contient les paramètres de connexion !**

Voici un diagramme de fonctionnement de l'application Client :



Elle dialogue avec le Service Fuzible à travers une base de données que vous paramétrez dans le client lourd.

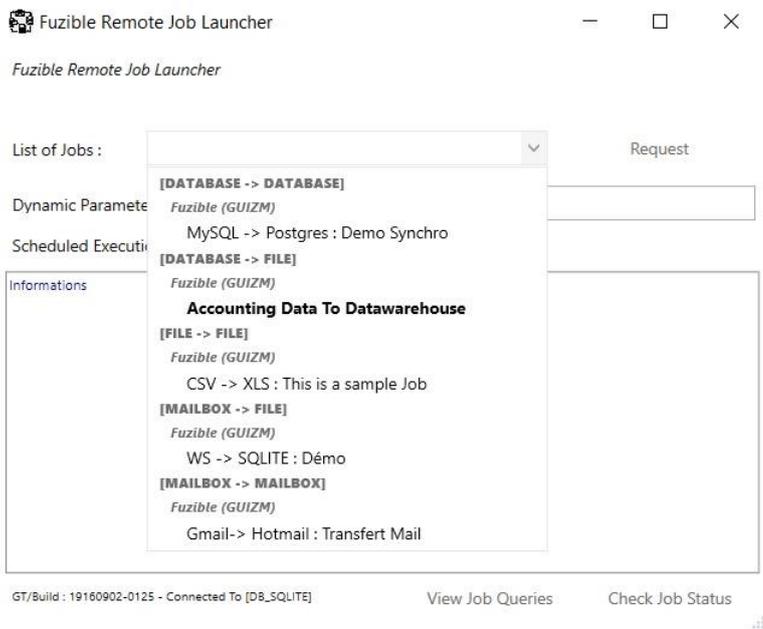
Le client léger utilise un système de "pile". En choisissant un job à exécuter, l'application écrit une ligne SQL dans l'instance de l'application « Service ». Les paramètres de cette connexion sont présents dans « CLIENTAPP.INI » et sont chiffrés pour des raisons de sécurité.

Le programme client n'exécute donc pas le Job. Il le charge dans une pile ; c'est l'application « Service » fournie qui se charge de dépiler, contrôler et lancer les jobs invoqués.

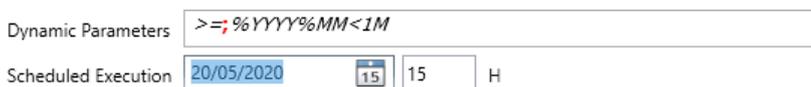
Un Job doit être au préalable configuré comme étant utilisable dans l'application « Client » :

En ouvrant l'application « Client », vous pouvez choisir un des Jobs disponibles dans la liste. Son mot de passe sera requis ; la personne qui crée le Job aura eu la présence d'esprit de le fournir à la (ou aux) personne(s) habilitée(s) à lancer le Job.

La liste des Jobs disponibles :



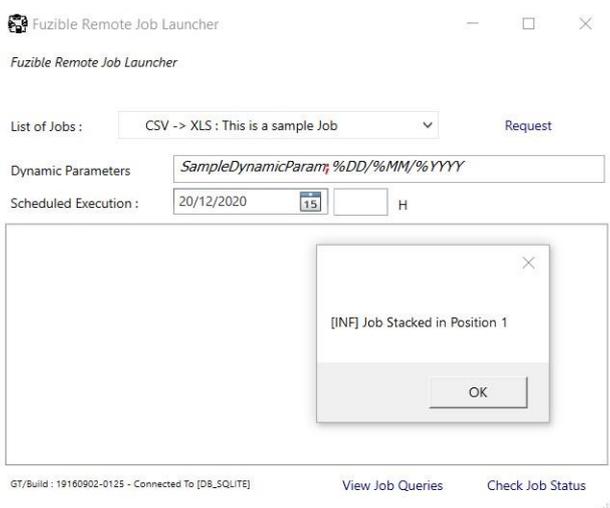
La personne qui manipule l'application client peut intervenir sur les paramètres dynamiques du Job sélectionné : très utile pour par exemple produire un Reporting spécifique à une période, définir un nom de fichier à importer, etc... tout est envisageable, tout dépend comment le Job est configuré !



Il lui est aussi possible de différer le lancement du Job, par défaut, en cliquant sur « Request », le Job sera dépilé le plus vite possible par l'application « Service », mais on peut aussi programmer le déclenchement à une date/heure particulière (par exemple pour lancer des traitements « lourds » en pleine nuit sans devoir se réveiller en pleine nuit pour déclencher le lancement)

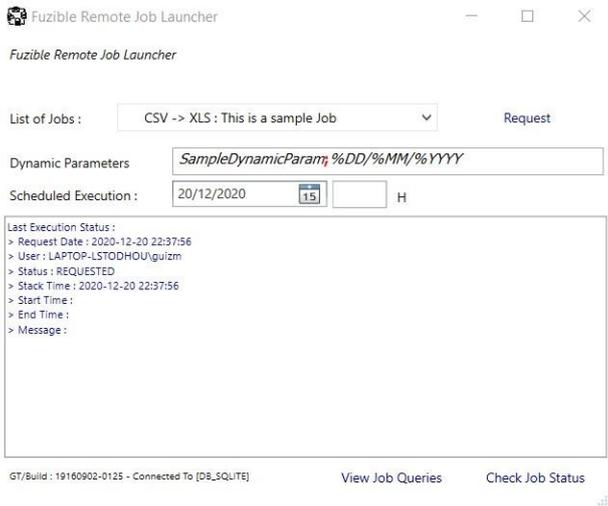
« View Queries » permet de voir les requêtes associées au Job, et leur traduction avec les paramètres dynamiques. Plutôt réservé à des utilisateurs ayant quelques notions de SQL !

Après avoir cliqué sur « Request », le programme indique la position dans la pile :

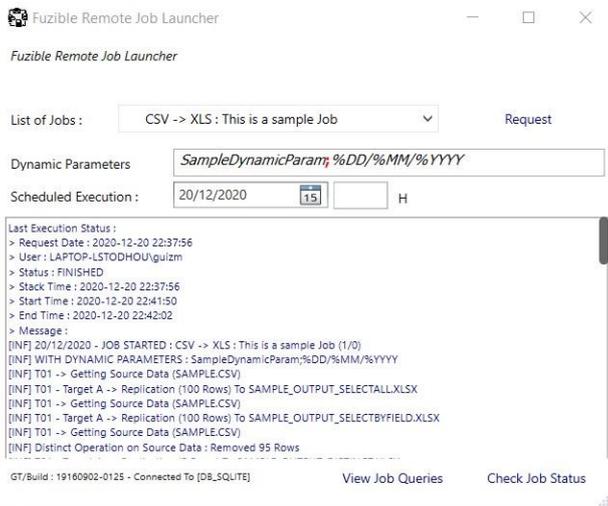


L'utilisateur peut suivre le déroulé en cliquant sur « Check Job Status ».

Le Job a été invoqué mais pas encore pris en charge par l'application « Service » :



## Job exécuté et terminé :



**Note :** Si l'utilisateur clique sur « Check Job Status » avant d'avoir demandé l'exécution, il verra le LOG de la dernière exécution en date, si toutefois celui =-ci n'a pas encore été purgé (le temps de conservation se définit dans les paramètres généraux du programme)

# Fuzible SQL : Lexique

## Fonctions supportées

Voici une liste des fonctions SQL supportées par le moteur de Fuzible pour le requêtage d'une source non-SQL. Pour l'utilisation, internet est votre ami (standard SQL). L'assistant de requête vous présentera également la manière de les utiliser.

TRANSFORMATION	
SUBSTRING	Extraire un morceau de chaîne dans une chaîne
CONCAT	Concaténer des champs ou des caractères
CASE field WHEN ... THEN ... ELSE ... END	Piloter une valeur en fonction d'une autre
CONVERT	Force la conversion d'un type de données pour un autre
LTRIM, RTRIM	Effacement des espaces avant/après une chaîne
ISNULL, COALESCE	Remplacer une valeur vide par autre chose
LPAD, RPAD	Compléter une valeur par une chaîne à droite ou gauche
LENGTH	Longueur d'une chaîne
CHARINDEX	Position d'une chaîne dans une chaîne
LOWER, UPPER	Mettre une valeur en majuscules ou minuscules
REPLACE	Remplacer une valeur par une autre
ANONYMIZE	Mélange le contenu d'une valeur pour « anonymiser » la donnée
AGGREGATION	
SUM	Somme d'un ensemble
MAX, MIN	Maximum ou minimum d'un ensemble
AVG	Moyenne d'un ensemble
COUNT	Quantité d'un ensemble
FONCTIONS ESSENTIELLES	
SELECT DISTINCT	Supprimer les doublons d'un résultat
SELECT TABLE x	Propre au SQL de Replicator : permet de définir la table sur laquelle on aliase les champs (appel : un webservice peut par exemple renvoyer plusieurs tables)
SELECT TABLE x ONLY	Propre au SQL de Replicator : permet de ne renvoyer qu'une table choisie dans un ensemble (cas des webservices par exemple)
LIMIT, TOP	Limite les résultats retournés Ex : SELECT TOP 100 * FROM monfichier Ex 2 : SELECT * FROM monfichier.csv LIMIT 100
JOIN (LEFT, OUTER, INNER, RIGHT)	Jointures entre sources
WHERE	Filtres (= < > != IN NOT IN) ainsi que les « SELECT » imbriqués Ex : WHERE LENGTH(li_test) > 0 Ex 2 : WHERE li_test in (SELECT li_data FROM matable)
ORDER BY	Organisation des résultats
GROUP BY	Regroupements d'aggrégations
UNION	Merge de plusieurs résultats aux schémas identiques
FONCTIONS AVANCEES	
Manipulations mathématiques sur fonctions	Additions et soustractions aux fonctions traitant de nombres entiers (charindex, length, substring) Ex : SELECT SUBSTRING(li_test, CHARINDEX(li_test, '-') + 1, 10) FROM monfichier.csv
Sous-requêtes	Ex : SELECT * FROM (select * FROM monfichier.csv) as subQ Ex 2 : SELECT * FROM monfichier.csv WHERE id_test NOT IN (SELECT id FROM monautrefichier.csv)

## Fonctions courantes non supportées

HAVING	Filtrer des fonctions d'aggrégation
« null »	Le « NULL » au sens d'une base de données n'est pas compris Ex : CASE WHEN x IS NULL THEN doit être saisi ainsi : CASE WHEN x = '' THEN
« GETDATE » ou « CURRENT_TIMESTAMP »	De manière plus générale, la saisie dynamique du datetime actuel. En revanche vous pouvez utiliser les paramètres dynamiques du Job pour contourner cette limitation Ex : SELECT * FROM monfichier WHERE annee > {%YYYY}
Sous-requête en tant que champ requêté	Ex : SELECT (select id FROM monfichier.csv) as id FROM monautrefichier.csv
Manipulations mathématiques sur agrégats	Ex : SELECT COUNT(*) + 10 FROM monfichier