# **Fuzible - General Documentation**

Author	Guillaume Tristant
CREATION	20171002
REVIEW	20221227
Resources	www.fuzible-app.com
SUPPORT MAIL	guillaume@fuzible-app.com
Forum	www.fuzible-app.com/wordpress/fuzibleForum/phpBB3/
Language	FR / EN

## **Table of Content**

Preamble	5
Information about documentation	5
Principle	6
Installation	6
Recommended configuration	6
Language	6
Enterprise installation	6
Installation for an Individual	6
Software Architecture	7
Software Registration	8
License Perimeter	9
License Upgrade	9
Program compatibility	
Supported Scenarios	
Security	
How it works, start-up	
Start the program in "UI" mode	
Start the program in "CONSOLE" mode (silent execution of a Job)	
First launch	
Working Path	
Integrated software support	
Demonstration jobs	
Quick Help Button	
Tutorials	
Query Assistant	
Online help	
Software Settings	
Connections	
Database	

Mongodb	
Files	
Webservice REST	
Mail	
Active Directory	
Log	
SHS Analyzer	
SQL	
File	
Mail	
Webservices	
Service/Client App	
Dev	
Tools	
Export Job (XML)	
Import Job (XML)	
Reorganize Jobs	
Move as Sub -Job	
Move as Main Job	
Move Down/Up	
Import External Job Parameters	
Load another Userspace (read-only)	
Job Creation	
Job Configuration tab	
Job Type	
Language Script	
Job Summary	
Orchestration	
Planning Calendar	
Source tab	
BDD	
MONGODB	
CSV file	
Excel file	46
XML and JSON Files	46
Webservice REST	
Mailbox	47
Active Directory	

Data Analyzer	47
Data Transformation	
Pre/Post-Job Commands	52
Looped Pre-Job Commands	53
Target tab	55
Common settings to all targets:	55
Database	56
CSV file	57
Excel file	57
File XML	57
File JSON	58
Webservice REST/NUXEO	61
Mailbox	62
Active Directory	62
Queries tab	63
Output	64
How field mapping works	64
Special cases of synchronization queries:	65
SELECT - From a database, to a database	67
SELECT - From a database, to a file	67
SELECT - From a database, to an email address	67
SELECT - From a file	68
SELECT - From a webservice using Fuzible SQL (A)	68
SELECT - From a webservice using Fuzible SQL (B)	69
SELECT – From Salesforce API using SoSQL	69
SELECT - From an e-mail box	70
SELECT - From Active Directory	70
SELECT Multi-tables, multi-files	73
Multi-Target Queries	73
Cross-Queries	75
Contextual Query Menu	83
QUERY ANALYZER - News Source	
QUERY ANALYZER - Target Info	
QUERY ANALYZER - View Data	86
QUERY ANALYZER - Query Details	87
EXECUTE QUERY - Run this individual Query	
SYNCHRO-QUERY - Transcoded for Target	
SYNCHRO-QUERY - Validity check for Synchro Query	

SCRIPTING - Get Full Header Query and copy/paste it	89
SCRIPTING - Transformations	
SCRIPTING - Add a Dynamic Parameter	
SCRIPTING - Basic Query Builder	92
ADVANCED QUERY SCRIPTING - Add Cross-Connections Join	92
ADVANCED QUERY SCRIPTING - Create Dual Target	92
Log Viewer tab	
Running a Job	
"Service" Application	
Setting up the Windows Task Manager	
Setting up an external job (excluding Fuzible) with the "Service" app	
"Client" Application	
Fuzible SQL: Glossary	
Supported functions	
Unsupported	

## Preamble

First, I would like to thank you warmly for using (or trying) Fuzible. This software is the culmination of several years of work, and the observation that many tasks related to data manipulation are too often time-consuming and redundant.

There are many alternatives, and the idea is not to create competition, but another way of designing data exchange and manipulation.

The program is aimed at developers with minimal knowledge of the SQL language.

#### The philosophy of the program is simple:

Any Data Source is a database and can be queried as such!

#### Some use cases:

- Data replication (copy)
- Data synchronization (smart comparison of 2 sources)
- Interfaces (ex: retrieve data from one software's webservice in SaaS and send it to another software in the form of XML files)
- Migration of data from one BDD to another (regardless of the driver)
- Fast data extraction (ex: SQL to CSV)
- Fast data importation (ex: EXCEL to BDD)
- Data comparison (ex: to control the integrity of 2 BDDs)
- Cross-join from different sources in real time
- Filling a data warehouse (ex:, integrating a file into a BDD is fully automated: from data analysis to table creation)
- Sync a pre-production environment with a production environment

## Information about documentation

The documentation was written in French and translated in English using an automatic translator. I checked a few key things but I am sorry if there are still weird phrases!

## Principle

Fuzible is a tool that allows you to import, export, mix, synchronize, replicate, compare data. The general principle is based on the definition of a Data Source (which will be queried), and a Target (in which data will be copied).

The tool can work with several SGBDs (see compatibility table), both for export and for data import. It offers an opportunity to analyze all the data needed to create intelligent import fields (consistent types and lengths). But beyond that, he can read and write in files, webservices, mailboxes, and in the Active Directory!

Finally, it is designed as modular, from data processing to LOG management and Jobs orchestration. A Swiss knife, in short. It can meet the most basic needs (copying a CSV file in a database) to the most complex (synchronizing 2 database environments).

## Installation

The program can be installed anywhere on the hard drive. You can choose to use the portable version or the installer version. In both cases, no data will be written anywhere other than in the installation directory.

## **Recommended configuration**

This assessment is based on my tests. Note that multithreading exponentially increases the needs of the machine running the program. Setup indicated for 4 threads.

- Microsoft Windows OS (from Windows 7, or Server 2008r2 and more)
- 4th generation Intel I5 CPU with at least 2 execution threads
- 8GB of RAM (large minimum)
- HDD 7200tr/min with10GBavailable.
- Internet access
- Microsoft NetCore Framework v3.1 (<u>https://dotnet.microsoft.com/download</u>)

## Language

The program is available in 2 languages (French, English). By default, it is set to use the system language, but you can switch from one language to another in the "File" menu.

## **Enterprise installation**

In a network environment, Fuzible is multi-user (and delivers all its possibilities when used like this), it is advisable to install it on a server accessible in RDP. In addition, a server will often have much more access rights to SQL instances, network paths, FTP... than a local computer.

Similarly, if you replicate from one BDD to another, and Fuzible is installed on a local computer, processing times could be horribly slow because the data will have to be retrieved first from the Source server to the local computer and then from it to the Target server. If you are behind a VPN, or if your connection to the network is slow, performance will be extremely degraded.

## **Installation for an Individual**

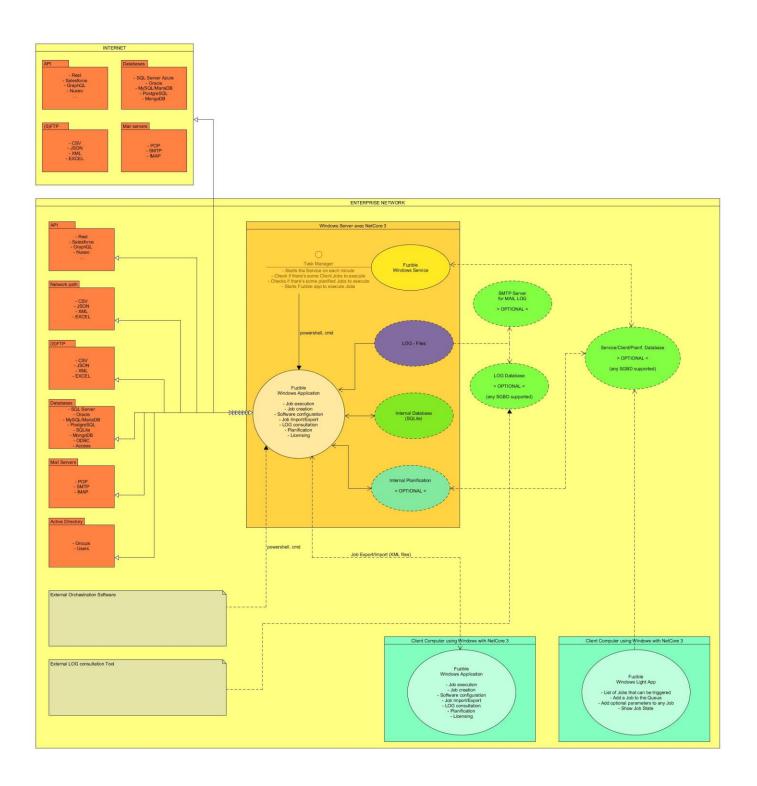
Just make sure that the computer on which you install the app accesses all the data sources you will handle (SQL instances, local network, Active Directory domain, Internet connection...) and that the user account is "administrator" of the computer.

## **Software Architecture**

The diagram shows the different bricks of the application. Some of them are optional and do not need to be configured other than the application itself.

The Core application is in the Orange rectangle.

All dotted lines represent optional bricks. Green ones are included with Fuzible, and gray ones are external.



## **Software Registration**

When starting the software, whether it was purchased or not, it runs in demonstration mode. It is limited in its use and a message will inform you at each start.

	×
Unregistered Version. Check 'Help/How to Register' to get informations about Registration.	
ОК	

You will not be able to do the following:

- Orchestrating Jobs
- Importing and exporting Jobs
- Use the "Client" and "Service" app
- Create more than 3 Jobs

#### To Register Fuzible, just go to the Help menu:

🚰 SHS Fuzible Data Replica	tor, Synchronizer	=
File Configuration Tools	Help	
Job Selection	About	
GUIZM	Browse the Website Documentation	
Job Configuration Sour	@ Contact us How to Register	Log Viewer
Job Description	Enter License Key Check for Updates	

#### A menu opens and asks you to enter your email address to register.

Registration System		-		×
The provided data	n System. ired fields and click 'Request Registration of will be sent to Fuzible server and be store Registration Code.		sed	
Email address				
Buy date	19/01/2021			
Transaction number	Free Version			
License type	A0 : 1 machine, 5 jobs, minor updat	es (FREE)	$\sim$	
	Request re	egistration co	ode	

If you purchased the program, you also need to choose the type of License you purchased from the drop-down list, and then specify the transaction number that was provided to you at the time of purchase, as well as the date the License was purchased.

Note: Your information is only stored on Fuzible's server, the data is not transmitted under any circumstances to anyone. Registering the free version allows us to measure the number of active users on the app.

The data sent is: email address, CPU model, amount of RAM, Windows version, version of the NetCore Framework, unique PC identifiers (main hard drive, CPU), amount of Jobs in Fuzible, amount of connections, local IP address, public IP address, program installation path, program version.

Click "Request Registration Code" to send your data to the Fuzible Server. Your registration request will be processed manually within a few minutes, and a registration code will then be sent back to you by email.

This should be entered in the Help menu:

😭 SHS Fuzible Data Replicat	tor, Synchronizer
File Configuration Tools           Job Selection           GUIZM           Job Configuration           Source           Job Description	Help About Browse the Website Documentation @ Contact us How to Register Enter License Key Check for Updates
License Number :	×
License Number : MyLicenseNumber	OK

The code authenticity is checked locally, and a message will inform you of the validity of the code.

The free version gives you all the features of the app, but the number of Jobs that can be created is limited to 5. Paid versions are built around the number of Jobs that can be created, as well as the type of updates that will be possible to download in the future (corrective updates, minor, major developments).

#### **License Perimeter**

A Fuzible License is multi-user (several users on the same computer can use the app independently) but is limited to the computer on which the application is installed. The following changes will cause the program to return to demo mode:

- Changing the installation directory
- Change of CPU
- Change of hard drive
- Manual changes to the program's internal database

If a change in circumstances proves to be legitimate, you can request assistance on the website:

## www.fuzible-app.com

#### **License Upgrade**

If you already own a License, and you just acquired a better one from our website, you can go to this menu to update the software. The Registration process will be the same as the first time you did register the app.

H	lelp	
	Tutorials	)
	About	
	Browse the Website	
	Download DB Browser for SQLite	
	Documentation	
	@ Contact us	
	How to Register	
	Enter License Key	
	Upgrade License	

## Program compatibility

Dbms	Support
SQL Server	SOURCE: Full support
	CIBLE: Total support
Mysql	SOURCE: Full support
	TARGET: Total support
Postgres SQL	SOURCE: Full support
	TARGET: Total support
Odbc	SOURCE: Full support
	TARGET: Partial support: Need you to set up some system queries
Oracle	SOURCE: Full support
	TARGET: Total support
Sqlite	SOURCE: Full support
	TARGET: Total support
Access	SOURCE: Full support
	TARGET: Total support
Mongodb	SOURCE: Full support
	TARGET: Total support
File	SOURCE: Support XML, JSON, CSV, XLS, XLSX (FTP)
	TARGET: Support XML, JSON, CSV, XLS, XLSX (FTP)
Webservices	SOURCE: API REST
	TARGET: Partial support (REST, http)
Mailbox	SOURCE: POP, IMAP
	TARGET: SMTP
ACTIVE DIRECTORY	SOURCE: users, groups
	TARGET: users, groups

## **Supported Scenarios**

Source	Target	REPLICATION	SYNCHRONIZATION
Database	Database	Yes	Yes
Database	File	Yes	Yes
Database	Webservice	Yes	No
Database	Mailbox	Yes	No
Database	Active Directory	Yes	Yes
File	Database	Yes	Yes
File	File	Yes	Yes
File	Webservice	Yes	No
File	Mailbox	Yes	No
File	Active Directory	Yes	Yes
Webservice	Database	Yes	Yes
Webservice	File	Yes	Yes
Webservice	Webservice	Yes	No
Webservice	Mailbox	Yes	No
Webservice	Active Directory	Yes	Yes
Mailbox	Database	Yes	Yes
Mailbox	File	Yes	Yes
Mailbox	Webservice	Yes	No
Mailbox	Mailbox	Yes	No
Mailbox	Active Directory	Yes	Yes
Active Directory	Database	Yes	Yes
Active Directory	File	Yes	Yes
Active Directory	Webservice	Yes	No
Active Directory	Mailbox	Yes	No
Active Directory	Active Directory	Yes	Yes

## **Security**

The program uses an SQLite database to work. Several information is encrypted (AES), including login chains and passwords, to protect data privacy for each user session.

## How it works, start-up

### Start the program in "UI" mode

Run Fuzible.exe

## Start the program in "CONSOLE" mode (silent execution of a Job)

#### Run Fuzible.exe with arguments

- 1 Userspace (basically, the user who is connected)
- 2- Job ID to run at start-up (if needed)
- 3 Password(encrypted)
- 4 Dynamic Parameters (see "Script Language" section)

#### Example:

#### Fuzible .exe "GUILLAUME" "[10]" Apza-7824

> Will launch the no.10 job of the user "GUILLAUME"

Given the austerity of entering arguments, the program proposes, in its UI, to display you the "launcher" code of each Job so that you can copy and paste it directly (in a scheduling tool for example).

#### In summary, the program has two operating modes:

- In silent mode (name of job to be performed), it does not load the graphical interface and performs the Job in the background.
- In UI mode, it opens the graphical interface: it allows you to edit, add and launch Jobs

## **First launch**

At the first launch of the program, it will create:

- Your default general configuration
- Two Demo Jobs containing about 20 test queries, to introduce you to the features.
- 3 local "file" connections, 1 SQLite connection, 1 connection to a demo webservice

You will also be notified that you are not registered, and because of that, you will not be able to create more than 3 Jobs. In addition, some features will be disabled.

#### **Working Path**

Fuzible stores its data into the « public » directory : C:\Users\Public\Documents\Fuzible.

This is where you get the internal Fuzible database, LOG files, the Client App binaries, and also demo files that are used by the default Job File connections.

## **Integrated software support**

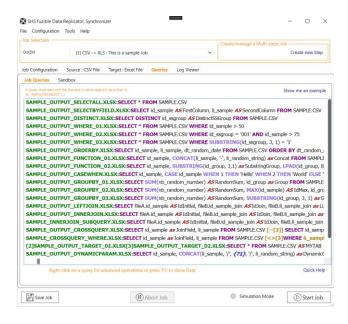
Like any program, the first use can sometimes be off-putting. Fuzible is no exception to the rule, but before closing the program permanently, let me show you how you can get some assistance.

#### **Demonstration jobs**

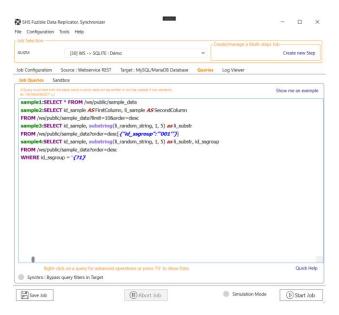
During the first launch, Fuzible will create 2 Demo Jobs for you that will help you understand how to query a Source other than a database using the SQL language. Those are accessed through the selection menu:

Job Selection		
GUIZM	1	~
	[File -> File]	
Job Configuration	[1] CSV -> XLS : This is a sample Job	
	[Webservice -> Database]	
Job Description	[2] WEBSERVICE -> SQLITE : This is a sample Job	

- A Job that copies data from CSV files into EXCEL (XLSX) files:



- A Job that retrieves data from the demonstration webservice and integrates it into Fuzible's local database:



### **Quick Help Button**

Some features do have a small "?" button to give you some quick explanations without having to look at the full documentation:



## **Tutorials**

Three tutorials have been programmed to make it easier for you to accommodate with the software. You can find them here:

lelp		
Tutorials	•	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	ľ	
Documentation		
@ Contact us		
How to Register		
Enter License Key		

A script then triggers, and you will be fully guided.

- "Escape" key: Leaves Tutorial mode
- "F1" key: Move forward at the next step.
- "F2" key: Go back to the previous step.

ile Configuration To	ols Help				
Job Selection			Cre	ate/manage a Multi-steps	Job ————
GUIZM			~		Create new Step
Job Configuration	Source Target Queries Log V	liewer			
Job Description					
		Renam	ne Job	Change Pacoword	
		Delet	. 100	Panification	Create New Job
Main Parameters	Select 'Data Replication' for a sir	mple data copy Job :			
Job Type	Data Replication 🗸				
sou type	a beget without any comparison				
Dynamic Parameters					
Dynamic, Parameters No. can write any feet or an No. can use them anywhere	and the connect fact parameter must be	equivalent lagram metriculare (; ) a deser des des 175, 175,		View July 7	10th Replaced Values
Ophanic Heameters You can write any field or an You can use them anywhen	available command. Such parameter must be o Sparse, test fields, connections) by observer	egended by a servicelar (, ) g theor the theory (1), (1),		View July V	
Dynamic, Parameters Yes can write any test in an You can use them anywhen	mailable command. Each parameter must be (games, test fields, connectional by observing	agantal (g. consistint)) g than the the (75, 75)-		View Jub 1	105 Replaced Values
Operating Property of the set	exhibite connected Seth promotion must be games, our fields, connections by indexect	egenthe Page a constant ( ) ig there like the ( PG, PG,		Vera Jub 1	
No car with any first or an Two car we have anywhen		g han bir bir (1, 17, 17).		View Jub 1	
Versen etter og versen versen og versen versen og versen versen og	Terrar + Informations	() Log in SQ.			
No car with any first or an Two car we have anywhen	Terrar + Informations	g han bir bir (1, 17, 17).	]		
Versen etter og versen versen og versen versen og versen versen og	Terrar + Informations	g han bir bir (1, 17, 17).	]		
Na an with any hird or an Na an an Alan Arganian LOG Level Send Mail When Friedh	Trong + Informations v	in the first of the second sec			
Versen etter og versen versen og versen versen og versen versen og	Trong + Informations v	g han bir bir (1, 17, 17).			
Na an with any hird or an Na an an Alan Arganian LOG Level Send Mail When Friedh	Trong + Informations v	in the first of the second sec			
No. on with any hid or an No. on with any hid or year 100 Lowell Send Matt When Freidh Marc. Wable in Cleast A	Torum + Informations v	in the first of the second sec			
No. on with any hid or an No. on with any hid or year 100 Lowell Send Matt When Freidh Marc. Wable in Cleast A	Torum + Informations v	in the first of the second sec			
No. on with any hid or an No. on with any hid or year 100 Lowell Send Matt When Freidh Marc. Wable in Cleast A	Imon + informations ~ ~	in the first of the second sec			

## **Query Assistant**

When you are creating your Job, you'll be writing queries, 2 options are available to help you understand Fuzible's philosophy.

- "Show Me an example" will show you a "generic" query adapted to the scenario you set.
- "Basic Query Builder" will accompany you in creating a simple query.

SHS Fuzible Data Replicator, Synchronizer		- 🗆 X
ile Configuration Tools Help		
Job Selection	Create/manage a Mu	Iti-steps Job
GUIZM	~	Create new Step
Job Configuration Source : SQLite Database Target : CSV File	Queries Log Viewer	
Job Queries Sandbox		
A Query must start with the filename in which data will be written in. ex : MyFile.CSV:SELECT ()		Show me an exemple
ReplaceBy_AnyFile.CSV:SELECT * FROM user_parameter	Chemin local -> ReplaceBy_AnyFile Query Analyzer	.CSV
	> 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	
	> Basic Query Builder	
	Advanced Query Scripting	I
	> Add Cross-Connections Join	
	<ul> <li>&gt; Create Dual Target</li> <li>&gt; Multiple files Output Patterns</li> </ul>	
	> multiple liles Output Patterns	F
0		
Right-click on a query for advanced operations or press	'F5' to show Data	Quick Help
Save Job Abort Jo	b Simulation	Mode Start Job
ADDIT JO		Start JOD

However, once you understand the logic of the program, these 2 options will be of no use to you, they are simply here to help you understand how Fuzible works, but I'm sure that Tutorials and Demonstration Jobs will be much more useful!

## **Online help**

If you need additional help, feel free to go to the forum (link at the top of this documentation), contact me, or download Demo Job.

Indeed, Fuzible allows you to create complex scenarios. Its highly modular design allows it to meet many needs, but it can be helpful to ask for help if you are not sure the best way to achieve your goals.

## **Software Settings**

## Connections

Go to the Configuration / Connections menu to set up the connection strings.

€n] S	HS Fuzible Data Replicate	or, Synchronizer
File	Configuration Tools	Help
Job	Connections	
GUI	Log	
001	SQL	
Job	File	Target Queries Log Viewer
100	Webservices	larger Queries Log viewer
Jo	Mail	
	Data Analyzer	
	Service/Client App	

#### A menu then opens:

nnections	Log - Errors	SHS Analyzer	SQL	File	Mail	Webservices	Service/Client App	Dev.
	list —			~		Create New Con Try Connection		Save Connection Delete Connection
Connection	name		~					Connection String Assistant
dvanced Pa	irameters							

You can either choose to change or delete an existing connection.

Beware of impacts, the red connections in the list are used in at least 1 job! You can see this impact by flying over the name connection with the mouse:

	Connection List	~	Creat
	[11] -> MySQL Raspberry		í
	Used in one or more Jobs : JOB [13] : [Queries] JOB [20] : [Conn. String Target][Queries] JOB [30] : [Cueries] JOB [32] : [Queries] JOB [35] : [Conn. String Source] JOB [36] : [Conn. String Source] JOB [41] : [Conn. String Source] IMB [41] : [Conn. String Source] Source] Source] Source Source Source] Source Source Source] Source Source Source Source] Source Source	l Driver (*.xls, *.xlsx, *. cess Driver (*.mdb, *.a	
	[Sqlite Database]		
L	[5] -> Local SQLite File		-
٢	[17] -> Copy of Local SQLite File		-
	[MS Access Database]		
	[18] -> Test Access OLEDB Driver		
	[MongoDB Database]		

Each connection is identified by an ID (ex: [10]) which allows it to be used in queries (see chapter related to queries)

Connect	ion List ——			Create Nev	w Connectio	n		Save Conn	ection
[11] -	> MySQL Rasp	berry	~	Try Con				Delete Con	
Connect	ion Parameters	;							
Connect	tion name	MySQL Raspber	ry				Con	nection Strir	ng Assistant
Driver	MySQL/Mar	iaDB 🗸	Scan Network	for instances	Dates	dd/MM/yyyy	$\sim$	Decimals	, (comma) 🗸

- The common settings allow you to create a new connection:
   1/ From an existing connection if already selected.
   2/ From the data entered by the user.
- You can also test the current connection to check if it is working.
- The connection string assistant will accompany you during the creation of the new connection.
- You can also change dates & decimals localization connection properties: you may need for example to extract US data (MM/dd/YYYY dates, dotted decimals) and import them into a European database (dd/MM/yyyy, comma decimals)
   It's not changing your OS locales you just change the way data is handled by Fuzible for a specific connection.
   Note : by default, the localization settings are set according to your OS localization.
- Finally, in the case of a database, an option allows you to scan the local network to find possible available instances (the tool scans the network for open ports: those are different depending on the SGBD you choose).

#### Database

nnections Log - E	rrors SHS Analyzer	SQL	File	Mail	Webservices	Service/Client	t App Dev.	
Connection List [11] -> MySQL Ras	pberry		~		Create New Con Try Connection		Save Conr Delete Con	
Connection Parameter							1	
Connection name Driver	MySQL Raspberry MySQL/MariaDB	~		Scan	Network for insta	ances	Connection Stri	ng Assistant
mysqu/manabb connectio	i string (ex : server=myserverAdd	iress;Databa	ase=myDat	aBase;Uid:	=myUsername;Pwd=my	Password;) :		
	0.105; <i>Uid=</i> guizmox; <i>P</i>						<b>/Mode=</b> none;	
Server=192.168.	0.105; <i>Uid =</i> guizmox; <del>P</del>	wd=my	Passwo	ord; <b>Por</b>	t=3306; <b>DATA</b>		/ <i>Mode=</i> none; Load default	narams

## A complete listing of connection strings can easily be found online: <u>https://www.connectionstrings.com/</u>

#### However, here is a basic example for the different SGBDs

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

ODBC Driver Case: Fuzible uses specific queries and settings for each type of database for its proper operation. By definition, those are unique to each SGBD and you may want to enjoy all the benefits of Fuzible with an unsupported native driver.

The "Connection Params" zone is there to meet this need. It sets up each of Fuzible's internal queries to make it compatible with any SGBD.

Parameter	Value
P_ESCAPE_CHAR	Escape character to delineate table and field names (often quotation marks)
P_DATE_FORMAT	Native date format (DATE, TIMESTAMP, DATETIME)
P_STRING_FORMAT	Native character format (VARCHAR, VARCHAR2, NVARCHAR)
P_CREATE_PRIMARY_KEY	SQL code for creating a primary key in a table
P_CREATE_TABLE	SQL code to create a table
P_CHANGE_COLUMN_ALLOW_NULL	SQL code to allow NULL values in a specific column
P_CHANGE_COLUMN_DISALLOW_NULL	SQL code to ban NULL values in a specific column
P_CREATE_COLUMN	SQL code to create a column in a table
P_CHANGE_COLUMN_TYPE	SQL code to alter the size of a column in a table
P_CHANGE_COLUMN_DEFAULT_VALUE	SQL code to change or put a default value in a column in a table
P_CHANGE_COLUMN_ADD_UNIQUE	SQL code to add a single key to a column in a table
P_SHRINK_TABLE	SQL code to clean a table
P_GET_COLUMNS_LIST	SQL code to retrieve the list of columns of a table (expected: name, type, max size, nullable, identity, type, numerical accuracy, default, single, key)
P_GET_COLUMN	SQL code to retrieve information from a column of a table (expected: name, type, max size, nullable,identity, type, numerical accuracy, default, single, key)
P_DISABLE_TABLE_CONSTRAINTS	SQL code to disable table constraints (for example, allow insertion without key control)
P_ENABLE_TABLE_CONSTRAINTS	SQL code to activate table constraints (key control)
P_GET_TABLES	SQL code to retrieve a BDD's table list
P_GET_TABLES_FILTERED	SQL code to retrieve the filtered list of tables of a BDD
P_GET_PRIMARY_KEY	SQL code to retrieve the primary key from a table
P_GET_FOREIGN_KEYS	SQL code to retrieve foreign keys from a table
P_CHECK_TABLE_EXISTENCE	SQL code to control the existence of a table on a BDD
P_DELETE_LOG_EVENTS	SQL code to remove a log line from the program
P_INSERT_LOG_EVENT	SQL code to insert a log line of the program
P_UPDATE_LOG_EVENT	SQL code to update a log line of the program
P_CREATE_TABLE_LOG_ENT	SQL code to create the program's LOG table (header)
P_CREATE_TABLE_LOG_LIG	SQL code to create the program LOG table (lines)
P_CREATE_FROM_SELECT	SQL code that creates a table from another (ex: SELECT * INTO [TABLE] FROM [ORIGINAL_TABLE] WHERE 0 = 1)

Note: You can use dynamic parameters in connection strings by using {?1}, {?2}... Those parameters will be replaced by those mentioned in the Job (explained later)

#### Mongodb

onfiguration							
nnections Log - Error	rs SHS Analyzer	SQL File	Mail	Webservices	Service/Client A	pp Dev.	
Connection List [20] -> MongoDB Rasp	pberry	~		Create New Con Try Connection		Save Connection Delete Connection	
Connection Parameters - Connection name	MongoDB Raspberry					Connection String Assistant	
Driver	MongoDB	$\sim$	Scan	Network for insta	inces		
MongoDB Connection String (ex	: mongodb://name:passwd4	@host:27017/db)					
mongodb://guizmox:r	myPassword@192.1	68.0.105:27	)17				]
Advanced Parameters —							
	Generic BSO	N document	•				
-Advanced Parameters —	Generic BSO	N document					
-Advanced Parameters —	Generic BSO Generic BSO	N document					
-Advanced Parameters —	Generic BSO Generic BSO Fuzible docu	N document					

The syntax of a MongoDB connection string can easily be found online: <u>https://www.connectionstrings.com/</u>

In addition, the MongoDB driver treats collection data as BSON documents by default. However, you can choose to use a Fuzible-specific type (containing additional METADATA) or simply as strings. It all depends on the use.

#### Files

onfiguration								
nnections Log -	Errors SHS Analyzer	SQL	File	Mail	Webservices	Service/Client A	App Dev.	
Connection List – [1] -> Local Path			~		Create New Conn Try Connection			Connection Connection
Connection Paramete								
Connection name Driver	Local Path CSV File	~			Browse Filesystem		Connection	String Assistar
Diivei	Covinc				browsernesystem			
	S)FTP URL (ex1 : c:\\MyFiles\\   ex2 purce\Workspaces\Fuzil				ebug\FILES\			
	ource\Workspaces\Fuzil				ebug\FILES\			
C:\Users\guizm\so	s				ebug\FILES\			
C:\Users\guizm\so	s	ble\Fuzil			ebug\FILES\			
C:\Users\guizm\so	S Local Path Local Path	ble\Fuzil			ebug\FILES\			
C:\Users\guizm\so	Local Path Local Path Network FTP	ble\Fuzil			ebug\FILES\			

You must choose here the local or network path that leads to the files we want to process in a Job.

If you change "Source Type," you can also access the (S)FTP settings:

onnections	Log - E	rrors SHS A	nalyzer	SQL	File Ma	il Webs	ervices	Service/Clie	ent App	Dev.		
Connection [23] -> FI	List TP Raspber	rry			~		New Conne onnection	ction		Save Con Delete Cor		
Connection	Parameter	s ———										
Connection	name	FTP Raspberr	у						Co	nnection Stri	ing Assistant	
Driver C		~		-			Det	1100			()	$\sim$
		FTP URL (ex1 : c:\\M	lyFiles\\   ex2 :		vse Filesyster	n 	Date	ad/MM	1/уууу ~	Decimals	, (comma)	
Local Path, Net	twork Path, (S)	FTP URL (ex1 : c:\\M	lyFiles\\   ex2 :			n 	Date	es dd/MM	//yyyy	Decimais	, (comma)	
Local Path, Net	twork Path, (S)	FTP URL (ex1 : c:\\M	lyFiles\\   ex2 :	mysftp.myd		n		es dd/MM	·/yyyy ~		, (comma)	
Local Path, Net	twork Path, (S)	FTP URL (ex1 : c:\\M		mysftp.myd		n 		es dd/MM	?	Port	21	
Local Path, Net 192.168.0 Advanced Pa Source Type	bwork Path, (S)	FTP URL (ex1 : c:\\M		mysftp.myd	lomain.com) :	••	•••••	connection				
Local Path, Net 192.168.0 Advanced Pa Source Type Username	twork Path, (S) 0.101 arameters	FTP URL (ex1 : c:\\M		mysftp.myd	lomain.com) :	•••	•••••	connection	?			

#### **Webservice REST**

onnections	Log - E	rora SUS	Analyzer	SQL	File	Mail We	bservices S	ervice/Client App		Advanced		Τ
		nois sea	Analyzer	SQL	File	iviali vve	Diservices 5	ervice/Client App	þ	Advanced		
Connection						Creat	e New Connec	tion		Save Conn	ection	
[9] -> Fi	uzible Webs	ervice de dé	nonstration		~	Try	Connection			Delete Coni	nection	
Connection	n Parameter											
Connection			bservice de	démonst	tration				Con	nection Strir	a Assistant	
connection				demonst	liauon						-	
			/				Dates	dd/MM/yyyy	$\sim$		, (comma)	
Webservice U	Rest API JRL (ex : http://n vww.fuzible			ex2 : https:/	//mydomain.:	salesforce.com/s	ervices/data/v51.0/)			Decimals	, (connit)	
Webservice U	JRL (ex : http://n vww.fuzible			ex2 : https:/	//mydomain.	salesforce.com/s					, (connit)	
Webservice L https://w	JRL (ex : http://n	e-app.com			//mydomain.							
Webservice L https://w Advanced I Choose Terr	JRL (ex : http://n vww.fuzible Parameters	e-app.com	pi/apirest.php		//mydomain.		ervices/data/v51.0/)					
Webservice L https://w Advanced I Choose Terr	VRL (ex : http://n vww.fuzible Parameters nplate (optiona 1=val1;p2=val2	e-app.com	pi/apirest.php	: API	//mydomain.		ervices/data/v51.0/)					
Webservice U https://w Advanced I Choose Terr Headers (p1	VRL (ex : http://n vww.fuzible Parameters nplate (optiona 1=val1;p2=val2	e-app.com	pi/apirest.php Generic Rest	: API ader)	v	✓ Loa	ervices/data/v51.0/) d Template			• ?		
Webservice U https://w Advanced I Choose Terr Headers (p1	VRL (ex : http://n vww.fuzible Parameters nplate (optiona 1=val1;p2=val2	e-app.com	jeneric Rest Seneric Rest API auth (he Value (	: API ader) keep empt	y for dynam	Coa	ervices/data/v51.0/) d Template					

A webservice is a little more complicated to set up because there are several ways to interact with them, that's why a drop-down menu allows you to simplify the task by loading a "template" for some known APIs.

onnect	ion name	Faceboo	ok API				Con	nection Stri	ng Assistant
river	Rest API		$\sim$		Date	es dd/MM,	/уууу 🗸	Decimals	, (comma)
Webservic	e URL (ex : http://n	nydomain.co	m/glpi/apirest.php   ex2 : https://mydom	ain.salesforce.	com/services/data/v51.0	n:			
	/graph.faceb	ook.com	/						
	d Parameters	il)	Facebook Graph API	~	Load Template				
Headers	(p1=val1;p2=val2	)	Generic Rest API Nuxeo API (NxQL)						
Authoriza	ation Method		GLPI Rest API Salesforce Rest API	So val2	ope Authorization	=client cre	dentials		
			Salesforce Rest API (SOQL) Cegid Webservices	ten l				en	
			Youtube v3 API Microsoft Graph API	45	5984123218489		•••••		

By choosing an example from the list and loading it, the fields will automatically be pre-filled to simplify the setup and understanding of the API connection manager.

If you want to configure a connection to an unreferenced API by yourself, use "Generic Rest API", and manually configure the settings, as well as the authentication mode among the list of those proposed.

The example below shows the setting of youtube's API v3.

Connect	tion Parameter	s								
Connect	tion name	Youtube AF	1 v3					Conr	nection Stri	ng Assistant
Driver	Rest API	×	,			Dates	dd/MM/yyyy	$\sim$	Decimals	, (comma) 🗸
Webservie	ce URL (ex : http://r	mydomain.com/gl	pi/apirest.php   ex2 : https:/	//mydomain.s	alesforce.com/s	ervices/data/v51.0/) :				
https:/	//www.googl	eapis.com/y	outube/v3/							
Advance	ed Parameters									
Choose 1	Template (option	al) Y	outube v3 API		~ Loa	d Template				
Headers	(p1=val1;p2=val	2)								
									_	
Authoriz	ation Method	А	Pl auth (param)	~	Key :	key				
			Value (keep empt	y for dynam	ic retrieval) :	•••••	•••••	•••••	• ?	
			Optional URL to	get value o	dynamically :					
										7
Proxy UF	RL (if needed)			Proxy p	oort	Proxy Type	NONE ~	Use	r,Pwd ,	

For other APIs, apart from entering the URL, one must also provide all authentication information and sometimes additional header information. For example, a call to the GLPI API requires a dynamic key... You have to add the URL from which you'll retrieve that token (Optional URL to get value dynamically) :

Each connection to the webservice will then automatically first invoke a call to get that token and will allow you to call any other API object.

nnections Log - I	Errors SHS Analyzer SQL File Mail Webservices Service/Client App Dev.	
Sintections Log - L	енова ана мнанузет аке тте мат мераетикез зетике/опент.жрр Беч.	
Connection List :	[93] -> API GLPI V Connection String Assistant	
Connection Name :	API GLPI Try Connection	I.
Driver :	Webservice REST 🗸 🗸	
Webservice URL (ex : ht http://	ttp://mydomain.com/glpi/apirest.php) : /glpi/apirest.php/	
http://		
http:// Parameters :	/glpi/apirest.php/	
Parameters : Proxy URL	/glpi/apirest.php/ Proxy Port val2) Authorization:user_token Ida3wdPGO98gmJowcjZTiu4jAzqrIF3IPVM4noji;app- token:NUjjYrbZik3s0YYW6jKxis9TvyfZTgwfVUB0S33U	
Parameters : Proxy URL Headers (p1xva11;p2xv	/glpi/apirest.php/ Proxy Port val2) Authorization:user_token Ida3wdPGO98gmJowcjZTiu4jAzqrIF3IPVM4noji;app- token:NUjjYrbZik3s0YYW6jKxis9TvyfZTgwfVUB0S33U	

Last example, to configure the Microsoft Graph API, we use an OAuth 2.0 authentication with some specific settings which is described in the online API documentation.

Connecti	on Parameter	s								
Connect	ion name	Microsoft Gr	aphQL					Con	nection Stri	ng Assistant
Driver	Rest API	~				Dates	dd/MM/yyyy	$\sim$	Decimals	, (comma) \vee
Webservic	e URL (ex : http://r	nydomain.com/glpi	/apirest.php   ex2 : https://mydo	omain.salesfor	ce.com/s	ervices/data/v51.0/) :				
https:/	/graph.micro	osoft.com/v1.	0/							
Advance	d Parameters									
Choose T	emplate (option	al) Mi	crosoft Graph API	~	Loa	d Template				
Headers	(p1=val1;p2=vali	2)	ontent-Type=applicati	on/json						
Authoriza	ation Method	OA	wth 2.0	~	Scope	https://graph.microsc	oft.com/.default			
			HTTP Params (p1:	=val1&p2=va	al2) :	grant_type=c	lient_credent	ials		
				Toke	n URL	https://login.microsof	ftonline.com/a65ed	48768	pi89-8999-j454	7-54898ft15g5sdf/oa
			Client ID + Cli	ient Secret	5654891	zef-azerffgg5684-454	5fgd-547741	•••••	•••••	?
Proxy UR	L (if needed)		P	roxy port		Proxy Type	NONE ~	Use	er,Pwd ,	

#### Mail

onnections	Log - Errors	SHS Analyzer	SQL	File	Mail	Webservices	Service/0	Client App	р	Advanced		
Connection [10] -> Fu	list Izible Email			~	C	reate New Conn				Save Conn		
Connection	Darameters					Try Connection				Delete Con	nection	
Connection		ole Email							Con	nection Stri	ng Assistan	t
Driver N	lailbox	$\sim$				Da	tes dd/N	/М/уууу	$\sim$	Decimals	, (comma	)
Customized Co	nnection String (Use a					t; <i>get_protoco</i>	o/=IMAP;	AUTH_	PRO	DTOCOL=	imap; <b>poi</b>	
Customized Co SERVER_ _SEND=	nnection String (Use a _ <b>SEND =</b> SSI0.04 465; <b>PORT_RE</b>	ssistant) : /h.net; <i>SERVER</i>				t; <i>get_protoco</i>	o/=IMAP;	AUTH_	PRO	DTOCOL=	imap; <b>poi</b>	
Customized Co	nnection String (Use a _ <b>SEND =</b> SSI0.04 465; <b>PORT_RE</b>	ssistant) : /h.net; <i>SERVER</i>	SERNAM		aume@	t; <i>get_protoco</i>	o/=IMAP;	AUTH_	PRO	DTOCOL=	imap; <b>poi</b>	
Customized Co SERVER_SEND= Advanced Pa Protocol	nnection String (Use a <b>SEND</b> =SSI0.00 465; PORT_RE	ssistant) : /h.net; <i>SERVER</i> : <i>CEIVE=</i> 993; <i>U</i>	SERNAM	<b>ME=</b> guilla	aume@	t; <b>get_protocc</b> Tuzible-app.con	o/=IMAP; n; <i>PASSV</i>	AUTH_	PRO	DTOCOL=	imap; <b>poi</b>	

The "Mail" connection string is unique to Fuzible. The "Connection String Assistant" can be used to help build it, but in short, the following parameters are required:

SERVER\_SEND: SMTP URL

SERVER\_RECEIVE: POP or IMAP URL

GET\_PROTOCOL: The reception protocol: IMAP or POP

AUTH\_PROTOCOL: The security protocol: TLS (10,11,12,13), NONE

PORT\_SEND: SMTP Port

PORT\_RECEIVE: IMAPT/POP Port

USERNAME: the associated email address

PASSWORD: the password associated with the email address

SSL: 1 or 0

You can also set up a proxy.

#### **Active Directory**

onfiguration								
nnections Log - I	rrors SHS Analy	er SQL	File	Mail	Webservices	Service/Client	App Dev.	
Connection List [21] -> Fuzible Act	ve Directory Demo		~		Create New Con Try Connectio		Save Connection Delete Connection	
Connection Paramete Connection name Driver	Fuzible Active Di Active Director		)				Connection String Assista	nt
LDAP URL (ex : LDAP://DC=	mydomain,DC=en) (ex2 : LI	AP://mydomain)	:					
Advanced Parameters Compatibility params (	don't edit default values	if you don't kn	ow how t	o use those	e parameters)		Load default params.	
P_AD_SEARCH_GROUP=(&( P_AD_SEARCH_GROUPS=(& P_AD_SEARCH_GROUPS=(& D_AD_SEARCH_USER=(&(o	(objectClass=group))	ory=person)(CN		EARCH_PRO	)PERTY]=*))			
P_AD_SEARCH_OSERS=(0(0								

The connection string is rather simple, here are two examples:

LDAP://DC=fuzible,DC=fr

LDAP://fuzible.lan

In addition to the connection string, you can modify compatibility settings:

P_AD_SEARCH_GROUP	Search query for a specific group	SEARCH_PROPERTY will be replaced by the value set in the Job (see below). This is the "key" field available in the AD that defines the uniqueness of a group
P_AD_SEARCH_GROUPS	Group search query	
P_AD_SEARCH_USER	Query to search a specific user	SEARCH_PROPERTY will be replaced by the value set in the Job (see below). This is the "key" field available in the AD that defines a user's uniqueness
P_AD_SEARCH_USERS	User search query	

This tab sets up all the software's LOG options. By default, ". TXT" LOG files are stored here :

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

... but it can also be integrated into a database, or sent by email.

Configuration					—		×
Configuration							I
Connections Log - Errors	SHS Analyzer	SQL File	Mail Webservices	Service/Client App	Dev.		
SQL Log - Connection String			guizm\source\Workspa oreign keys=true	ces\fuzible\Fuzible\bin	ı\x64\Debug\wir	1-x64	
Driver	SQLite	~		Crea	te required LOG ta	ables	1
Mail Log - Connection String	OTOCOL=IN	1AP; <b>auth_p</b> i	ffice365.com; <i>SERVER</i> rotocol=NONE;PORT com;PASSWORD=m				
	Configure you	ur mail	<ul> <li>Use SSL connection</li> </ul>	Authentificatio	on NONE	v	
Keep log until (days) Keep processed files until (day	8 30						
Don't send mail if job fini	shed without any erro	r					
Show CPU/RAM alerts							
Abort Job when errors exceed	s 10	)					
					Save Configurati	on	

#### SQL Log

It is strongly advised to change SQL instance to store LOG, by default, it is the local Fuzible database that is used, but if you want to manipulate/get the LOG from elsewhere, you should use a network database (MySQL, SQL Server, Oracle, Postgres...)

In this case, the program will automatically create the required SQL tables in this new instance.

#### Mail Log

By default, this field is empty. This connection to an inbox allows you to send Job reports to one or more email addresses. The connection is configured like any connection (as seen above).

Abort Job when errors exceeds	If this number is exceeded during the execution of the Job, then it is interrupted
Keep log until (days)	The LOG files and SQL entries can clean themselves beyond a number of days you can set here
Keep processed files until (days)	With a "File" connection string, Fuzible can clean up the directory by erasing files that are too old. The number of days of file preservation is defined here.
Don't send mail if job finished without any error	If the job is set up to send an e-mail at the end of the process, and if it went smoothly, you can avoid the mail to be sent
Show CPU/RAM alerts	When the CPU reaches 100%, the warning is logged, allowing you to check the need for additional resources on the server/computer that hosts the app

Fuzible contains a data analysis engine: it analyzes all the source data, and can then create SQL tables as accurately as possible when they are non-existent and a Job uses a database as a Target.

These operations are very resource-intensive and can easily solicit the machine at 100%. To avoid this saturation, Fuzible can be set to use only a part of the available resources.

Configuration				•		_		$\times$
Configuration								
Connections Log - Errors SHS Analyz	er SQL	File	Mail	Webservices	Service/Client App	Advanced		
Multithreading (cores) 8								
Parallel Threads when loading/copying huge S	QL dataset	s or CSV	files	2				
O Cast rounded .0 decimals as integers (ex :	5.00 will b	e interpr	eted as a	n integer)				
O Cast numbers starting with '0' as integers	(ex : 0150 v	vill be in	terpreted	as an integer)				
Multi-target Queries : write both targets i	n parallel (C	PU-RAM	1 intensiv	e)				
						Save Configura	ation	

Multithreading	Number of CPU cores that can be used by the software (minimum - 2)
Parallel Threads when loading/copying huge CSV	Fuzible usually loads the whole Source Data in RAM, and then proceed with the Target
datasets or CSV	replication/synchronization. In case of huge CSV or SQL tables (millions of rows), it loads data in smaller
	chunks. That way of loading data can be performed using more than one thread to increase speed : the
	next chunk can be loaded while the last one is copied in the Target.
	The more you add threads, the more you'll need CPU power and RAM.
Cast rounded .0 decimals as integers	When analyzing numerics, Fuzible can consider 5.00 to be an integer rather than a decimal
Cast numbers starting with '0' as integers	When analyzing numerics, Fuziblz can consider values like 0546 or 0000477 to be integers rathen than
	strings
Multi-Target Queries: Write both targets in	For a Job using multi-target feature, you can choose to the 2 targets in parallel for better performances
parallel	(requires LOT of computing power)

## SQL

All the settings for the "SQL" connections are grouped here.

Configuration		1000 Direct Stream Mode) 100							
Configuration									
Connections Log - Errors	SHS Analyzer	SQL	File	Mail	Webservices	Service/Client App	Advanced		_
Command timeout	360								
Transaction size	128								
Bulk Insert Mode Rows/Batch :	1000	]			_				
Rows quantity to get before co	pying (Direct Strea	m Mode	)	100					
Synchro. table Log	fuzible_synchro_re	cords							
O Auto shrink tables									
		•							
							Save Config	uration	

Command Timeout	Timeout to execute an SQL command
Transaction size	When a Job whose target is an SQL database, "INSERT/DELETE/UPDATE" statements are framed in
	transactions (which allows a ROLLBACK if the query fails).
	The number of statements / transactions is to be defined here. The larger the number, the more the
	target database resources will be solicited
Bulk Insert Mode Rows/Batch	Applies when configuring a Job using the "Bulk Insert" mode. It defines the amount of rows you want to
	copy at one. le : if there are 10 000 rows to insert and you did choosed a 1000 rows batch, 10 Bulk
	batches will be sent to the target Database.
Row quantity to get before copying (Direct	The « Direct Stream » feature is configured in the Job and is mostly used when querying huge SQL tables
Stream Mode)	(millions of rows) : To avoid memory issues, data is loaded and transferred to target in small chunks.
	You can set here the length of each chunk (rows quantity).
	Increasing the value requires more RAM.
Syncho. Table LOG	For Jobs running in "Synchronization" mode, synchronization statuses are stored on a separate table
	(for consultation and information). You can choose the name here.
Auto shrink tables	An option to clean a Target table after processing it.

## File

All the settings for the	"FILE" connections	s are grouped here.
--------------------------	--------------------	---------------------

Configuration							•			_		$\times$
Configuration	n											
Connections	Log - Errors	SHS Ana	alyzer	SQL	File	Mail	Webservices	Service/Client App	Dev.			_
Working path		Ρ	rocesse	d								
Source files mov	/e path	E	xport				🔴 Add d	atetime prefix				
CSV separators		<i>i</i> /	\ <b>t</b> .									
CSV/XLS : Fo	orce Integratior	n of row(s)	not ma	tching th	ne heade	er length						
									Save Co	onfigurat	ion	

Working path	When a Job is set with a "File" Source connection, and it has been set up to move those files when the
	Job is finished, the directory in which they are moved is defined here
Source files move path	Unused for now
Add datetime prefix	In addition to moving files at the end of a Job, these files can also have a prefix in the form of
	"YYYYMMddHHMMss_Myfile.xxx" to eventually facilitate their subsequent search (if needed)
CSV separators	List of accepted CSV separators ("\t" means "tabulation"). You can add more if you are dealing with files
	using some other separator.
CSV/XLS : Force Integration of row(s) not	If for some reason your source files have inconsistent row length, you can force the integration of those
matching the header length	rows or bypass them. In any case, a "WARNING" LOG message will be triggered

## Mail

All the settings for the "MAIL" connections are grouped here.

Configuration									-		×
Configuration											
Connections L	og - Errors	SHS Analyzer	SQL	File	Mail	Webservices	Service/Client App	Dev.			_
Admin email addre	ess										
Max. length before	e sending shee	et as an XLS attac	hment		65536						
Max. attachment fi	ile size (in kB)	2048									
Timeout		30000									
								Save Co	onfigurati	ion	

Admin email address	The program administrator's email (will be cci'ed of any mail produced by a Job)
Max. length before sending sheet as an XLS attachment	When a Job is configured with an email address as a target, you can choose the data to be included in the mail body.
	However, if this content is too big, Fuzible can, instead, create an EXCEL file that will then be attached to the mail. This setting shows the maximum number of characters from the Data Source before the Job switches to "attachment" mode.
Max. attachment file size	Maximum size of an attachment. Beyond this limit, the attachment will not be sent
Timeout	Timeout to run an operation on the mail server

## Webservices

All the settings for the "WEBSERVICES" connections are grouped here.

Configuration					=		—		×
Configuration									
Connections Log - E	rrors SHS Analyzer	SQL	File	Mail	Webservices	Service/Client App	Advanced		
Timeout	30000								
Default encoding	ISO-8859-1 UTF-	3							
• Save responses in	source : Allow schema a	Iteration							
SOQL (Salesforce)	Queries : Get Records D	atatable	Only						
							Save Configura	ation	

Timeout	Timeout to run an operation on the remote server
Default encoding	When A WS's data is processed, the program determines the encoding from what the server answers. If this information is unavailable, a custom encoding may be forced by default. 2 values to indicate: the 1st for the "REST API's" and the 2 <sup>nd</sup> for the Nuxeo API.
Save responses in Source: Allow add-alter columns	When sending data to a webservice, this one sends back an answer (in XML or JSON format). Those answers can be retrieved and stored in the Source database (if SOURCE-BDD) or as a file (if other). In case of a Database, you could allow Fuzible to modify the SQL tables if the data returned by the API is not compatible with them.
SOQL Queries : Get Records Datatable Only	This setting allows, when using the SOQL language, to retrieve only the records datatable : will not get the additional tables : attributes and query summary

## Service/Client App

Fuzible comes with a "Service/Client" module.

The "Service" background application is the subject of a dedicated paragraph, but in short, it executes Jobs that are invoked either by a user (via the "Client" application) or by the Orchestrator (Jobs Orchestrator)

The "Client" app is a mini-application that simply allow any user to trigger a Job execution remotely, whenever he wants.

Connections Log - Errors	SHS Analyzer	SQL File	Mail	Webservices	Service/Client App	Dev.		
iQL Service app - Connection s	trin <b>Data Sou</b> \Fuzible.db	r <b>ce=</b> C:\Users ;	\Guizm\	source\Workspa	ces\Fuzible\Fuzible\bi	in\x64\Deb	oug	
Driver	SQLite		~					
Create W	indows Task (Task	Manager)	Cre	ate required servi	ce tables (planification, s	stack, jobs)		
Vorking userspace	GUIZM		~					
Parallel Jobs	2							
ceep stack until (days)	30							
Client app flooding delay (min)		15						
Sharing the 'Client' app requi	res you to provide	'FuzibleClient.	exe', as w	ell as 'CLIENTAPP.	INI' files to users			

Connection string: By default, Fuzible uses the local SQLite connection, but this way of working is not really recommended. Indeed, the "Service" application is intended to communicate with the "Client" application, distributed to any user. In a network environment, you probably do not want anyone to have the Fuzible network path opened and accessible to anyone. On the other hand, your network can be configured so that client computers can make calls to a database instance.

Create Required Service Tables	This button allows you to automatically create all the required SQL tables for the service app to work
	properly (in case of a change in connection string)
Create Windows Task	Creates the "Fuzible Service App" task in the Windows task manager.
Working User	This is the account the "Service" app uses to work. This account corresponds to one of the Fuzible users:
	the service application can only interact with one of the users, to avoid anarchic management of the
	orchestration and Jobs made available to the Client application.
Parallel Jobs	The Service app detects the Jobs invoked as they go along. It can run several in parallel but beware of
	the risk of overdlow. Here you choose the number of Jobs it can launch in parallel: This setting should
	be based on the resources allocated to the server/computer that runs the application.
Keep Stack Until (days)	Retention time before the Jobs stack LOG must be cleaned.
Client App Flooding Delay	The "Client" app allows users to remotely launch Jobs. They could trigger the same Job several times,
	flooding the system. This setting allows you to set a delay between 2 successive launches of the same
	Job, in order to prevent them from "spamming" the queue and overloading the system.

#### Dev.

Here are several parameters related to how the program internally works. Even if you would probably never need to change those settings, I choose to make them available. However, it is not advised to change them without understanding how they work, it could compromise your existing Jobs. The online forum may allow you to chat with other users about it.

Configuration								
Connections Log - Errors S	HS Analyzer	SQL I	ile Mail	Webservices	Service/Client App	Advanced		
CSV/EXCEL File Analysis								
Maximum rows analyzer (header,	separator)		10000					
Source CSV splits when row coun	t exceeds		100000	Don't ch	eck files coming from (S	)FTP		
Header analyzer - Depth analysis	resemblance of	fset	0,5					
Header analyzer - Depth analysis	unicity offset		0,75					
Header analyzer - Row offset befo	ore depth analys	is	100					
QL Operations								
Errors overflow (more failed queri Max decimals (more will be round		error) 32						
Query characters (debug)	,		384					
Security			<b>C</b> 111					
Shared Users (all users shares			GUI	ZM	$\sim$			
Software Registration Method	Mail (	Offline)	~					
Miscellanous								
Enable Query Assistant	Analyze Fi	les - Max.	Size :	50485760 She	ell operations timeout (m	nin) 120	48	30

XLS/CSV maximum rows analyzer	Fuzible automatically scans the EXCEL and CSV files to determine if the first line is a header.				
	In the case of files with a large number of lines, the ent are analyzerdoes not necessarily need to				
	analyze all the lines to detect it. The maximum number of lines to be analysed can be set here				
Source CSV split when row count exceeds	If the source is a CSV file, the program systematically counts the number of lines of it. If this number				
	exceeds a certain amount, Fuzible can process the file into several chunks to avoid overload of the				
	server's RAM. The value indicates the maximum number of rows contained in each chunk.				
Don't check files coming from (S)FTP	In connection with the previous option, if the files come from a server (S)FTP, it is possible to bypass the				
	line count, because this analysis requires downloading the file, which can severely penalize the				
	performance <b>Of</b> the program.				
Header Analyzer - Depth Analysis Resemblance	A threshold that determines the percentage of resemblance between the first line of the analyzed file				
Offset	and all the others. The resemblance is calculated internally by doing several tests on the file. The scan				
	can be displayed if the Job is configured in "Debug" mode				
Header Analyzer - Depth Analysis Unicity Offset	When automatically analyzing the contents of an EXCEL or CSV file, Fuzible determines the uniqueness				
	of the first line of the file compared to the others.				

	- If the percentage of uniqueness exceeds thethreshold, and in addition, the percentage of			
	resemblance of the first line is above the "Offset Resemblance" threshold, it is considered that			
	the first line is not a header			
	- If the percentage of uniqueness exceeds the threshold, and in addition the percentage of			
	resemblance of the first line is below the "Resemblance Offset" threshold, the first line is considered a header			
	- If the percentage of uniqueness is below the threshold, and in addition the percentage of			
	resemblance of the first line is less than or equal to the threshold "Resemblance Offset", the			
	first line is considered a header			
	- If the percentage of uniqueness is below the threshold, and in addition the percentage of			
	resemblance of the first line is above the "Resemblance Offset" threshold, it is			
	considered that the first line is not a header			
Header Analyzer - Row offset before depth analysis	In the case of files with few lines, header detection can be tricky (sample too small to calculate a percentage of resemblance between the first line and all the others). Below the threshold (number of lines) entered here, the program goes into "end" mode. It will finely analyze the file, and play on the 2			
	parameters mentioned above to determine the header			
Max Decimals	In "target- BDD" mode, defines the maximum number of decimals tolerated when inserting data(where source data would have for example 35 digits after the comma and one wishes to limit this amount)			
Errors Overflow	If an executed query didn't work, Fuzible marks it as a "warning." You can set here a threshold of			
	queries that have not resulted from whichthe Fuzible considers it to be an error and no longer a "warning" (will trigger an error message)			
Query characters (Debug)	When a Job is set up with log in "Debug" mode, and the Job target is a BDD, all INSERT/DELETE/UPDATE queries will be entered into the DEBUG file. The volume can be considerable,			
	so it is possible to limit the amount of characters built into the LOG			
Shared Users	By default, any Windows user has its own Fuzible session : connections, configuration, jobs are not shared with the others. In that mode, you can still import Jobs from another session and load another			
	userspace but in a « read-only » mode only. In your organisation, you could need a shared session : any user will be routed to the one who's			
	choosed here. The whole program is then shared with all computer users.			
	It implies some confidentiality compromises (ie : connection strings are visibles by any Fuzible user)			
Software Registration Method	How to communicate with Fuzible's <b>SerVer</b> when you want to save the program. You should leave "Webservice" by default. The "Mail" mode will only serve in case you are unable to communicate with the server (offline mode)			
Enable Query Assistant	Turns the assistant on interface queries (colorization, consistency controls, input proposals) is activated			
	ordisables. The assistant can consume a lot of resources because in case the Source is a database, he			
	asks him to know the tables available and the fields of each table.			
	If it's a file-type source, it scans the directory to find the names of the files, and scans each of them to find the headers.			
Shell Operations Timeout	When a Shell Pre or Post-Job command is scheduled, the maximum default execution time is 60			
	minutes. Beyond this time, the task is interrupted. So we can intervene here on this parameter			
	This setting is also used by the "Service" app. When she performs Jobs, she casts Fuzible in a set-up manner. The program is then subject to the same Timeout rules			
Json Parser: Replace special tanks in columns	When a JSON file (or webservice response) is interpreted, column names sometimes contain special			
names	characters: it can be decided here to replace them with a more conventional character(the underscore)			
Analyze Files – Max Size	When using the Query Assistant with "FILE" as the Job Source Connection, it will parse the file you are			
	querying to find all the columns and add them to auto-completion system. If the file reaches a max. size			
	(in kb), it won't be analyzed because it will consume too much power (CPU/RAM). In case you don't			
	know the available columns in the file, you should first perform a "SELECT * FROM" to show available			
	columns and use them in your Query after that			

## Tools

#### Export Job (XML)

This menu allows you to export a Job using XML format: This extraction contains all job settings, associated connections (as well as those that may be called by script fields), and queries. The file is encrypted so that it can only be imported into another environment if the user knows the password: it is the job's default password.

#### **Import Job (XML)**

This menu allows you to import a Job using XML format: Integration into your environment includes creating associated connections with the Job, settings, and queries. The Job password is required to perform the importation.

#### **Reorganize Jobs**

You can reorganize Jobs here. Indeed, it is possible to create "multi-step" jobs (which are no more or less than several jobs launched one after the other.

This menu allows you to:

- Drop a main job to another job as a step.
- Extract a "step" in a "multi-step" Job to put it as a single Job.
- Reorganize the steps order in a "multi-step" Job.

in Jobs —	 Sub-Jobs (Steps)	
4] - GLPI -> Ysera : Tickets 36] - Debug : MySQL -> Postgres Full Synchro 35] - MySQL -> Postgres : Synchro Complex WHERE 31] - Postgres -> Access : Full Copy 41] - Synchro Error 17] - Test Access -> SQLite	[9-2] - Payroll data to Datawarehouse [9-3] - Test Partial delete source	Move dow
<ul> <li>15] - Test ODBC Excel -&gt; SQLite</li> <li>16] - Test SQLite -&gt; ODBC Excel</li> <li>9] - Accounting Data To Datawarehouse</li> </ul>		Move up
34] - ALL SCENARIO JOB 13] - CrossQuery Demo		
11] - Demo Job 002 33] - TUTORIALJOB - WORDPRESS		

Multi-step Jobs are bolded. When you click on one of them, you see the list of sub-jobs.

#### **Move as Sub -Job**

Moves a Job into another one. It will then become a sub-job. If it is moved to a Job that already has sub-jobs, it will be positioned last.

#### Move as Main Job

Extracts a sub-job to make it a main job. You will be asked to set a password for him.

Moves a sub-job to change the execution order of a "multi-step" Job.

**Note:** If you move a Job to a sub-Job, and automatic launch schedules are associated with it, you will be alerted, and the app will ask to delete or maintain this schedule.

### **Import External Job Parameters**

You can import a job from another user here (you have to know the password)

Import Job parameters fr	om another userspace		-		×
	$\checkmark$	Load another usersp	ace		
BATMAN GTRADMIN		~	Import Job	paramet	ers
GHVEMIN			Car	ncel	

Shows the list of available users.

... then the list of Jobs associated with it:

B Import Job parameters from a	another userspace		-		×
BATMAN	$\sim$	Load another userspa	се		
		~	Import Job p	aramete	ers
[1] CSV -> XLS : This is a	sample Job		Can	cel	

This information is read from the Fuzible's configuration file. You can also load another "Load Another Database file" to retrieve Jobs from other environments.

Note: Importing a "multi-step" Job is not possible. They are displayed as normal Jobs, with the difference that they appear in black and not blue.

After entering the password, the Job is imported (filling the settings fields) and the associated connections, imported automatically if you do not have them in your list.

If a connection ID is mentioned in the Job queries (cross-join queries, multi-target), they are also imported and transcoded automatically. You don't have to do anything to set up the Job again.

After import, you then have to click "Save configuration" to create a new job, based on these new parameters.

## Load another Userspace (read-only)

You can load the whole configuration of another user, for example to launch one of its jobs.

On the other hand, it is impossible to change any parameter of its configuration and its jobs. Only the dynamic parameters field can be changed. This allows you to benefit from the dynamic setting, which may be necessary.

You can also change the orchestration of his Jobs, a crucial feature because if you urgently need to have an hand on the orchestration and the account owner is not here, you will need to be able to change the setting for him.

# Job Creation

# Job Configuration tab

Time to create Jobs.

ile Configuration Tools H Job Selection GUIZM	Help		
GUIZM		Create/manage a Multi-steps Job	
	~		Create new Step
Job Configuration Source	e Target Queries Log Viewer		
Job Description			
	Rename Jo	b Change Password	C . N . I .
	Delete Job	o Planification	Create New Job
Main Parameters			
Job Type	Data Replication 🗸		
Will copy a source data into a targe	et without any comparison		
			?
Dynamic Parameters You can write any text or any availa	ble command. Each parameter must be separated by a semicolon ( ; )		
	es, text fields, connections) by referencing them like this : (?1), (?2)	View Job With	Replaced Values
			~
Log			
LOG Level E	errors + Informations V Log in SQL		
Send Mail When Finished	W	Vrite e-mail adress(es) here (separated by a	semicolon)
Misc.			
-	Dumon Dest Commonde (Seurce (Target) if Job has Free		
Visible in Client App	Bypass Post-Commands (Source/Target) if Job has Erro	rs	
Command Line F	uzible.exe "GUIZM" "" ""		

# Set up the general settings of your new Job here.

# **Options**

Job description	A text field that allows you to describe the purpose of the Job.		
	This field also serves as an object for a Job whose Target is an inbox.		
Job Type	Data Replication (copy) or Data Synchronization		
	→ See "Job Type"		
Dynamic Parameters	Dynamic variables: allows you to dynamically change the behavior of some Job parameters/queries.		
	Dynamic Parameters %YYYY%MM<1M;[4]/		
	Each variable is separated by a "; " Up to 9 variables are accepted → See "Script language"		
Visible in Client app	Check this if you want the Job to be available in the Client App (can be then remotely launched)		
Abort next steps on errors	During a multi-step Job execution, you may want to abort next steps if errors are detected		
Bypass post-commands if job has	Avoid post-job commands if it has encountered errors		
errors	→ See "post-commands"		
LOG level	The desired LOG level		
	LOG Level Errors + Informations ~		
	Send Mail When Finished Errors		
	Errors + Informations		
	Errors, Informations, Details		

	<ul> <li>ERRORS: You will only receive WARNING and ERROR messages from the Job</li> <li>ERRORS - INFORMATIONS: You'll also have informative messages that indicate progression</li> <li>ERRORS, INFORMATIONS, DETAILS: You will also receive details of internal operations : calculations, information related to data analysis</li> </ul>
Log in SQL	If the SQL connection string is set up (Program Configuration), it is possible to define here if you also want the Job to be logged into that SQL database
Send mail when finished	You can enter e-mail addresses here (separated by a comma) : A report of the Job will be sent to these people (report containing the execution status, as well as the LOG files
Command Line	Code that allows you to launch the Job without launching Fuzible (silent mode) Command Line Fuzible.exe "GUIZM" "[34]" "88BuRGfN+VN8Ovvp0qCebQ==" "%DD%MM%YYYY" Arguments: User Job Number Job password Dynamic parameters (those seen in the Dynamic Parameters field)

# Job Type

# **Data Replication**

Data Replication	$\checkmark$		
rget without any comparise	on		
			- 1
		View Job With Replaced Value	łS
			~
	rget without any comparise	Data Replication	rget without any comparison

This mode is available for any type of target. It consists of simply copying data from a Source connection to a Target.

# Data Synchronization

Main Parameters			
- Maint and the cers			
Job Type	Data Synchronization $\checkmark$	Allow Delete, Update, Insert 🛛 🗸 🔵 Historize U	PDATED and DELETED rows
Will compare source and target	data, and update target according to	Allow Delete, Update, Insert	
		Allow Update, Insert	
Dynamic Parameters		Allow Update	?
You can write any text or any ava	ailable command. Each parameter mu	Allow Insert	
You can use them anywhere (qu	eries, text fields, connections) by refe	Allow Delete, Insert	View Job With Replaced Values
		Allow Insert, Tag Deleted + Updated row(s)	~
			•

This mode is only available for AD, File and database Target connections.

It compares what is sent from point A to what already exists at point B. To do this, Fuzible will dynamically transcode the Source query to be comprehensible by the Target (assuming that the source and target may be or different or/and share different column names)

The comparison is based on the search for a Primary Key.

Fuzible uses several methods to find a primary key and proceeds from the simplest to the most complex:

#### In SQL mode

1/ Searching for the primary key by querying the BDD schema (Target first, then Source)

2/ Search for unique keys by querying the BDD schema (Target first, then Source)

3/ Automatic detection by analyzing all the combinations of columns and values that exist in the Source (this method can take a lot of time if the number of rows is huge)

## In FILE mode

The automatic detection mode is used.

# In ACTIVE DIRECTORY mode

The Primary Key field is defined in the "Target" tab

It is then possible to define the behavior on the Target:

- Insert new Source rows missing in the Target.
- Update rows that already exist in the Target, but whose content is no longer the same in the Source.
- Remove rows that are in the Target, but no more in the Source
- TAG: If data is to be deleted in the Target (not existing in the Source anymore) you can chose to keep it but create a "SYNCHRO\_TAG" field using the "D" value (DELETED) on rows that should have been erased Similarly, the updated rows can be tagged "U" (UPDATED)- however, they will be updated.

The "Historize Updated and Deleted Rows" option creates or fills a table next to the target table (with the \_hist extension) that contains all rows that have either been deleted or updated during sync. It basically creates a table with the exact same schema and adds a new index column to it, as well as a timestamp column.

# Language Script

All Fuzible "text" fields accept scripted parameters that will be interpreted and replaced by associated values.

Those can be made Dynamic:

- A connection string (for example, changing the server, the user, the database, the path of a file...)
- Queries (for example, making dynamic filters)
- Add additional, custom columns to content.

By design, any "text" area of Fuzible understands and interprets script language. The script zones are framed by brace brackets.

Here are the variables that can be used:

- ?1, ?2, ?3...: parameters from the "Dynamic Parameters" field
- %MM: month number on 2 characters
- %YY: 2-character year number
- %YYYY: 4-character year number
- %DD: 2-character day number of the month
- %WW : 2-character week of the year
- %HH: Time of Day on 2 characters
- %mm: minute on 2 characters
- %SS: second on 2 characters
- %DTTS / DTTSMILLI: current date in Unix Timestamp format
- <XM : removes X months to MM (if existing) (ex : %MM<2M)
- <XY : removes X years to YY (ou YYYY) (if existing) (ex : %YY<2Y)
- <XD : removes X days to DD (if existing) (ex : %DD<2D)
- <XW : removes X weeks to WW (if existing) (ex : %WW<2W)
- >XM : add X months to MM (if existing) (ex : %MM>2M)
- >XY : add X years to YY (ou YYYY) (if existing) (ex : %YY>2Y)
- >XD : add X days to DD (if existing) (ex : %DD>2D)
- >XW : add X weeks to WW (if existing) (ex : %WW>2W)
- %USER: The user's name connected to the app
- %QUERYTARGETNAME: The name of the target.
- -%CT1, %CS1, %CT2, %CS2... : If specified in the dynamic parameters list, this setting will be replaced by the returned value from a Source pre-command (%CS1) or a Target pre-command (%CT1)

Examples (base: March 20, 2018, consider that ?1 = 100 et 2 = 'TEST')

SELECT * FROM MONFICHIER_{ %MM%YYYY}.csv	SELECT * FROM MONFICHIER_032018.csv	
SELECT * FROM MONFICHIER{%MM_%DD_%YYYY}.csv	SELECT * FROM MONFICHIER03_20_2018.csv	
SELECT * FROM MONFICHIER_{%MM%YYYY>3M}.csv	SELECT * FROM MONFICHIER_062018.csv	
SELECT * FROM MONFICHIER_{%MM%YYYY>3M>1Y}.csv	SELECT * FROM MONFICHIER_062019.csv	
SELECT * FROM MONFICHIER_{%DD<10D}.csv	SELECT * FROM MONFICHIER_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'	

Additional note about the use of script language:

- In the Dynamic Parameters field, each setting must be separated by a "; " this character reserved for separation cannot therefore be used in a dynamic value!
- Date "codes" can be used as dynamic parameters:
   What if you're assigning "MM" value to {?1} will be replaced by 03 (March if it is March)
   You can then schedule more complex things, for example (it's March 20, 2018):
   {%YYYY01<2Y-?1} will then be transformed as 201601-\_20TEST if ?1 has been set with value \_%DDTEST</li>

In addition to this setting, when a Job has been created, you can see a couple of key information (at the top of the screen): Creation date, last execution, and status of the last execution.

Job Description				
Creation Date : 19/12/2020 09:28:54 Last Modified : 14/01/2021 09:47:38		Rename Job	Change Password	C . N . I .
	- RUNNING TIME : 00:00:09 - ERRORS : 1 - WARNINGS : 79	Delete Job	Planification	Create New Job

When you save a Job, you will be asked to set a password. It prevents anyone to import your Job without asking your agreement, but it also serves to prevent unwanted users to launch it from the "Client" app.

This password can be changed, as is the name of the Job.

# Orchestration

If the "Service" app is set up correctly, the Job Orchestration tool is available to you. The purpose of this is to launch Jobs on pre-defined dates/intervals.

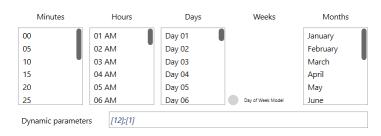
When you open a Job, you click "Orchestration," a new screen opens and allows to create, delete, and change your scheduling plan(s) for that Job.

Creating an Orchestration model works in two modes:

#### In the form of "days of the week"

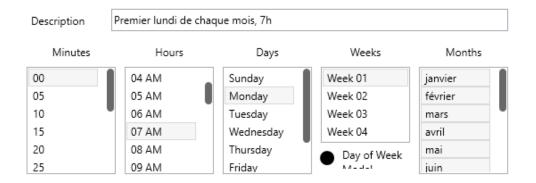


# In the form of "days of the month"

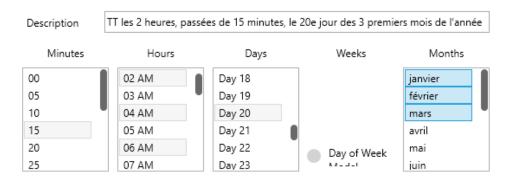


You will necessarily have to choose an item from each column to set an orchestration.

- If, for example, you want to start a Job on the first Monday of each month, at 07:00, you will set the schedule as follows:



- If you want to start a Job every 2 hours, past 15 minutes, on the 20th day of the first 3 months of the year, you will set the orchestration as follows:



A planification must include a description, and optional dynamic parameters: These are, by default, deferred from the Job but you could, for example, set multiple schedules for the same job with a different setting each time (ex:, launch a data replication on a preprod DB at 07:00, and then launch that same replication on a production DB at 08:00)

Description	Sunday, connection	is 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		May
25	06 AM	Day 06	Day of Week Model	June
Dynamic param	neters [12];[1]			

You can obviously edit or delete any of the planifications. You can also choose whether to make them active or not.

# **Planning Calendar**

Fuzible offers you a complete visualization tool of the current week schedules.

SHS Fuzible Data	Replicator, Synchronizer		- 🗆 X
File Configuration	Tools Help		
GUIZM	Export Job (XML) Import Job (XML) Reorganize Jobs	Create/manage a Mu	ulti-steps Job Create new Step
Job Configuration	Import Job from another Userspace Load another Userspace (read-only) Planification Week Calendar	get : Postgre Database Queries Log Viewer	

By clicking on this menu, Fuzible will generate a simple HTML file of the current week's schedules and display it in your default browser. This will give you a view of all the scheduled tasks and information on the unfolding of previous iterations.

^

### Example of a calendar:

Week 51 : 82 record(s)

	Monday, 14/12/2020	Tuesday, 15/12/2020	Wednesday, 16/12/2020	Thursday, 17/12/2020	Friday, 18/12/
00:00	- beb : MOSAIC → MAIL : Mérique BDD [Pland: Towa is: Inndi, 60] [Ages : grinnaus]iseni f, aronnes@seni. ft, ablanc@seni. ft, ifavere@seni.ft] [Sanna. Active] [Dag: Finaided without errors - Running Time : 8,5 sec.]				
06:05	- bob : COMETE ≫ Mousic : Mise a jour de la table de plazating	- Job - COMETE >> Monaic : Mine a jour de la table de planning	- bob : COMETE >> Mousie :: Muse a jour de la table de planning	- fob : COMETE >> Moust: : Mus a your de la table de planning	- Job : COMETE → Mosaic : Mise a jour de la table
	[Plazaf.: Toutes heures de fai 2005]	[Plandi : Teotre heures de Ré à 20h]	[Pland]. Toutes heures de fais 2005]	[Plandi. Torten heures de fai 2003]	[Flanif. : Toutes heures de 6h à 20h]
	[Arga: .ese.(2af)	[Ang. : react_fag]	(Arga: :reactAge]	(Aga: :resc.[Aga]	[Args.: ress;Clagt]
	[Statis: .Artive]	[Status : Active]	[Status Active]	[Status: Active]	[Stanus : Active]
	[Log: Finanked without errors - Running Time : 47 sec.]	[Log : Finalidad withour errors - Ranning Tures : 48,4 sec.]	[Log: :Finished withous errors - Ranning Tume : 47,5 soc.]	[Log: Finniheds without errors - Running Tume : 46,8 soc.]	[Log :]
08:45	- Job : ALL -> BD7 : Synchronization du Data Warshouse [Pland: Prod. Da India zu vondredi chaque 2h] (gaz. :WYTYTMADAVIAL5[4]) [Status - Active] [Dag : Familes without errors - Romning Time : 2,3 min.]	- Job - ALL -> BD <sup>-</sup> . Synchronismien du Data Warshouse [Pland: Prod. Du innd as windred chapte 3.] (prg NYTYPHEMACHA14]() [States. Active] [Gates. Active] [Dog - Familed without errors - Raming Time : 2,4 min.]	- Job : ALL -> BD7 - Synchroniumien du Data Warshouse [Pland: Fred. Do lundi av swadrefi chaque 20] (Jara: - NYTYTY-NANCHAL9][] [Status - Active] [Log - Franched writhout errors - Romning Time : 2,4 min.]	.160 : ALL -> BD7 - Synchronization du Data Warehouse [Plandi, Fred. Du lundi au voudrehi chaqua 20] (aga:	- Job : ALL -> BD7 : Synchronisation dn Data Ware [Fland: . Frod : Dn Inndi an vendredi chaque 2h] [Argn : 16YYYY9MDdv1M[4]] [Stans : Active] [Log : ]
07:00	<ul> <li>Job : MOGAUC = 0LB( &gt; MARL : Salarisis Sortis [Pland. : Toro iss Insidia 07b.] [Quana : activo info@gama.0] [Quana : Activo] [Log : Familubed without errors - Running Time : 9,2 sec.]</li> </ul>				
07:05	- Job : COMETE >> Moust :: Mise a jour de la table de planning	- Job : COMETE ~> Mosaic : Mine a jeur de la table de planning	- Job : COLETE > Mousit : Mise a jour de la table de planning	- Job : COLETE >> Morate : Mine a jour de la table de planning	- Job : COMETE -> Mosaic : Mise a jour de la table
	[Pland] : Toures heures de la 2005]	[Plant]. Tourns heures de frà 200a]	[Plandi. Totes heures de fa 2005]	[Plandi: Tottes heures de da 200a]	[Planif: : Toutes heures de 6h à 20h]
	[Equa:::eux:char]	[Eng. : react-fapt]	[Arga : resc./argh]	[Arga : reux-class]	[Args:: ress,clagt]
	[Equa:::Active]	[Eng. : Activo]	[Enten : Accuva]	[Plandi: Activa]	[Banus : Active]
	[Log ::Fanished without errors - Romning Time : 47,3 sec.]	[Log : Finished without errors - Running Time : 45,4 sec.]	[Log : Finnibed without errors - Ranning Time : 46,8 sec.]	[Log : Finisheds without errors - Running Time : 46,6 sec.]	[Log : ]
07:15	- Job : Cambe -> Mail : Intégraté des données Comète [Pland: - Tout iss Jours de la semante à Thi 5] [Args : . offerweig giern fa normang/giern fa chapterg@senis.fr,sheurg@senis.fr,%YYYY%MD.0] [Status : Active] [Log : Finished without errors - Ranzing Time : 37,2 sec.]	- Job : Comète -> Mail : Imègnié des données Comète [Pland: - Toor les jours de la senance à Thi 5] [Agns : sfaveregieurs fix geomographieurs fix, desprez gients fix, lesur gierts fix, "NYYYYMMM [Dans: Active] [Log : Finished without errors - Ranning Time : 38,4 sec.]	- Job : Comère -> Muit : Intégraié des données Comète [Pland: Toor les jours de la semme à Thi 5] [Plang: : sittevee]géne fin genomegéne fin cleapyer@senis finleeur@senis fin'6vYYYY%MMG [Plang: : Finished without errors -Running Time : 37,3 sec.]	- Job - Comite -> Mail : Insignië des données Comète [Pland:	- Job : Comète -> Mail : Insègnié des données Comé [Planif. : Tout les jours de la semaine à 7h15] [Arga : infavere@seris.fr,aromme@seris.fr,tlequyet [Stama : Active] [Log :]
07:30	- Job : BD5->BD7 : Extraction des FAE + Avoirs	- Job : EDG->ED7 : Entraction des FAE + Avoirs	- Job : EDG->ED7 : Extraction des FAE + Avoirs	- fob : BDd->BD7 : Extraction des FAE = Avoirs	- Job : BD6->BD7 : Extraction des FAE + Aveirs
	[Planit : Du hundi su vendredi à 7h20]	[Planit : Du head au vendredi à 'hilo]	(Planit : Du brait au ventholt à "ha0)	[Flani: Du hudi su vendedi à 7h30]	[Planif : Du bundi au vendredi à 7h30]
	[Args : 9sYYYY9h0M <im]< td=""><td>[Args : 59/YYY9MDN<im]< td=""><td>(Arga : %9YYYY%M2M<im)< td=""><td>[Args: ?9VYYY96MD41M]</td><td>[Args. : %VYYY%MM4&lt;1M]</td></im)<></td></im]<></td></im]<>	[Args : 59/YYY9MDN <im]< td=""><td>(Arga : %9YYYY%M2M<im)< td=""><td>[Args: ?9VYYY96MD41M]</td><td>[Args. : %VYYY%MM4&lt;1M]</td></im)<></td></im]<>	(Arga : %9YYYY%M2M <im)< td=""><td>[Args: ?9VYYY96MD41M]</td><td>[Args. : %VYYY%MM4&lt;1M]</td></im)<>	[Args: ?9VYYY96MD41M]	[Args. : %VYYY%MM4<1M]
	[Simu: Active]	[Stama: : Active)	(Etmu : Active)	[Estau: Active]	[Stanu : Active]
28:05	- Job : BEU -> COMETE : CENI Emperi Clienti di Sins [Pland: . Touri de la senziane, dangos beury] (Aga: . grod) [Stans: Activit] [Log : Famileed without errors - Ranzing Time : 26,3 soc.] [Log : Famileed without errors - Ranzing Time : 26,3 soc.]	- Job: BLEU >- COMETE: CRM Export Cleants & Sins [Planif: Jours do is sensition, charges heart] [Ages: provide [Status: Activity] [Status: Activity] [Bag: Finished without servers - Running Time : 9,2 sec.]	- Job: EEU-> COMETE: CRN Export Cleans & Sites [Fland: Jours de la constant, chaque berrd] [Aga: _prod] [Status Activit] [Log - Frainbed without errors - Ranning Time : 9,7 sec.]	- Job: EEU -> COMETE: CENA Expert Cleans & Sites [Pland: Jours de la escatar, chaque heure] [Aga:, prod] [Status: Activity] [Log: Franched without error: - Running Time : 9 sec.]	- Job : BLEU -> COMETE : CRM Expert Clients & [Flantif. Jours de la semaine, chaque heure] [Args : prod] [Status : Active] [Log : ]
10.00	-bb : COMETE → Mouste : Mise a jour de la table de planning	<ul> <li>- bob : COMETE → Mousic : Mine a jour de la table de planning</li></ul>	<ul> <li>- bob : COMETE → Mousit : Mine a jour de la table de planning</li></ul>	- bob : COOAETE → Mousaic : Mise a jour de la table de planning	- Job : COLUETE → Mosaic : Mine a jour de la table
	[Pland. Toven hourse de fis 200]	[Plant]. Tourse heures de fui à 10h]	[Pland: Tootes heures de fai 20b]	[Plandf. Toutes heures de fa 200b]	[Flauif. : Toutes heures de dh à 20h]
	[Arga : rest; Cap]	[Ann: resuccidant]	[Aga: :reactApt]	(Arga: :resc.7adp]	[Args. : rou;;Cagt]
	[Batas : Activa]	[Bansa: Activa]	[Patan: Activa]	[Plandf. Activa]	[Stanis.: Active]
	[Dag : Fanished without errors - Ranning Time : 45,5 sec.]	[Log: Finished without errors - Ranning Turse: 48 sec.]	[Log : Finalded without errors - Ramaing Tame : 47,4 sec.]	[Log: Finisheds without errors - Romaing Time : 46,7 sec.]	[Log :]
08:10	- 50: MOSAIC -> COMETE : Export Salariés, Centrats, Avenants	- Job : MOSAIC -> COMETE : Enport Salariés, Contrats, Avenants	- Job : MOSAIC -> COMETE : Enport Salariés, Comms, Avenants	- Job : NOSAIC -> COMETE : Export Salariés, Contrats, Avenants	<ul> <li>Job : MOSAIC -&gt; COMETE : Export Salariés, Car</li></ul>
	[Pland: -] Joon de la senaine, chaque hears]	[Plant] Jours de la seatine, chaque heure]	[Pland: Jours de la seatine, chaque heure]	[Pland: Jours de la senaine, chaque heure]	[Flanif. : Jours de la semaine, chaque heure]
	[Arga: :ess.(24)	[Ang. : rest.;2nd]	[Ang: :resc.[Ang]	[Arga: :resc.(apt]	[Args.: ress;Clag()
	[Stans: Active]	[Status Active]	[Status: Activa]	[Status: Activa]	[Stanus : Active]
	[Log: Finishes without errors - Running Time : 1,2 min.]	[Bug : Finished without errors - Running Time : 1,0 min.]	[Log: :Finitabed without errors - Running Time : 1,2 min.]	[Log : Finisheds without errors - Running Time : 1,1 min.]	[Log :]
08:20	- lob: ELEU >> COMETE : CRM Export Clients & Sites	- Job : BLEU >> COMETE : CRM Export Cleants & Sites	- Job : ELEU >> COMETE : CEM Export Cleants & Sites	- Job : ELEU -> COMETE : CRM Export Cleans & Stres	- Job : BLEU -> COMETE : CRM Expert Clients &
	[Pland: Jours de la semainie, chaque heure]	[Plass]. Fours de la semainte, chaque heure]	[Final: Jours de la estatistic, chaque heure]	[Pland: Joars de la sensinie, chaque herne]	[Flanif: : Jours de la semaine, chaque heure]
	[Aga: , proid	[Atam. pred]	[Atap:: proof]	[Aga: : pto0]	[Args : prod]
	[Sama: , Active]	[States: Active]	[Status: Activa]	[Status : Activit]	[Status : Active]
	[Log: [Finished without errors - Ramning Time : 9,2 sec.]	[Rog: Finished without errors - Running Time : 7,2 sec.]	[Rog: Finaliskd without errors - Ranning Time : 7,5 sec.]	[Log : Finished without errors - Running Time : 7 sec.]	[Log :]
08:35	- be: ELEU ~ COMETE - CRM Expert Clients & Sites	- Job: ELEU >> COMETE: CEM Export Cleants & Sites	- Job : ELEU ~ COMETE : CEM Export Cleants & Sites	- Job : ELEU ~ COMETE : CRM Export Cleants & Sites	- Job BLEU -> COMETE - CRM Expert Clients &
	[Pland: Jours de la semanie, chaque herne]	[Flass]. Jours de la semainte, chaque heure]	[Fland: Joars de la estainte, chaque heure]	[Fland: Jours de la semante, chaque beure]	[Flanif, - Jours de la semaine, chaque heure]
	[Arga: prob]	[Arga: prod]	[Args: proof]	[Agas: proof	[Args:-prod]
	[Status - Active]	[States: Active]	[Status: Acatwa]	[Status: Activa]	[Status - Active]
	[Log: Finalided without errors - Running Time : 7,4 sec.]	[Log: Finanded without errors - Running Time : 9,3 sec.]	[Log : Finished without errors - Ramsing Time : 9,3 sec.]	[Log : Finisheds without errors - Running Time : 9,3 sec.]	[Log - ]
38:45	- be: ALL → BD7. Synchronismin du Data Warshouse	- Job : ALL → ED7 : Synchronismin du Dau Warshouse	- Job : ALL → ED <sup>+</sup> : Synchronismicn du Data Warshouse	- Job : ALL → BD7 : Synchronisation du Data Warshouse	- Job : ALL -> BD7 : Synchronisation ds Data Ware
	[Pland: Prod. Do lundi su vendruhi chaque 2h]	[Plani: Prod : Du lumă au vundrută chaque Zh]	[Pland: Pred - Du lundi au vundretti chaque 2b]	[Pland: Prod. Do lundi au vendredt chaque Ib)	[Flanif : Frod : Du lundi au vendredi chaque Ih]
	[Agn: SWT17PAMACHA[4]]	[Aga:5V1771/NADACHAL[4]]	[Agn: : SVTTPNNANCHA[4]	[Aga: :WYCTPNAMCHAL[4]]	[Args : 94YYYY9MMM<[M][4]]
	[Batas: Active]	[Bana: Activo]	[Status: Activa]	[Starm: Active]	[Stams : Active]
	[Log: Finalised without errors - Running Time: 2,3 min.]	[Log: Finaished wuhout errors - Ramaing Time : 2,3 min.]	[Log: Finished without errors - Raming Time : 2,3 min.]	[Log : Finishek without errors - Running Time : 2,3 min.]	[Log : ]
	<ul> <li>Job : BLEU -&gt; COMETE : CRM Export Clients &amp; Sites</li></ul>	- Job : BLEU -> COMETE : CRM Export Chests & Sites	- Job - BLEU -> COMETE - CRM Export Chents & Sites	- Job : BLEU -> COMETE : CRM Export Clients & Sites	- Job : BLEU -> COMETE : CRM Export Clients &
	[Planif: : Journ de la semaine, chaque heure]	[Planif: : Jours ds la semains, chaque houre]	[Planif: : Jours de la semaine, chaque heure]	[Planif: : Jours de la samaine, chaque beure]	[Flanif: : Jours de la semaine, chaque heure]

# Source tab

Here you choose the Source connection. In the case of a database, you can choose to view the list of available BDDs and use one that is different from the one from the connection string.

Fuzible - Data Replic	ation & Synchronization			- 🗆 X
File Configuration To	ols Help			
-Job Selection			Create/manage a Multi-steps Jo	b
GUIZM	[34] ALL SCENARIO JOB	~		Create new Step
Job Configuration S	ource : SQLite Database Target : Exce	I File Queries Log View	er	
Choose a Source	[5] -> Local SQLite File	✓ Edit Conne	ection	
Databases	~	Try Conne	ction and Get Databases	
CData Analyzer		CData Transformation		
Smart Data Analy	zer	0 - No Transformation	$\checkmark$	
Data Scanning Level	Analyze each row 🗸			?
Source Driver	~		Parallel Queries Executi	on 1 ?
Driver Parameters —				
O DataReader Mode	(slower but uses less RAM. Can also avoid	errors with some ODBC drivers	5)	
Remove data from	Source after having been inserted in Targe	et		
- Will only be executed	for queries with 1 table and will be bypassed in cas	se of errors during Job execution.		
Pre/Post Job Command	l(s) : any SQL raw command (ex : SELECT M	IAX(field) FROM table)		
Perform Post-Job Con	nmands 🗸			?
Save Job	(II) Abo	ort Job	Simulation Mode	(D) Start Job

Try Connection	A quick connexion check
	- BDD: checks the connection to the SQL instance and brings back the databases list
	- FILE: checks the existence of the path (or (s)ftp) and displays the list of available files
	- WEBSERVICE/MAIL: pings the server
	- ACTIVE DIRECTORY: checks AD domain availability
Parallel Queries Execution	If the Job has multiple queries, you can choose to run multiple queries in parallel.
	Beware of the resource consumption associated with these simultaneous executions.
Intelligent Data Analyzer	This is the data analysis engine, thanks to it, for example, Fuzible can, among other things, to automatically create SQL tables that do not exist with the most accurate data types, resulting from data analysis.
	If you're dealing with large datasets, the data analysis engine can be resource-consuming, and it's not always useful to analyze
	all the data. You can set either scan all the rows or all the X rows.
	→ See "Data Analyzer"
Data Transformation	Fuzible can transform source data, like a PIVOT operation would do.
	→ See "Data Transformation"

BDD					
	t uses less RAM. Can also avoid errors with some ODBC drivers)				
	Direct Stream Copy (parallelized read & write data stream)				
Remove data from Source aft	er having been inserted in Target				
Ũ	ith 1 table and will be bypassed in case of errors during Job execution.				
DataReader Mode	This is an alternate way of getting Source data. It can be useful with ODBC drivers which sometime It's also less RAM consuming.	es uses buggy drivers.			
Direct Stream Copy In DataReader mode, data is read row-by-row, it means that they can be transferred to Target by chunks (100 rows by default, can be set in program configuration / SQL tab). That method is useful as it parallelizes read and write operation consumes less RAM, and most of the times, offers great performances. The downside is that in case of your Job's Target is a database, and Fuzible is asked to automatically create the target t working with small data chunks may prevent the engine to create an accurate data schema. It is advised to use that feat the Target Table is already created.		d and write operations, Ily create the target table,			
Remove Data from Source	If you simply want to transfer Source data to the Target, this option allows you to delete the data	that has been retrieved			
after having been inserted in Targetfrom the Source. If, however, the Job contains errors during execution, this step will be avoided.TargetIn addition, if the Source query contains multiple tables (joins), it will also be avoided.					

MONGODB Driver Parameters	
Remove internal MongoDB ID colored in the second	ımn from retrieved data
Remove internal MongoDB ID column from retrieved data	Each MongoDB collection identifies its records with an ID. This is shown as an additional column when retrieving data. This option allows you not to get this column

CSV file			
Driver Parameters			
Are the files to be processed zipped? (Enter the Zi	P filename) Browse Post-process (Source Files) Nothing V		
Raw Output (won't create a dataset, v	will only extract data in a single column)		
Row Offset - Read files starting at row	w : 0		
Read Multiple Files At Once - Name	has to contain : ^ ?		
Source File(s) are zipped in	If the files you want to query are in a ZIP file, the filename is defined here (its name may be or contain dynamic		
	parameters)		
Post-Process	You choose what you want to do with source files once processed:		
	1 - Nothing: We leave them where they are.		
2 - Move: They are moved in a sub-directory.			
	3 - Zip: Compact them into a ZIP file.		
4- Delete: they are removed.			
	These 4 options also apply in the case of network path and (S)FTP		
Raw Output	Instead of creating datatable from the files content, it is simply extracted "raw", as shown in Notepad (for example)		
Row offset	Tells Fuzible from which row he should start to read the file.		
	→ See term "header detection"		
Read multiple Files at once	Allows you to indicate a "pattern" for the file name: Fuzible will then get and merge all data from all files with a name		
matching that pattern			

Errol Ch			
Excel file			
Driver Parameters			
Are the files to be processed zipped? (Enter the Z	P filename) Browse Post-process (Source Files) Nothing V		
Raw Output (won't create a dataset,	will only extract data in a single column)		
Sheet to Read (0=all)	1		
Row Offset - Read files starting at ro	w: 0		
Password			
	i fuls fils a suite suite suite suite fils and fils de fils and fils fils and the fils and the suite suite suite		
Source File(s) are zipped in	If the files you want to query are in a ZIP file, the filename is defined here (its name may be or contain dynamic parameters)		
Post-Process	You choose what you want to do with source files once processed:		
	1 - Nothing: We leave them where they are.		
	<ol> <li>Move: They are moved in a sub-directory.</li> <li>Zip: Compact them into a ZIP file.</li> </ol>		
	4- Delete: they are removed.		
	These 4 options also apply in the case of network path and (S)FTP		
Raw Output	Instead of creating datatable from the files content, it is simply extracted "raw", as shown in Notepad (for example)		
Row offset	Tells Fuzible from which row he should start to read the file.		
	→ See term "header detection"		
Sheet to read	Tells Fuzible the sheet index data will be retrieved from		
Password	If the Excel file is password protected, this is where it should be indicated. On the other hand, if you try to write multiple queries using multiple Excel files, which don't all have the same password, you'll be forced to create multiple Jobs.		
VML on d ICON Elles			
XML and JSON Files			
Source Driver	Parallel Queries Execution 3		
XML File 🗸	Parallel Queries Execution 3		
CDriver Parameters			
Are the files to be processed zipped? (Enter the ZI	P filename) Browse Post-process (Source Files) Nothing V		
Raw Output (will keep data as it was	retrieved)		
Source File(s) are zipped in	If the files you want to query are in a ZIP file, the filename is defined here (its name may be or contain dynamic		
Deet Deeree	parameters)		
Post-Process	You choose what you want to do with source files once processed: 1 - Nothing: We leave them where they are.		
	2 - Move: They are moved in a sub-directory.		
	3 - Zip: Compact them into a ZIP file.		
	4- Delete: they are removed.		
	These 4 options also apply in the case of network path and (S)FTP		
Raw Output	Instead of creating datatable from the files content, it is simply extracted "raw", as shown in Notepad (for example)		

# Note on head detection

By default (« Raw Output » unchecked), Fuzible analyzes the contents of CSV and EXCEL files on its own and automatically determinates the presence of a header. Its analysis is based on a set of tests that make it reliable in 99% of cases.

Webservice R	EST				
Raw Output (won't o	Raw Output (won't create a dataset, will only extract data in a single column)				
SQL Language	Fuzible SQL	✓			
Query Method	GET ~				
Body Sent As	Raw - JSON ~				
API Endpoint (Nuxed	o only)				
SQL Language		Some APIs can use their own simili-SQL language that can be used instead of "Fuzible SQL", which is the default engine. For example, the Salesforce CRM uses SoQL			

Method	Data retrieval method: GET or POST	
Body Sent As	You can send some body content in your queries to an API. Here you specify what kind of body it is: Form-Data, JSON, XML	
Raw output	The responses returned by the API will be extracted « raw », the content won't be serialized as data table(s).	

Note that depending on the connection "template" you choose (connection configuration), you may have a slightly different setting; for example, if you use the Salesforce SoQL "template", the settings are automatically set for you:

-Source Driver	~	Parallel Queries Execution	1	?
- Driver Parameters				
SQL Language	SoQL (Salesforce)			
🔘 Raw Output (will k	eep data as it was retrieved)			

Mailbox Driver Parameters Get Unread Messages	Only (IMAP)	
Post-Operation Limit to retrieve	Nothing 100	~
Get unread messages of	only	Chooses to retrieve only emails that have not yet been read (IMAP only)
Post-Operation		Defines what you do with the email after Fuzible read it.         -       Leave it as it is.         -       Move it to a "Fuzible" folder         -       Delete it.         -       Mark it "as read".         Note: The POP protocol only supports deletion.
Limit to retrieve		By default, the program recovers all the mails from the inbox, which can take an extremely long time. It is best to set a limit here.

Active Directory	,
Search Scope Ba	se v
Search scope	Search depth for AD objects (base, one level, all levels)

# Data Analyzer

Intelligent Data Analyzer			2
Data Scanning Level	Analyze each row	$\sim$	•
	Analyze each row		
	Analyze every 5 rows		
Advanced Parameters	Analyze every 10 rows		
Advanced Farameters	Analyze every 20 rows		

Fuzible may need to analyze the data it collects to determinate the most accurate type. This feature is particularly useful in 2 cases:

- The target is a database and the Target table do not exist or has an inaccurate schema: Fuzible can either create them or "improve" them if requested (change the field type(s))
- Sync mode: To best compare Target and Source data that sometimes come from quite different sources, the analysis helps to better translate a value.
- Filtering and aggregating data from Sources other than a database

To date, Fuzible is capable of creating and/or transcoding the most standard fields: CHAR, NVARCHAR, VARCHAR, TEXT, DATE, DATETIME, TIMESTAMP, INTEGER, DECIMAL (X,X), FLOAT, BIGINT, SMALLINT, BIT. Any other type of field will be seen as a "TEXT" field.

The lower the accuracy requested by the analyzer (scan all X lines), the more it will weight its analysis to ensure that the import is done correctly.

Example: A field identified VARCHAR(5) will be created in VARCHAR(10) in the target if the scan mode scans only all 50 lines)

Note: For all queries retrieving less than 100,000 rows, the analyzer scans all rows, regardless of the chosen setting.

### **Data Transformation**

The collected data can be completely transformed before its integration into the Target:

0 - No Transformation	$\sim$
0 - No Transformation	
1 - Hyperfile Arrays to Rows	
2 - Pivot By Common Root(s)	
3 - Switch Rows and Columns	

# Hyperfile Arrays Transformation:

Data Transformation	1 - HyperFile Arrays To Rows 🗸 _		
Avoid Transformation if Variable Arrays Sizes are Detected			

This is a kind of transformation specific to Hyperfile databases: some fields of this SGBD are in fact arrays, and when these arrays are retrieved by the ODBC driver, it produces as many columns as there are columns in these fields of "array" type (example: PPXRUBP\_01, PPXRUBP\_02, PPXRUBP\_03...) Transformation automatically analyzes and counts all columns of this type (they always have the name of the field, followed by \_XX) and flips them so that there is only one, and the data is thus transformed into rows and no longer columns.

Example: If you have na array-type column of 50 entried in HYPERFILE, the ODBC sends you 50 different fields! The transformation engine retains only one to favor a better display of this data (i.e. 50 rows for one originally)

**Example:** on the right, the initial data source, on the left, the result produced by the transformation:

l	id_sample;data_01;data_02;data_03;data_04;li_sample
2	l;hellol;hello2;hello3;hello4; <u>firstrow</u>
3	2;worldl;world2;world3;world4; <u>secondrow</u>
4	3;wolfl;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

As can be seen, the fields data\_01, data\_02,... follow a "label + number" logic, which allows you to take "data" as a label, and to display the index in "IDX\_COL"

# **Pivot by Common Root:**

Data Transformation	2 - Pivot By Common Root(s)	$\sim$	-	
---------------------	-----------------------------	--------	---	--

Avoid Transformation if Variable Arrays Sizes are Detected

This kind of transformation reverses source data according to a common root field name. For example, if you put "x," the program will flip all the fields that start with "x."

To this end, the source will be increased by 3 fields:

- An "x" field, containing the value.
- A "x\_lbl" field, containing the rest of the fields name (ex: « x\_001 » -> 001)
- A "x\_idx" field containing the row index

If 7 fields with the "x" root have been detected, then 7 rows will be produced. These fields will of course be removed from the source and replaced by the 3 fields shown above.

**Example:** on the right, the initial source data, in the middle, the query, on the left, the result produced by the transformation:

Dat	ta Transformation	2 - Pivot By Common Root(s)	$\sim$	split
1 2 3 4 5	id_sample;dataW;dataX;dat l;hellol;hello2;hello3;ha 2;world1;world2;world3;wu 3;wolf1;wolf2;wolf3;wolf4	ello4;firstrow orld4;secondrow		TRANSFORMED.CSV:select id_sample, li_samp dataW <i>as</i> split01, dataX <i>as</i> split02, dataY <i>as</i> split03, dataZ <i>as</i> 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

We see that the fields "W,X,Y,Z" were deliberately renamed with a common root in the query.

- "split": the root name
- "split\_lbl": the data that was contained in the column
- -"split\_idx": the original column index

## Switch Rows and Columns:

Data Transformation	3 - Switch Rows And Columns 🗸
	Add a Column With Label (PROPERTIES)

Simply flips the columns into rows and vice versa.

The "Add a column with label (PROPERTIES) option allows you to add a column with the original name of the original column associated with the reversed data.

**Example:** on the right, the initial source, in the middle, the query, on the left, the result produced by the transformation:



# If Any, also transform Cross-Queries

If a Query is built with Cross-Queries (data coming from other Sources), the Transformation will only be executed on the Main Query, then all Data coming from Cross-Queries won't be transformed but merged with the already transformed Data from the Main Query. You can choose to Transform any Dataset that is collected through all Cross-Queries which means that Cross-Query behavior will be executed on full transformed datasets. Not only the main one.

CData Transformation	
2 - Pivot By Common Root(s) 🖌	_
Avoid Transformation if Variable Arrays are Detected	?
If Any, also Transform Cross-Queries	

# **Pre/Post-Job Commands**

Accessible from the "Source" tab and the "Target" tab, for "Database" or "File" connections. This feature is an answer for two identified needs:

- Sometimes a simple data copy Job is not enough, you need to launch something before or after this copy, and you want to avoid having to program these scenarios in an external Orchestration software.
- You want to retrieve some data before starting the Job to exploit it (conditioning the behavior of the job according to this data, for example)

This feature allows you to launch one or more commands before or after the Job is executed.

ne de la constante de la const				
Pre/Post Job Command(s) : any CME	raw command (ex : DATE /T)			
Perform Post-Job Commands 🗸 🗸	DATE /T	?		

Any command you could launch from the Windows shell is supported. If multiple commands are to be launched, they must be separated by a ";"

Ex : c:\Tools\mycommand.bat

Ex2 : DATE /T (returns the actual date)

Fuzible adds any returned value/message/error from those commands into its LOG.

### Database

....

Pre/Post Job Command(s) : any SQL	raw command (ex : SELECT MAX(field) FROM table)	
Perform Post-Job Commands 🗸 🗸	SELECT LAST(dt_import) FROM MY_IMPORTS;	?

Any SQL command compatible with the selected SGBD. The execution of a stored procedure, an UPDATE... If multiple commands are to be launched, they must be separated by a ";"

Ex : UPDATE myTable SET sent = 1 WHERE month = {%MM}

```
Ex2 : EXECUTE myProcedure('1')
```

In example 1, you see, as a reminder, that a dynamic parameter of the Job can be used.

These commands can return a value. Fuzible can exploit these values as dynamic parameters. For example, if I write in the dynamic settings of Job %CS1, it means that this dynamic setting will be replaced by the first value of the first command of the "Source" connection:

- C Command
- S Source
- 1 Command No.1

We can also write %CT1 (Command Target No. 1) or %CT2...

# **Looped Pre-Job Commands**

You may notice that if only one Pre-Job command has been set, an option appears. "Loop Job for each Result ».

This option allows you to make the Job scenario more complex by associating a dynamic parameter to the result of a command, which, if it returns several rows, allows you to loop the Job as many times as there are results, assigning a different dynamic parameter each time.

#### **Explanations :**

You call a stored procedure "getMails" (CALL is the MySQL syntax for calling a stored procedure)

Pre/Post Job Command(s) : any SQL raw command or Stored Proc. (ex : SELECT MAX(field) FROM table)					
Perform Pre-Job Commands 🛛 🗸	CALL geMails;	•	Loop Job for each result	?	

#### ... which returns this data set:

mailField	IdPeople	Comment
Leon@mymail.com	1	Our beautiful CEO
Arthur@mymail.com	2	Our incredible COO
Samantha@mymail.com	3	Our amazing CTO

You might want to send an email to each of these people, with an email containing, for example, their information summary.

This option makes it possible to carry out this scenario, provided that you program the Job accordingly.

In this case, one or more dynamic parameters must be assigned to the results produced by the call to the stored procedure (or any other command returning a set of results).

For example, a single variable can be associated in the following way ("Job Configuration" Tab):

**Dynamic Parameters** 

%SC1<mark>;</mark>

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}...

### Or more than one, by specifying the column number (in base 1) to which the parameter is associated:

# 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}... You can then set a Query which, for each mail returned by the stored procedure, will send a mail to the person in question, with his personal information:

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}** -> Will be replaced by the dynamic parameter N°1, which is filled with the data of the first column of the stored procedure, i.e. the person's email.

**{ ?2}** -> Will be replaced by the dynamic parameter N°2, which is filled with the data of the second column of the stored procedure, i.e. the ID of the person.

Thus, the Job will run in a loop until the result set of the stored procedure called in Pre-Job has been consumed.

In our example, the Job will run 3 times, with the following values

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

With this option, it is therefore possible to make your job a little more scenic and dynamic.

Restriction: Only one pre-job command can be entered for this option to be available

Furthermore, if it has been activated on the "Source" tab, it cannot be activated on the "Target" tab (and vice versa), even if a pre-Job command has been entered. This is to avoid making the Job too confusing and to avoid scenarios that require a more visual presentation of the Job's behavior (in the form of a diagram, for example)

# Target tab

Here you choose the Target connection. In the case of a database, you can choose to view the list of available databases and use one that is different from the one in the connection string.

b Selection ———				Create/manage a Multi-step	s Job ———
UIZM	[65] BIG Data	Perf Tests	~		Create new Ste
b Configuration Sc	ource : CSV File	Target : SQLite Database Queries	Log Viewer		
Choose a Target	[17] -> Loca	al SQLite File BIG 🗸 🗸	Edit Conne	ection	
Databases		~	Try Connec	tion and Get Databases	
rget Driver ———— SQLite	✓ ● Ti	rim Data (Left, Right)		Parallel	linsertion 1
-	✓ ● Ti	rim Data (Left, Right)		Parallel	l insertion 1
SQLite		rim Data (Left, Right) p + Recreate	~	Parallel	[•]
SQLite	Dro		yze up to 5 fi	If non-existent, create table	[•]
SQLite iver Parameters	Dro s to be created, t	p + Recreate		If non-existent, create table ields - may take a lot of time)	[•]
SQLite iver Parameters Target table behavior If Target Table ha Allow Schema Alt	Dro s to be created, t teration in Target	p + Recreate try to find and add a Primary Key ? (can ana	a VARCHAR	<ul> <li>If non-existent, create table</li> <li>ields - may take a lot of time)</li> <li>(5) column)</li> </ul>	[•]
SQLite iver Parameters Target table behavior If Target Table ha Allow Schema Alt	Dro s to be created, t ceration in Target	p + Recreate try to find and add a Primary Key ? (can ana : (ie : when trying to insert a 10-char value in g/updating/deleting data (faster but require	a VARCHAR	<ul> <li>If non-existent, create table</li> <li>ields - may take a lot of time)</li> <li>(5) column)</li> </ul>	[•]
SQLite iver Parameters Target table behavior If Target Table ha Allow Schema Alt Disable Constrain Insert NULL inste	Dro s to be created, t ceration in Target its when inserting ad of empty value	p + Recreate try to find and add a Primary Key ? (can ana : (ie : when trying to insert a 10-char value in g/updating/deleting data (faster but require	a VARCHAR	<ul> <li>If non-existent, create table</li> <li>ields - may take a lot of time)</li> <li>(5) column)</li> </ul>	[•]
SQLite iver Parameters Target table behavior If Target Table ha Allow Schema Alt Disable Constrain Insert NULL inste	Dro s to be created, t teration in Target ats when inserting ad of empty valu (s) : any SQL raw	p + Recreate try to find and add a Primary Key ? (can ana : (ie : when trying to insert a 10-char value in g/updating/deleting data (faster but require	a VARCHAR	<ul> <li>If non-existent, create table</li> <li>ields - may take a lot of time)</li> <li>(5) column)</li> </ul>	[•]

# **Common settings to all targets:**

Additional Columns			
Add a Row Count Column in Target :		Add a Column with Source DB/Pat	h 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}

# All of the following special columns use a default name that can be changed by the user.

Add a row count column in target	Adds a "ROWNUM" column to the data retrieved from the source, which is simply a row counter			
Add a timestamp column in target	Adds a "DTLOAD" column to the data retrieved from the source that contains the data retrieval date			
Add a column with source database/path	Adds a "DBNAME" column to the data retrieved from the source that contains the source of that data			
Add a column with a dynamic param	Adds one or more optional columns to the data retrieved from the Source (ex:, MYCOLUMN={ ?1} : will add a			
	"MYCOLUMN" column and fill in its data with the dynamic setting n°1			
	It is possible to add several columns by separating them like this:			
	MYCOLUMN1={ ?1} ;MYCOLUMN2='test'			

Postgres	``	/	Trim Data (Left, Right)	Parallel insertion	]
Trim data			Removes any whitespace befor	e and after a string	
Parallel Insertic	on			ET = Database" mode: It allows you to perfo juires a high-performance Target database, execution (Source tab)!	

Database			
CDriver Parameters			
Target table behavior	Truncate	Y If non-evistent create table(s) automatically	
Target table behavior	Tuncate	If non-existent, create table(s) automatically	
<ul> <li>Bulk Insert (very fast, but less</li> </ul>	reliable if Source Data needs some	e transcoding to be properly inserted in Target)	
If Target Table has to be created and the c	ted try to find and add a Primary K	ey ? (can analyze up to 5 fields - may take a lot of time)	
Allow Schema Alteration in I	arget (ie : when trying to insert a 10	-char value in a VARCHAR(5) column)	
<ul> <li>Disable Constraints when ins</li> </ul>	erting/updating/deleting data (faste	er but requires sufficient privileges)	
<ul> <li>Insert NULL instead of empty</li> </ul>	values		
			J
Target Tables Behavior	Available in "Data Replica	tion" mode: Defines what to do on the target table when you fill it out.	
	Target Tables Behavior	Truncate v	
		Drop + Recreate	
	Allow Add+Change Type	Truncate a	
	<ul> <li>Disable Constraints (requi</li> </ul>	ires Full Delete	
		Partial Delete (using Query (Where Condition)	
	Set NULL for Empty Value	Partial Delete (using Dynamic Param Column(	
	Set NULL for Empty Value	Nothing	
	1 - Drop - Recreate: delet	e the destination table and then rebuild it.	
	2 - Truncate: delete all da	ata from the table using a "TRUNCATE" statement.	
	3 - Full Delete: delete all	data from the table using a "DELETE" statement.	
	0	iery (Where): will use the "WHERE" filter(s) from the source query to remo	ve data with the
	same filter in the destinat		
		FROM MYSOURCE WHERE id > 50	
		nove all "id > 50" in "MYTARGET" before inserting new data namic Param Column: Will use the dynamic column as a filter	
	• ,	al column MYCOLUMN={ ?1} with { ?1} using 'TEST' as a value, the DELETE I	FROM
		YCOLUMN = 'TEST' statement will be performed before inserting new data	
	, .	be done before inserting new data.	
Bulk Insert		opying data into the Target Database. While it offers amazing performance	es, it is sometimes
	less reliable than tradition	al transactional SQL, especially when data needs to be converted on the f	ly between Source
	and Target.		
If non-existent, create table(s)	By default, if the Target ta	ble does not exists, Fuzible will automatically create it on-the-fly with the	most accurate
automatically	data schema. You can byp	ass this behavior. If the Target table does not exists, an error will be insert	ed in the LOG
Allow add+change type columns i		ts data analysis engine, to modify the target table schema if necessary (cha	-
target		nt privileges on the target database. This is especially useful if Source data	is often changing
	and Target table needs to	, , ,	
Disable constraints		rformed by disabling foreign key constraints.	
Truto add primaru kov		eges on the target database.	ible combinations
Try to add primary key	-	It have a primary key, Fuzible can create it on its own by analyzing all possi nalysis is limited to a maximum of E fields. If the number of fields and date	
		nalysis is limited to a maximum of 5 fields. If the number of fields and data time and it is not advisable to use this feature.	ns nuge, the scan
		rivileges on the target database.	
Set NULL for empty values		urce can be replaced as a "NULL" value in the Target data	
Set itole for empty values	in chipty data in the 50	aree can be replaced us a more value in the ranger data	

CSV file	
CSV IIIe	
Driver Parameters	
File Creation Behavior	
Rows / created file : 1000000 H	elp for scripting multiple output filenames ? O If exists, append output file(s)
CSV separator ;	
Add header row (using Source query field	s -or aliases- names)
-	
Embrace values with double-quotes (ie : "	Value1 ; Value2 ; Value3 )
Rows/ File	How many rows you want to copy into a single file? If the number of rows in the Source data exceeds this value, a "pattern" must be set to name the files that are going
	to be created.
	See "Multiple files naming pattern"
Append Existing File	If the target file already has rows, you can decide not to overwrite it but add data into it.
CSV separator	Sets the separator character of the target file
Add header row	The header is built using Source field names
Embrace values with double quotes	Double quotes will be added before and after the value (ex : "test";"125";"hello")
Encel Cla	
Excel file	
Driver Parameters	
File Creation Behavior	
Rows / created file : 1000000	Help for scripting multiple output filenames ? O If exists, append output file(s)
Set password	Visual Style : None ~
Add header row (using Source query fi	elds -or aliases- names)
Add a Title Row (using 'Job Description')	))
Rows/ File	How many rows you want to copy into a single file?
	If the number of rows in the Source data exceeds this value, a "pattern" must be set to name the files that are going
	to be created.
	→ See "Multiple files naming pattern"
Append Existing File	If the target file already has rows, you can decide not to overwrite it but add data into it.
Append Existing File Set password	
	If the target file already has rows, you can decide not to overwrite it but add data into it.
Set password	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file
Set password Add header row	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names
Set password Add header row Add a title Row	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description
Set password Add header row Add a title Row	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description
Set password Add header row Add a title Row Style	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description
Set password Add header row Add a title Row Style File XML	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style.
Set password Add header row Add a title Row Style File XML Driver Parameters	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ? O If exists, append output file(s)
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style.
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0'	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ? O If exists, append output file(s)
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0'	If the target file already has rows, you can decide not to overwrite it but add data into it.         Sets a password on the Excel file         The header is built using Source field names         Adds a general head row, the value will be the Job's description         Allows you to pick-up a graphical style.         Help for scripting multiple output filenames         Row Tag script builder         Row         Tag (i.e : <myfield>myvalue         Add CDATA for all values (ie : <![CDATA[<sender>John Smith</sender>]]></myfield>
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields ->	If the target file already has rows, you can decide not to overwrite it but add data into it.         Sets a password on the Excel file         The header is built using Source field names         Adds a general head row, the value will be the Job's description         Allows you to pick-up a graphical style.         Help for scripting multiple output filenames         Row Tag script builder         Row         Tag (i.e : <myfield>myvalue</myfield> )         V         Add CDATA for all values (ie : <sender>John Smith</sender> If the number of rows in the Source data exceeds this value, a "pattern" must be set to name the files that are going
Set password         Add header row         Add a title Row         Style         File XML         Driver Parameters         File Creation Behavior         Rows / created file :         1000000         Header Row :       xml version='1.0'         Write Mode :       Mode 1 : Fields ->         O Don't create Tag for empty values	If the target file already has rows, you can decide not to overwrite it but add data into it.         Sets a password on the Excel file         The header is built using Source field names         Adds a general head row, the value will be the Job's description         Allows you to pick-up a graphical style.         Help for scripting multiple output filenames         Row Tag script builder         Row         Tag (i.e : <myfield>myvalue         Add CDATA for all values (ie : <![CDATA[<sender>John Smith</sender>]]>         If the number of rows in the Source data exceeds this value, a "pattern" must be set to name the files that are going to be created.</myfield>
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ?
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ? O If exists, append output file(s) Row Tag script builder Row ? Tag (i.e : <myfield>myvalue</myfield> )
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ?   If exists, append output file(s) Row Tag script builder   Row ? Tag (i.e : <myfield>myvalue</myfield> )
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ?
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ? If exists, append output file(s) Row Tag script builder Row ? Tag (i.e : <myfield>myvalue</myfield> )
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ?
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ? ○ If exists, append output file(s) Row Tag script builder Row ? Tag (i.e : <myfield>myvalue</myfield> ) v ○ Add CDATA for all values (ie : <i[cdata[<sender>John Smith]]&gt; If the number of rows in the Source data exceeds this value, a "pattern" must be set to name the files that are going to be created. → See "Multiple files naming pattern" If the target file already has rows, you can decide not to overwrite it but add data into it. Choose the XML header (usually: xml version='1.0') Choose the way you want to build the XML schema : → Mode 1 : Each row is written inside a main row tag (Row Tag Script Builder), each field is a sub-tag, and contains the associated value Ex : <row><myfield>myValue</myfield><myfield></myfield></row></i[cdata[<sender>
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ?
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version="1.0" Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header Write Mode	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ? If exists, append output file(s) Row Tag script builder Row Tag (ie: <myfield>myvalue</myfield> )
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version='1.0' Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ?
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version="1.0" Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header Write Mode Add CDATA tag for each value	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ? If exists, append output file(s) Row Tag script builder Row ? Tag (i.e : <myfield>myvalue</myfield> ) Add CDATA for all values (ie : <[[CDATA[-sender>John Smith]]> If the number of rows in the Source data exceeds this value, a "pattern" must be set to name the files that are going to be created. See "Multiple files naming pattern" If the target file already has rows, you can decide not to overwrite it but add data into it. Choose the XML header (usually: xml version='1.0') Choose the way you want to build the XML schema : Mode 1 : Each row is written inside a main row tag (Row Tag Script Builder), each field is a sub-tag, and contains the associated value Ex : <row><myfield>myValue</myfield> Mode 2 : Each row is a row from the Source, each value is an attribute (field name). Ex : <row myfield="myValue" myfield2="">myValue2//s Mode 2 : Each row is a row from the Source, each value is an attribute (field name). Ex : <row myfield="myValue" myfield2="myValue2//s&lt;br">(Mode 1 only) If the source ontains exotic values, the standard tag "CDATA" allows the data to be framed so that an XML interpretation engine understands that the values framed by this tag contain special characters</row></row></row>
Set password Add header row Add a title Row Style File XML Driver Parameters File Creation Behavior Rows / created file : 1000000 Header Row : xml version="1.0" Write Mode : Mode 1 : Fields -> O Don't create Tag for empty values Rows/ File Append Existing File Header Write Mode	If the target file already has rows, you can decide not to overwrite it but add data into it. Sets a password on the Excel file The header is built using Source field names Adds a general head row, the value will be the Job's description Allows you to pick-up a graphical style. Help for scripting multiple output filenames ?

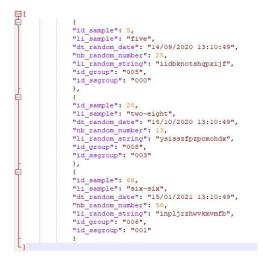
File JSON Driver Parameters CFile Creation Behavior					
Rows / File	10000	Append existing file	Help for multiple files output scripting	?	
Row Tag script builder	[JOBNAME]		?		
Rows/ File		If the number of rows in the So to be created. See "Multiple files of the solution of the sol	ource data exceeds this value, a "pattern" r naming pattern"	nust be so	et to name the files that are going
Append Existing File		If the target file already has ro	ws, you can decide not to overwrite it but a	add data i	into it.
Row Tag script builder		Sets a behavior script for the t → See "Tag Builder"	ag of each row.		

#### **TAG BUILDER**

By default, the structure of an XML and JSON file produced by Fuzible:



#### The JSON file will be as follows:



#### ... and the XML file:



The main tag uses the name of the input table (the first if the query contains joins)

Now, if I want to change the main XML tag, I just need to ask an alias in my query:



"SampleTag" is the alias that will be used in the XML body



#### Supported keywords:

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

I will set the script in the "Target" menu:

ows / File	1000	Append existing file	Help for multiple files output scripting	?
leader xml v	ersion='1.0'		Don't create Tag for empty values	
ow Tag script builder	CurrentID=	id sample]	? Add CDATA Tag for each value	

#### The JSON file creates:



#### ... and the XML file:



#### **MULTIPLE FILES NAMING PATTERN**

When using a File Connection as the Target, you can decide to split the result into several files from a number of rows. For example, if the Source data contains 1000 rows, you can split the result into a single file containing 1000 rows, or 5 files of 200 rows each, or 1000 files of 1 row each.

			<b>.</b>		<i>.</i>
This multi-file	nattern can	he smartly	configured from	n the "Oueries'	' tab (see below)
THIS HIGHLINC	puttern cun	DC Sinutry	configured from	The queries	

[QUERYALIAS]	Alias from the first source table of the query				
	Ex : MYFILE [QUERYALIAS].CSV : SELECT * FROM MYTABLE AS MYQUERY				
	Gives : MYFILE_MYQUERY.CSV				
[FILECOUNT]	File counter: returns the number of files created.				
	If the program has already created 3 files, it will return "4" to the next [FILECOUNT] pattern.				
	Ex : MYFILE_[FILECOUNT].CSV : SELECT * FROM MYTABLE				
	Gives : MYFILE_1.CSV, MYFILE_2.CSV				
[ROWCOUNT]	Row counter: returns the source query row number to the start of the file.				
	If we define a file change every 1000 rows, at the creation of the <sup>2nd</sup> file, the software will return				
	1001 (to the 3rd file, 2001)				
	Ex : MYFILE_[ROWCOUNT].CSV : SELECT * FROM MYTABLE				
	Gives : MYFILE_1.CSV, MYFILE_1001.CSV				
[COLUMN]	Returns the value from a field when the new file is created				
	If the value of the "my_field" field is "Hello" at the time of the creation of the new file, then "World"				
	when the next one is created, the engine will return "Hello" and so on.				
	Ex : MYFILE_[myField].CSV : SELECT myField FROM MYTABLE				
	Gives : MYFILE_Hello.CSV, MYFILE_World.CSV				

#### What you can do:

- Use those keywords in any order.
- Use them multiple times.
- [COLONNE] can be used several times, with several different columns (make sure the column exists in the source, if any, the name of the column is returned and not its value!)
- Intersperse characters between each keyword (ex: [FILECOUNT]-\_-[ROWCOUNT])

#### Restriction:

- The special characters will be automatically replaced with a "\_"

#### Some examples:

Postulate: Source produced 2000 rows and we want to have 1000 rows/file. Filename = TEST, The output file is a CSV.

2 files will be created:

hello[ROWCOUNT]world[NOM_CLIENT].CSV : SELECT	<ul> <li>hello0worldFNAC.CSV</li> <li>hello1001worldAUCHAN.CSV</li> </ul>
Hello[FILECOUNT]_[ID_CLIENT] . <i>CSV : SELECT</i>	- Hello1_283.CSV - Hello2_81036.CSV
[QUERYTARGETNAME][ROWCOUNT][FILECOUNT] .CSV : SELECT	- TEST11.CSV - TEST10012.CSV
[QUERYTARGETNAME]%§£[FILECOUNT] .CSV : SELECT	- TEST1.CSV - TEST2.CSV

Webservice REST/NUXE	0					
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> ) : <pre></pre>						
Add some Source column(s) to server responses (ie : myField1;myField2) : myIDcolumn						
Call method POST v	Source data will be proce	ssed as Build JSON Bo	dy from data		~	
Format URL with upper chars	Oon't send empty values	Source Data of	fset - Process (	data starting at field :	0	
Save HTTP responses in source		enerate answers (XML, s a BDD, in a table, if it	,		and int	egrate into the connection that
Table Log/File Name	(Optional) table name	(or file) that will receiv	e answers f	rom the queries mad	le to th	e API
Track source column(s) in responses						the API, and it is not easy to find
				in define one or mor	e fields	from the source to be kept and
		table/file to track dowr				
String that says success in WS						swers that identifies the call as
Answers Format URL with upper chars	, and a second s			· · · · ·		e a "WARNING" in the LOG t capital-formatted URLs
Don't Send Empty Values		TP query, Fuzible will a		•		t capital-formatted OKES
Call Method		d by the API (POST, PUT	Ŭ		ild	
can wethou		, , ,	, DELETE, T	Arenj		
	Call method	POST				
		POST				
	re/Post Job Comman	PUT	-			
	errost Job Comman	DELETE				
		PATCH				
Content Type						er in the form of JSON or XML
	data in the body, as H	TTP parameters or in "r	aw" mode (	when your source fil	e is a ra	aw JSON file for example)
	Content type	Raw Text	v			
	E	Build HTTP Query from d	ata			
	E	Build JSON Body from da	ita			
	E	Build XML Body from dat	а			
	F	Raw Text				
Columns send offset	A query to an HTTP we	ebservice builds a conca	itened chair	n of fields and their v	alues.	However, if the source query
	returns for example 10	D fields, you can decide	to send onl	y the last 8 in the AP	l if you	set an offset of 2

Mailbox	
Driver Parameters	
Assemble queries with same recip	ient in a single mail
Data Presentation	HTML table in mail body V
Use an HTML template file :	C\Users\Public\Documents\Fuzible\FILES\modeleMail.html
Keyword identifier in	at will be replaced by Query Results (formatted as an HTML table) TABLE_DATA ?
Assemble queries with same recipient in one mail	If the Job has multiple queries with the same email address as the Target, you can decide to group all the results into one email rather than send 1 mail / query.
Data Presentation	Here, we choose how the Source data will be presented in the email:
	- HTML table in mail body: an HTML table in the body of the mail
	- Excel file with (or without) a header: an attachment in Excel format
	<ul> <li>CSV file with (or without) a header: an attachment in CSV format</li> <li>Note: If the table contains too many rows (to be set in the software configuration), and one has chosen "HTML</li> </ul>
	table", it is a CSV attachment that will be attached to the mail rather than a table in the body
	Note 2: The name of the table in HTML table mode will match the alias of the first table of the query.
	Ex : SELECT * FROM MYTABLE -> MyTable will be the table header Ex2 : SELECT * FROM MYTABLE as My_Reporting -> My Reporting will be the table header (any underscore will be
	replaced by a whitespace as well)
Use an HTML Template file	Is showned only if you choosed « HTML table in mail body »
	E the sector little control of a data set is a face for a face of a data. The data is a data is a face is a bath
	Fuzible creates HTML content using data retrieved from Source Queries. The default Template is quite simple, that's why you can choose a customized one.
	In that case, Fuzible needs to know where to include the Source Query data into that Template. This is where the
	Keyword option stands for :
	1/ If no keyword is specified, Fuzible will behave like this :
	It will take the Query Alias (ie : mymail@mail.com:select * from myCustomers as MyAlias) and try to find it in the
	HTML Template.
	<ul> <li>If found (in the example, MyAlias), it will be replaced by the HTML code that has been produced from Query results.</li> </ul>
	- If not found, the results will be concatenated to the Template HTML code.
	<ul> <li>2/ If a keyword has been specified :</li> <li>- Fuzible will replace that keyword by the HTML code that has been produced from Query results.</li> </ul>
	- If the keyword has not been found into the Template, the results will be concatenated to the Template HTML code.
	3/ Special case when using "Assemble Queries with same Recipient in a single mail" If you have multiple queries that will be merged in a single mail, the Template can be populated smartly.
	- Your Template uses a keyword (ie : MYTABLE01) that is intended to handle results from the first Query
	- Your Templace uses another keyword (ie : MYTABLE02) that is intended to handle results from the second Query
	-> You can set your Queries like this :
	<ul> <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_TIMESTEMP</li> </ul>
	Results from 'MyCustomers' Query will be injected into the Template by replacing 'MYTABLE01' keyword.
	Results from 'MySuppliers' Query will be injected into the Template by replacing 'MYTABLE02' keyword.
	Practical note: if you have injected dynamic parameters recognized by Fuzible in your HTML Template (e.g.: {?1}),
	these will be replaced by the dynamic values of the Job!
Keyword Identifier	The optional Keyword included in the Template that will be replaced by the HTML table

# Note: The mail subject will be the description of the Job

Г

Active Directory Driver Parameters Key Attribute (Primary Key, name Existing Objects : Remo	
Activate created Entries	
Key Property	Sets the unique property that identifies an object in the AD (for example, "name" is the default single property for a user account)
Existing Objects	Chooses how to behave when writing an AD object: - Remove: It is removed for re-creation - Ignore: Leave the object as it is, without overwriting it Note: In "Data Synchronization" mode, this option is disabled because data is compared, so inputs will be updated, inserted or deleted depending on the mode of sync chosen in the main settings of the Job
Activate New Entries	When a new entry is created, it will not be activated by default. You can force its activation by checking this box

The heart of Fuzible is here. As the software is an IN/OUT reader/writer, a Data Source is no more than a group of fields and with values.

The software aims to greatly simplify the tedious work of mapping and transforming data. It uses the principle of SQL queries to work.

In case the source is an SGBD, no problem, it is the SQL language of the SGBD that will work, you can enter any query (simple or complex) compatible with it to extract data.

In all other cases, Fuzible relies on the standard SQL language (SQL92 compatible): The queries you write are translated and applied to the type of Source you are querying.

The sub-menu reminds you of the SQL language you are using when building your queries.

Job Configuration	Source : MySQL/MariaDB Database	Target : SQL Server Database	Queries	Log Viewer	
Job Queries (MyS	QL SQL) Sandbox				
A Query must start w ex : MyTable:SELECT (	ith the table name in which data will be written $(\dots)$	in (will be created if non-existent).			Show me an exemple

Each query for a Job is written as such: **OUTPUT:SELECT** [...]

If you click right over the query when it is empty, you'll be proposed a demo Query if you're not familiar with how it works. If the Source is a database or a file, you will be asked to perform a full Replication of everything that is available:

: MyTable:SELEC	
	Full Replication ? $ imes$
	Would you like to perform a full path (available files) replication ?
	Oui Non

This will produce the following result (in the example, my Source connection is a local path containing CSV files):

Job Queries	Sandbox
A Query must star ex : MyTable:SELEC	with the table name in which data will be written in (will be created if non-existent). T ()
*:SELECT *	FROM *.CSV

If you don't want a full replication, a demo Query will be added:



#### Now let us see how to write a query manually, since all the interest is there!

Output	
BDD	Destination table name         A Query must start with the table name in which data will be written in (will be created if non-existent).         ex : MyTable:SELECT ()         AX_f_fae_avoirs:select * from (SELECT T1.RECID AS id_ecriture,
	CAST(T1.ACCOUNTINGCURRENCYAMOUNT <i>AS</i> numeric(12,3)) <i>AS</i> nb_montant, CAST(LEFT(CONVERT(varchar, T2.ACCOUNTINGDATE,112),6) <i>AS</i> INTEGER) <i>AS</i> id anneemois,
FICHIER	Destination filename
	A Query must start with the filename in which data will be written in. ex : MyFile.CSV:SELECT ()
	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
	Note: In the example, the name of the output file is dynamic!
MAIL	The recipient's email address. It is possible to define several by putting a ";" between each! 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 ()
	mymail@hotmail.com:select * from (select snp.id_matricule as Matricule,
	<pre>concat(snp.li_nom, '', snp.li_prenom) as Nom_Prenom,</pre>
	ptc.li_typecontrat,
	pmfc.li_motiffincontrat as Motif_Fin_Contrat,
	pqc.li_qualifcontrat <i>as</i> Qualification_Contrat,
	pa.li_agence <i>as</i> Agence,
WEBSERVICE	Name of the API object that will receive the data (basically, the tail of the URL)
	A Query must start with the target Webservice Object (URL queue) in which the source data will be sent in. ex : /ws/mywebservice:SELECT ()
	sobjects/Account/:select * from Account.csv
ACTIVE DIRECTORY	Name of the AD object on which we will write data. The Source field names must match an existing attribute from the AD. You can alias the fields if they don't have a valid attribute name. (ex : SELECT myname as name, account as sAMAccountName FROM)
	users:select addata.sAMAccountName, addata.name, addata.description from ad_users.csv as addata

# How field mapping works

By simplification, a SELECT statement is written as follows:

SELECT \* FROM [SOURCE]

Or

Г

SELECT field1, field2 [...] FROM [SOURCE]

Or

SELECT field1 as dest1, field2 as dest2 [...] FROM [SOURCE]

Or

SELECT CONCAT(field1, field2) as dest1, TRIM(field2) as dest2 [...] FROM [SOURCE]

The field alias serves as a reference to Fuzible to build and fill the Target connexion. If it is not present, the name of the field is used, otherwise, it is the alias:

On a « SELECT \* », the Target's column names will be the same as the Source.

On a « SELECT field1 », the column name in the target will be "field1."

On a « SELECT field1 as dest1 », « dest1 » will be used as a field name in the Target.

... and so on.

Hence the usefulness of preparing your Source query well.

Important note: If you want to put an alias on the fields you retrieve, you must use "AS" Indeed, if a CSV file includes field names with whitespaces, Fuzible will be unable to separate the alias and the fields. Field framing is not supported (quotes or hooks around the fields) Example:

SELECT mon champ, mon deuxième champ FROM monfichier.CSV	ОК	
SELECT mon champ 1erchamp, mon deuxième champ 2emechamp FROM monfichier.CSV	NON-OK	
SELECT mon champ as 1erchamp, mon deuxième champ as 2emechamp FROM monfichier.CSV	ОК	

In case the Target is a SGBD, Fuzible compares the Source and Target fields, performs its "INSERT, DELETE, UPDATE" operations based on what is available in the Target only. For example, if you're querying a Source that has 50 fields and the Target has only 25 of these fields, that's no problem. The reverse is also true.

Regarding inter-compatibility (the Source would be SQL Server, the Target would be MySQL), Fuzible transcodes the data on the fly to make it compatible between both Source and Target, you don't have to worry about the data types.

Please also note that Fuzible SQL understands field framing if column/table names do not only use numbers/letters :

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

# **Special cases of synchronization queries:**

Sync. works by comparing Source and Target data. This way of working requires the same query to be performed in both environments. The software knows how to transcode most queries but there are some limitations:

### Example 1:

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

→ The query that will be executed on the Target will be :

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

The « WHERE » filter has been transcoded.

Another case :

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

→ The query that will be executed on the Target will be :

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

The problem is that "champ3" does not exist in the SELECT statement. Fuzible cannot know what "field3" refers to in the Target table. If by chance this field exists (ISO-perimeter tables), it will not be a problem, but if this field does not exists (because the source query is complex, the conditions refer to fields on join tables, sometimes very complex conditions (nested SELECT ...)) the query will not succeed, and the sync. will surely fail.

An SQL trick to make up for this particularity: have the source SGBD execute a nested query:

Let us take this complex query in Dynamics AX. It contains several transformations and conditions. By framing it as a sub-query, Fuzible will then only worry about the main query to make its transcoded query.

AX\_f\_ecritures:SELECT \* FROM (SELECT T1.RECID AS id\_ecriture, T1.ACCOUNTINGCURRENCYAMOUNT AS nb\_montant, T2.ACCOUNTINGDATE AS dt\_ecriture, 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\_ecriture, 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 GENERALJOURNALENTRY T2 ON (T1.GENERALJOURNALENTRY=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.PRIMARYFORLEGALENTITY=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

```
Will be transcoded for the target as: SELECT * FROM AX_f_ecritures WHERE id_anneemois_saisie >= 202001
```

Now let's see what can be done with an SQL query:

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 He
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] <i>as</i> 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 MAINACCO	OUNTID) B <mark>ON</mark> A.R
ORDER BY A.[MAINACCOUNTID]	

Show me an exemple

Any "SELECT" statement that is compatible with the source SGBD. This can be a simple or a more complex one.

# SELECT - From a database, to a file

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

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

Any "SELECT" query that is compatible with the source SGBD. This can be simple or complex.

### SELECT - From a database, to an email address

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 (..)
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

Any "SELECT" query that is compatible with the source SGBD. This can be simple or complex. **Note:** The source query makes use of Dynamic Parameters, in this example, a dynamic @mail address is set in the "Job Configuration" tab, and used as the Output

Apart from a query on a SGBD, Fuzible's SQL engine takes over:

# SELECT - From a file 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 = 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\_OV *AS* fileB ON fileA.id\_sample = fileB.id\_sample = fileB.id\_sample\_join

See here a somewhat complex example of queries made on multiple files, with joins, and even a sub-query. The SQL syntax is strictly the same as that of a traditional SGBD.

#### SELECT - From a webservice using Fuzible SQL (A) A Query must start with the table name in which data will be written in (will be created if non-existent). $\kappa$ : MyTable:SELECT (...) Quick Help glpi\_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, Reg.id as id requesttype, Reg.name as li\_requestype, 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 This example is quite complex and shows how much a query can be made in such a way that it would be carried out in a SGBD. On the other hand, unlike a SGBD, the header is not known in advance (fields retrieved from the API), it is advisable to do first some "SELECT \*" to see what's returned by the API before using joins and other transformations.

The "Sandbox" tab allows you to do some testing.

Example of the GLPI software API:

Simple query:

SELECT \* FROM Computer/?range=0-5000 Brings back the list of all computers (5000 entries)

I can of course use some filtering:

SELECT \* FROM Computer/?range=0-5000 where states\_id = 1

→ I know that the field "states\_id" exists in what brings me the API so I can filter on this field

Similarly, if I know the list of fields, I can ask as follows:

SELECT serial as Serial, name as Machine, users\_id as Glpi FROM Computer/?range=0-5000 WHERE states\_id=1

# SELECT - From a webservice using 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 : MvTable: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"'}]

For example, this API returns 2 data tables. I can choose to get both of them (will create 2 tables in the Target), or get only one of the 2 : The "TABLE x" function here determines the data table on which the query applies. The other table will be returned as a simple "SELECT \*" statement, because it is not possible to query several tables from a single query.

It is possible to get only one of the 4 tables thanks to the syntax "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 "" }]

Regarding the API we are querying, we can integrate some body content into the query. The API documentation of the webservice specify how to filter one of the fields using a body using JSON.

→ The hooks are used to integrate some body content (XML, JSON or Form-data)

On the other hand, unlike a SGBD, the header is not known in advance (fields retrieved from the API), it is advisable to do first some "SELECT \*" to see what's returned by the API before using joins and other transformations. The "Sandbox" tab allows you to do some testing.

# **SELECT – From Salesforce API using SoSQL**

When you have set up the Source connection with a template that has a specific simili-SQL language (graphQL, NxQL, SoQL), you can use this language to query the API instead of using Fuzible SQL, which will make the transcoding work much easier for you. In the following example, we're querying the Salesforce CRM using SoQL :

Job Queries (SOQL) Sandbox

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

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

While the same query, using Fuzible SQL mode, looks like this, which is much less convenient:

Job Queries (Fuzible SQL) Sandbox

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

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

#### **SELECT - From an e-mail box**

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'

Example of a GMAIL address:

SELECT \* FROM mymail@gmail.com[mypassword]

Will bring back the emails list from the address "mymail@gmail.com" using the password entered in brackets.

SELECT \* FROM <u>mymail@gmail.com</u>

→ Will bring back the emails list from the "mymail@gmail.com" address using the password entered in the connection string.

I can of course filter and name the columns if I know them (recall: the SANDBOX is made for this): SELECT SUBJECT, SENDER, TEXTBODY, TO FROM mymail@myProvider.com WHERE DATE >= '01/01/2019'

**SELECT - From Active Directory** 

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

test\_ad\_grp.csv:select \* from groups

As with webservices and maiboxes, Fuzible engine does not know in advance all the fields that can be returned by the AD domain. It is advisable, to test to make a simple SELECT \* FROM USERS

Quick Help

"FROM" refers to the AD object being queried (USERS or GROUPS)

The example brings back the list of AD groups that exist.

#### **QUERY ASSISTANT**

Like SQL Tools, the software offers query entry facilities through a pop-up menu that displays based on the words you are writing:

In the following example, I am querying an SGBD and just have entered "FROM": the menu then shows me a list of available tables. A simple "TAB" allows me to access this floating menu, the arrows allow to choose a table, while "ENTER" inserts the chosen item in the query.

Query must start with the table name : MyTable:SELECT ()	e in which data will be written in (will be created if non-existent).	Show me an exemp
yTable:select * from		
, rabieletetetetetetetetetetetetetetetetetete	•	
	user_connstrings	
	user_connstrings_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 g	uery for advanced operations or press 'F5' to show Data	Quick Hel

In the following example, I already have a table and I complete my "SELECT" statement: the list of fields is presented, as well as the classic SQL functions that can be used.

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 ()	Show me an exemple
myTable:select st1. from sample_table_1 as st1	
id_sample	
li_sample	
dt_random_date	
nb_random_number li random string	
id_group	
id_ssgroup	
Right-click on a query for advanced operations or press 'F5' to show Data	Quick Help
	Gardeerholp
Synchro : Bypass query filters in Target	

The Query Assistant works for all types of Sources, it is much more advanced for file queries and SGBD because for other cases, there is no method to, for example, expose the list of objects of a webservice and even less the list of available fields. In this case, it simply serves as an assistant to the creation of an SQL function (CONCAT, ISNULL...).

# ADVANCED "SELECT" STATEMENT

The previous examples are relatively simple, they show how one can basically query a Source in the form of "SELECT" when it is not a database.

Fuzible's SQL language understands more complex syntax, and writing an SQL query into the tab is augmented by a Query Assistant that allows you to see and use all available query options on the fly.

If for example I type "SELECT C", a small pop-up menu will offer me several things:

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

I can then use one of these elements to manipulate the source data. Examples:

- 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

To measure the full range of possibilities offered by the SQL engine, you have a default "Sample" Job that has several queries using all of these advanced features. You can learn from it.

#### There's also some kind of "Anonymization" feature :

- SELECT ANONYMIZE(myField) FROM myFile
   Will mix "myField" values between them
- SELECT ANONYMIZE(myField, "RANDOM") FROM myFile Will randomize "myField" values
- SELECT ANONYMIZE(myField, "myRandomizationFile.csv") FROM myFile
   Will load the specified file and use the values to replace original "myField" values

#### **Fuzible SQL Technical limitations:**

1) Nested SELECT only work on "tables" and "where" filters (see example "webservices A"), you can't make a sub-select for a field:

#### Valid:

A Query must start with the filename in which data will be written in. ex : MyFile.CSV:SELECT ()	Quick Hel
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
alid:	
. Query must start with the table name in which data will be written in (will be created if non-existent).	
x : MyTable:SELECT ()	Quick He
test:select *	

from (select \* from monfichier) as a

#### Invalid:

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

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

You cannot nest the sub-selects. The query will not succeed.
 ex: SELECT \* FROM (SELECT \* FROM (SELECT \* FROM monfichier) as data) as data

#### Additional note:

Some sources may return multiple data tables (webservices, JSON or XML files).

By default, the software will create as many targets as returned tables. However, all SQL processing operations associated with your query (SQL, Group By, Where, Order By...) will be applied to the first table only.

For some reason, if you want your query to apply to another table, it's possible to indicate it by doing so:

OUTPUT :SELECT TABLE x champ1, ...

... indicating, thanks to "TABLE x," the table number to which the query applies.

If "x" is invalid (example: I put TABLE 10 when the source returns only one table, all operations associated with the query will be cancelled.

*Of course, you cannot guess in advance what the data source is made of. The "sandbox" tab is there for that. Make a simple "SELECT \* FROM masource" and an* **F5** *to find out all the results returned by the query. If the result returns multiple tables, you will see it, and then you'll be able to see the contents of each table.* 

You can also say "ONLY" if you only want to retrieve the requested table (otherwise, the others will also be processed).

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

Look at the example job to understand what can be done and most importantly, experiment using SANDBOX. The program is verbose enough to allow you to debug step by step a query that would be wrongly written.

## **SELECT Multi-tables, multi-files**

## BDD

You can get multiple tables at once, if for example you want to retrieve all the tables that contain the word "param," you can enter the query as well:

OUTPUT : SELECT \* FROM %param%

Fuzible detects the use of the "%" and will search for all the tables that match this pattern. It then turns a single query into several. It goes without saying that unless the names of the fields of these tables are all the same, the "SELECT \*" is recommended...

There are three scenarios for driving the output:

- If you put « \* » (ex: \*:select \* from %param%), « \* » will be replaced by the name of the table.
- If you put a forced name (ex : param:select \* from %param%), the data will all go in the same output, here: "param". All data coming from any query with that pattern will be merged.
- If you put "import\_\*" (ex : param\_\* :select \* from %param%), the "\*" will be replaced by the name of the table, but the final table will have a name that will begin with « import\_ »

#### **FICHIERS**

In the same way as with a database, you can query multiple files at once, for example, if you want to retrieve all the files that contain the word "param," you can enter the query like this (like a "DIR" in the MSDOS command prompt)

OUTPUT :SELECT \* from \*.CSV

Fuzible detects the use of the "\*" and will query all CSV files from the connection string (path).

The use of the '\*' can be extended as follows:

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

→ This means that files starting with "FI" and then having something else afterwards, then "test", then something else, will be loaded.

As with databases, you can name fields and work the query in an advanced way (with transformation functions... etc. ..) but in this case, the files must be of the same structure and, if they have a header, it must be the same.

Driving the Output works in the same way as previously explained.

#### **Multi-Target Queries**

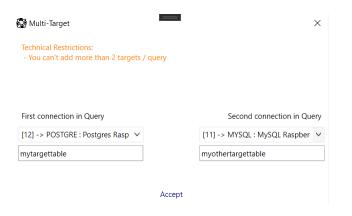
By default, a Job works with a single Target.

However, this can be changed to work with 2 Targets in parallel (in case for example we want to feed 2 databases identically, simultaneously)

Let us take a simple example: build a Job query, then click right on it. A pop-up menu appears and offers you several options including "Create Dual Target"

Job Queries Sandbox		
A Query must start with the table name in which data will be written in (will be create ex : MyTable:SELECT ()	d if non-existent).	Show me an exemple
mysynchrotable:select tb1.id_sample as idSample, tb	1.li_sample <i>as</i> liSample,	
tb1.dt_random_date as RandomDate, tb2.SecondColumn	n <i>as</i> AnotherColumn	
from sample1 as tb1	Postgres Raspberry -> mysynchrotable	
inner join sample2 as tb2 ON tb1.id_sample = tb2.Firs	Query Analyzer	
where 100 > tb1.id_sample	> Source Infos	
and tb1.id_sample > 10	> Target Infos	
and (tb1.id_sample > 11 and tb1.id_sample < 99)	> Query Details	
and tb1.li_sample not like 'weirdstring'	Synchro Query	
order by tb1.id_sample desc	> 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	
0	> Add Cross-Connections Join	
	> Create Dual Target	

A prompt will then ask you to choose your 2 Targets, and the associated output. Confirm.



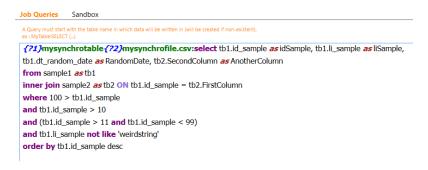
The query will be augmented by a small script:



Between brackets, the Connection ID, followed by the name of the target table. A new right-click to view the pop-up menu, to which a "Target B Info" item has been added, that provides information about the chosen Target.

But we can also imagine filling a database table, and a file, in parallel. Anything is possible. Target connections can even be dynamic using the Dynamic Parameters:

The query:



#### Dynamic parameters:

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			
Dvnamic Parameters	[12];[1]				?
	ailable command. Each parameter m series, text fields, connections) by ref			View Job With Replaced Value	s
					$\sim$

This is of interest when you want to switch data from one environment to another (pre-production, production, developmen, etc.) on the fly.

Moreover, one can choose to fill the 2 Targets in parallel or one after another, it all depends on the performance of the computer that hosts Fuzible (see General Configuration / SHS Analyzer)

#### **Technical limitation:**

Multi-Target is limited to 2 targets.

#### **Cross-Queries**

This feature is one of Fuzible's most powerful. It allows you to query different data sources within a single query: you can get data from a database from a file and complete it with data from a webservice. Anything is possible and can be achieved in a fairly simple way.

To make this concept clearer, imagine that instead of joining between several SQL tables to complete a dataset, you make joins between several different connections.

Let us take the following example. My main connection is a MySQL database.

Build a simple query, then click right on it. The pop-up menu appears and offers several options including "Add Cross-Connections Join."

Query must start with the table name in which data w : MyTable:SELECT ()	ill be written in (will be created if non-existent).	Show me an exemp
iytargettable:select * from sam		
	MariaDB Synology -> mytargettable	
	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	
	> Basic Query Builder	
	Advanced Query Scripting	
	> Add Cross-Connections Join	
	> Create Dual Target	
Right-click on a query for adv	anced operations or press 'F5' to show Data	Quick He

A prompt then asks you to choose a new data Source, and the type of join that will be made between the two Sources. I pass on the very advanced "Optional Query Filter" feature, which filters the results.

Cross Queries		×
Same column names between both sour SELECT idclient as mylink, li_client as cli [[2]] SELECT client_id as mylink, creati -> The column called 'mylink' will becor You can use up to 5 key columns to per	ent_name FR on_date as cl me the prima	OM mytable ient_creation FROM myfile.csv my key to link both sources
First connection in Query	Join Type	Second connection in Query
[11] -> MYSQL : MySQL Raspber \vee	$_{\rm INNEF} \sim$	[5] -> SQLITE : Local SQLite File 🗸 🗸
Optional Query Filter (WHERE somethin	INNER LEFT RIGHT	

A script area [--[5]] will appear in the query. You can then write a new query. This one will be associated with the second connection. A right-click and you will find that the menu contains a new sub-menu associated with this cross-query.

Query must start with the table name in : MyTable:SELECT ()	which data will be written in (will be created if non-existent).		Show me an	exemp
ıytargettable:select * fı	om sample1 [[5]] select * from sample_table_3 MariaDB Synology -> mytargettable Query Analyzer > Source linfos	3		
	> Target Infos > Query Details	•		
	+ Cross-Query (Local SQLite File)	•	> Cross-Query Behavior	•
	Execute Query   Load Source Data (FS)  Run this individual query		<ul> <li>&gt; Search Join-Link and Check the Cross-Query</li> <li>&gt; Load Source Data</li> <li>Fields</li> </ul>	•
	Scripting  Set full header from query and copy/paste it Add a dynamic parameter Basic Query Builder	•	Tables Conditions Group By Order By	•
	Advanced Query Scripting > Add Cross-Connections Join > Create Dual Target			
0				
Right-click on a que	ry for advanced operations or press 'F5' to show Data		Qu	ick Hel

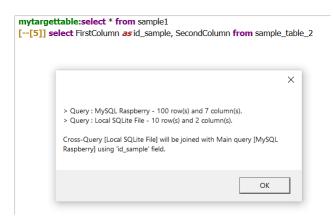
In the script area, the type of join expresses itself as follows:

	INNER JOIN
>-	RIGHT JOIN
<-	LEFT JOIN
$\diamond$	OUTER JOIN

The connection is shown between brackets.

Fuzible has only one way to define how to join the two sources between them: he uses the fields with the same name to make his join. In the previous example, a "SELECT \*" is performed on both sources, but I know that the "id\_sample" field is exists on both sources, Fuzible will join using this field.

On the other hand, if the fields names are different from one connection to another, the fields should be specifically named. For example:

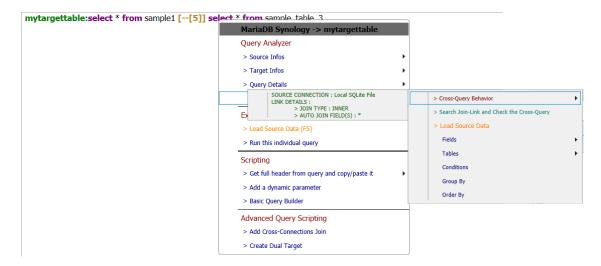


Thanks to the alias, I force the link between the two sources with "id\_sample".

### Behavior of an cross-query

The contextual menu makes it easier for you to understand Fuzible's behavior related to the cross-query. In the example below, SELECT \* being used in both connections, the engine will only be able to determine the link when the queries are executed: You need to be sure that the "id\_sample" field exists in both datasets.

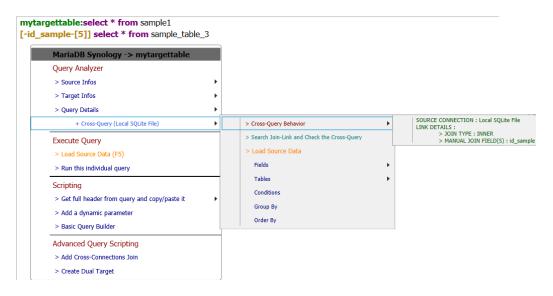
Warning: Any other field with the same name would also be considered as part of the key!



Another way is to make a SELECT statement that specifies fields names. The contextual menu then tells you which join field or fields are associated with it. There is no need to wait for the execution to know the behavior that will be performed.

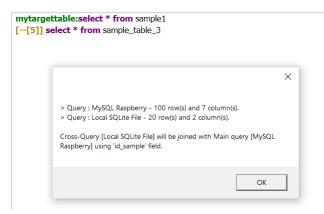
MariaDB Synology -> mytargettable				
Query Analyzer				
> Source Infos	•			
> Target Infos	•			
> Query Details	•			
+ Cross-Query (Local SQLite File)	•	> Cross-Query Behavior	•	SOURCE CONNECTION : Local SQLite File LINK DETAILS :
Execute Query > Load Source Data (F5) > Run this individual query		<ul> <li>&gt; Search Join-Link and Check the Cross-Query</li> <li>&gt; Load Source Data</li> <li>Fields</li> </ul>	•	> JOIN TYPE : INNER > AUTO JOIN FIELD(S) : id_san
Scripting > Get full header from query and copy/paste it > Add a dynamic parameter > Basic Query Builder	۲	Tables Conditions Group By Order By	•	
Advanced Query Scripting				
> Add Cross-Connections Join				
> Create Dual Target				

Finally, the third way is to change the cross-query script: you can manually define the join field(s) by separating them by a comma. The contextual menu specifies that the type of join is now manual.

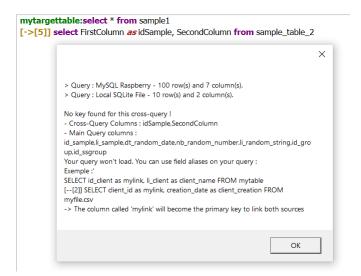


#### **Results**

By going to "Cross-Query/Search Join-Link and Check the Cross-Query", Fuzible will perform a test to check the viability of the cross-query. The result will be presented as follows:



On the other hand, if there is no link, here is what will be displayed; in this example, I forced the names of the columns of the second Data Source, putting an "idSample" name that does not exist in the first Data Source (id\_sample).

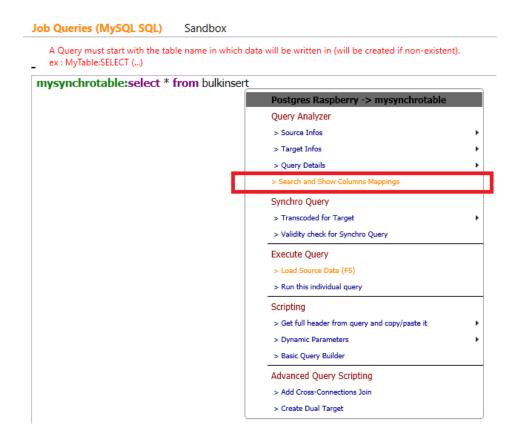


#### Search and Show Column Mappings

Because one of the aspects of Fuzible is to avoid the well-known ETL mapping tasks, there's an option that allows you to see the mapping that will be performed between Source and Target columns, and if there are some orphan columns, a Query suggestion will also be showned if you need those columns to be mapped as well.

Fuzible will automatically find the closest name correspondances from the orphan Source and Target columns.

You'll be able to copy-paste the suggested Query instead of your existing one.



		mation
our Que	ry.	that unmap tom of tha
		table : ill be ma
Source column	Link	Target column
id_bench_01	OK	id_bench_01
S_beach_01	OK	S_beach_01
nb_bench_01	ок	nb_bench_01
dt_bench_01	OK	dt_bench_01
id_bench_02	ок	id_bench_02
%_bench_02	OK	%_beach_02
	ок	nb_bench_02
nb_bench_02		dt_bench_02
nb_bench_02 dt_bench_02	OK	
	OK OK	id_bench_03
dt_bench_02		id_bench_03
dt_bench_02 id_bench_03	ок	
dt_bench_02 id_bench_03 %_bench_03	OK OK	%_bench_03
dt_bench_02 id_bench_03 %_bench_03 nb_bench_03	OK OK OK	%_bench_03

#### Showing Source Data

The contextual menu contains a "Load Source Data" option that allows you to view the results, whether it is all the data from the cross-query, or just the data from each individual query.

In the table below, I made a cross-query using a "LEFT JOIN", and loaded the data from the cross-join between the two Sources. Only "id\_sample" (from 90 to 100) records are present in the second connection, which explains why "SecondColumn" is empty in other cases.

review (I	rows)	500			Data	Successful	ly Loaded.	With data analyzer	
ource Da	ata								
DataTable N Namespace Row Count : Fields Coun	: sample1 : 100				al Properties ROSSJOIN_L		le-≻id_sample		
id_sample	li_sample	dt_random_date	nb_random_number	li_random_string	id_group	id_ssgroup	SecondColumn		
0	seven-zero	10/8/2020 1:10:49 PM	9.0625	piynoetghykxighe	000	000			
'1	seven-one	10/19/2020 1:10:49 PM	31.0625	hsxnmgwrqwhvjnyc	001	001			_
2	seven-two	10/27/2020 1:10:49 PM	59.6250	kudzjrucwhciottx	002	002			
3	seven-three	12/19/2020 1:10:49 PM	33.6250	dbvsgxnxzmpxzgui	003	003			_
74	seven-four	1/22/2021 1:10:49 PM	41.6250	iijftkkooihnclam	004	004			-
75	seven-five	11/13/2020 1:10:49 PM	29.1250	dsdfmlsimyiwnfzc	005	000			_
76	seven-six	8/31/2020 1:10:49 PM	16.4375	dxhkhbjeqljdxnhj	006	001			-
77	seven-seven	9/24/2020 1:10:49 PM	9.0625	mcvxnlymgsubozwv	007	002			_
78	seven-eight	11/19/2020 1:10:49 PM	7.1250	cskbwhvhyiavoeay	008	003			
79	seven-nine	9/12/2020 1:10:49 PM	33.1250	ifkwmvvqjpwkhceh	009	004			_
30	eight-zero	10/25/2020 1:10:49 PM	34.4375	qkgocfmplkdqbpjk	000	000			
31	eight-one	9/17/2020 1:10:49 PM	12.4375	pffogfxgkobyrgwn	001	001			_
32	eight-two	12/2/2020 1:10:49 PM	16.8750	gudejxkIntohaxpb	002	002			-
33	eight-three	8/18/2020 1:10:49 PM	8.6875	hceagwvynmswtnhe	003	003			-
34	eight-four	9/5/2020 1:10:49 PM	45.5000	zlwieuserdoejppv	004	004			_
35	eight-five	8/3/2020 1:10:49 PM	18.3750	afegjljxsgzazvtx	005	000			-
36	eight-six	8/26/2020 1:10:49 PM	20.3125	fyxkileskjddhyhc	006	001			_
37	eight-seven	11/25/2020 1:10:49 PM	58.3125	nsbkygcbfrdslmni	007	002			_
38	eight-eight	9/28/2020 1:10:49 PM	22.0625	hqxscraatpibsxqv	008	003			
	eight-nine	12/23/2020 1:10:49 PM	25.5000	euaqhggnorxrpyha	009	004			-
90	nine-zero	8/19/2020 1:10:49 PM	38.0000	avqkjxnntdygcqce	000	000			-
91	nine-one	12/26/2020 1:10:49 PM	54.8125	gqirlocjcbzkuscb	001	001	nine-one		۰.
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	gijbhaulnijfdbvp	003	003	nine-three		- 1
94	nine-four	1/1/2021 1:10:49 PM	26.8125	menoejauuumppjzz	004	004	nine-four		
95	nine-five	1/26/2021 1:10:49 PM	31.7500	dqmkhshesifnmzxl	005	000	nine-five		
96	nine-six	10/14/2020 1:10:49 PM	38.1250	qseabnfzyxtagdfj	006	001	nine-six		
97	nine-seven	9/27/2020 1:10:49 PM	55.9375	wbwugaqrpginsdkk	007	002	nine-seven		- 1
98	nine-eight	11/8/2020 1:10:49 PM	13.3125	hdozehujkscbidoa	008	003	nine-eight		
99	nine-nine	8/6/2020 1:10:49 PM	55.8750	rvlekjyyrlfqpvuc	009	004	nine-nine		-
100	one-zero-zero		23.3125	ixxpaxpxpnyfyrpn	000	000	one-zero-zero		- 1

Export content as CSV

.....

## Conversely, an "INNER JOIN" would have given this:

Preview	rows)	500			Data	Successfu	Illy Loaded.	With data analyzer
Source D	ata							
DataTable I Namespace Row Count Fields Cour	: 10				nal Propertie CROSSJOIN		ple->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	gijbhaulnijfdbvp	003	003	nine-three	
94	nine-four	1/1/2021 1:10:49 PM	26.8125	menoejauuumppjzz	004	004	nine-four	
95	nine-five	1/26/2021 1:10:49 PM	31.7500	dqmkhshesifnmzxl	005	000	nine-five	
96	nine-six	10/14/2020 1:10:49 PM	38.1250	qseabnfzyxtagdfj	006	001	nine-six	
17	nine-seven	9/27/2020 1:10:49 PM	55.9375	wbwugaqrpginsdkk	007	002	nine-seven	
18	nine-eight	11/8/2020 1:10:49 PM	13.3125	hdozehujkscbidoa	800	003	nine-eight	
	nine-nine	8/6/2020 1:10:49 PM	55.8750	rvlekjyyrlfqpvuc	009	004	nine-nine	
99	one-zero-zero	12/12/2020 1:10:49 PM	23.3125	ixxpaxpxpnyfyrpn	000	000	one-zero-zero	

#### The data from the cross-query:

Preview (	rows)	500	Data Successfully Loaded.	With data analyzer
Source D	ata			
			Additional Properties :	
id_sample	SecondColumn			
100	one-zero-zero			
00	nine-nine			
99	THE THE			
	nine-eight			
98				
98 97	nine-eight			
98 97 96	nine-eight nine-seven nine-six nine-five			
98 97 96 95 94	nine-eight nine-seven nine-six			
98 97 96 95 94 93	nine-eight nine-seven nine-six nine-five nine-four nine-three			
99 98 97 96 95 94 93 92 91	nine-eight nine-seven nine-six nine-five nine-four			

#### Cross-queries, data comparison feature

This mode allows you for example to compare data from two databases (to make a report of differences between the two, for example)

By default, the "cross-query" feature is simple to use. But like any automatic feature, it is rigid. However, it can be made more flexible by filtering the results.

This example compares tables in a production database with a pre-production database (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: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, '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

You can see that the framed script area [<>[75]...] uses a conditional statement that allows comparison between the two result sets.

→ This way of scripting allows, once the data is collected, to filter the results according to the filter shown in the script area.

If the target is an email address, the result will be as follows:

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	

## **Contextual Query Menu**

The principle of using SQL to Synchronize or Replicate data can represent a fairly significant level of abstraction. That is why Fuzible has a useful contextual menu, accessible when the mouse is on a query. A right click and you can do the following:

- Get advanced information about the query
- Test the validity of the Sync. query (Synchronization Mode)
- Execute any of the job's queries without necessarily executing them all
- To have a script assistant for advanced functions

## Source Connection Summary

FILE	For a local file, the path and type of connection
	MyTutorial:SELECT id_sample, SUBSTRING(ii_sample, 0, 5) as ii_sample, id_group as RenommedColumn, dt_random_date, ii_sample as ii_test,
	CONVE Local SQLite File -> MyTutorial
	WHER         Query Analyzer           > Source Infos         DRIVER : CSV File CONNI. NAME : Local Path INPUT PATH : Clubers/gutzmisource/Workspaces/Fuzbiel/Fuzbie
	For a (S)FTP, the configuration of the (S)FTP and the path on the server test:select * from /var/www/sftp/SAMPLE.CSV
	Local Path -> test
	Query Analyzer     DRUKEN: CON File       > Source Infos     CONN. NAME: SYTP nuble       CONN. NAME: SYTP Fuelbe     (SYTP FUE):
	> larget thros PATH + > Query Details US SFTP : True USENIAME, PASSWORD :
	PORT: 22 Execute Query AUTHENTICATION BY KEY FILE : False SOURCE FILE(S) : /var/www/sfbg/SAMPLE.CSV
BDD	For an SQL connection, the driver, and the name of the database MyTutorial.csv:SELECT user as userColumn, connstring_id as idColumn, connstring_name as nameColumn
	from user_connstrings WHERE user LIKE '{?4}'
	Local Path -> MyTutorial.csv Ouery Analyzer
	Source Infos     DRDRE: Selite Database     CONN, NAME: Local SQLte File     CONN, NAME: Local SQLte File     DATABASE:
WS	Shows the whole behavior Fuzible uses to call the API. Check the URL, the authorization method.
	sample1:SELECT * FROM /ws/public/sample_data
	MySQL Raspberry -> sample1           Query Analyzer           > Course Infer           > Course Infer
	CONIN. NAME : Fundle Webservice de démonstration BASE URL : https://www.fuzible-app.com BASE URL : https://www.fuzible-app.com FULU RL: https://www.fuzible-app.com/vs/jubic/sample_data
	Query Details     AUTHORIZATION NETHOD: API_AUTH_HEADER     YOKKI WILL BE ADDED TO QUERY HEADER: Authorization:     METHOD: POST
MAIL	Summary of settings used to connect to a mailbox
	DRUVER : Mallox
	CONN. NAME : Gmail Connection Source Intos Source Intos P SMTP HOST : straty.gmail.com > Target Infos >
	SMTP PORT: \$37 SOURCE MALL : SRICE:
	PASSWORD:     > Load Source Data (F5)       SSL: True     > Load Source Data (F5)       PADTOCOL: IMAP     > Run this individual query
	UNXNOVIM FIELD(5) : SUBJECT, UID INFO : Use 'SELECT A final methods names. It will retrieve all available fields. Otherwise, you can for exemple write a query like this : 'SELECT BODY, FROM, DA
AD	The search query, the perimeter
	ad_users:SELECT * FROM users
	MySQL Raspberry -> ad_users
	Query Analyzer           > Source Infos           DRIVER : Active Directory
	COUNT MARE: Fuzible Active Unectory Demo     AD 00BECT : (&(objectClass=user)(objectClasseuser))(objectClasseuser)(objectClasseuser)(objectClasseuser)(objectClasseuser)(objectClasseuser)(objectClasseuser)(objectClasseuser)(objectClasseuser)(objectClasseuser)(objectClasseuser))(objectClasseuser)(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser))(objectClasseuser)
	> Query Details

BDD	- The name of the target database	
000	- The destination table	
	- Any pre/post-Job command(s) that have been set	
	MySQL Raspberry -> mytable	
	Query Analyzer	
	Source Infos     DRIVER : MySQU/Maria0B Database	
	> Target Infos     Collin, Value: ImySQL manufor busidesse     Collin, Value: ImySQL manufor busidese	
FILE	- All the information about the target: the path, and, in case of an "OUTPUT" with a pattern (creat	ion of
	several files), the behavior that will be adopted.	
	- Any pre/post-Job command(s) that have been set	
	CSV:	
	myfile.csv:select id_sample, li_sample from sample	
	Local Path -> myfile.csv	
	Query Analyzer Source Infos	
	> Target Infos	
	Query Details     OUTPUT PATH: C:\Users[guicm]source\Workspaces[Fuzble[Fuzble[bin]x64]Debug FILES\     OUTPUT FLIENAME: mm/lic.cv     CSV HEDR: id_smm/le_smm/	
	XML:	
	myfile.xml:select id_sample, li_sample from sample	
	Local Path -> myfile.xml	
	Query Analyzer  > Source Infos	
	Target Infos     DRIVER : XML File CONN. NAME : Loal Path	
	> Query Details OUTPUT PATH: :: (Users)quizm(source)Workspaces/Fuzible/Sirubielbin/s64Debug/FLES\ OUTPUT PLENAME: rm(blenam) XML STRUCTURE : xml version=1.0? <sample>&lt;@users/lisample&gt;value</sample>	
	EXCEL:	
	myexcelfile.xlsx:select * from sample as MySpreadSheet	
	Local Path -> myexcelfile.xlsx	
	Query Analyzer  > Source Infos	
	Target Infos     DRIVER : Excel File     CONN. NAME : Local Path	
	Voury Details     OUTPUT PATH : C:Usersignitm/source/Vordspaces/Fucible/F	
	STVLE : Light     STVLE : Light     CSVHEADRE : Will be determined at runtime. (SELECT ** vas used in Source Query, Fuzble cannot retrieve column names now)	
WS	webService=dossier.{??}.{?1}:SELE( Webservice RHPI (Personnel/Contrats/RIC) -> webService=de Query Analyzer	ssier.ric.modifier
	> Source Infos	•
	AUTIORIZATION NETION OF HTTP > Land Source Data (75) > Land Source Data (75)	,
		•
	VisS_QERY / Wile determined at runtime(SEEC*** was used in Source Query, Aleplants cannot retrieve solumn names nov)     VisS_QEREFICEVENT, Wile determined at runtime(SEEC***********************************	
	• DTLAP 1: Data of quary     • DTLAP 1: Data of quary     • DTLAP 2: The determined at nutrime     • DTLAP 2: The	,
	OFTIONUL FEED S: ul pop, shautonote, manoularid, contract Note : Vieleension will automatally create, add and fill to som fields in target table     Some Second Sec	,
	Advanced Query Scripting	
	> Add Cross-Connections Join > Create Dual Target	
	Shows the entire URL Fuzible has built to send data to an API.	
	Checks the URL and behavior of the call.	
MAIL	A Query must start with the mail address(e) for which you want to send data to (addresses have to be separated by a `.).	
WAIL	A Query must start with the mail addresses for which you want to send data to (addresses have to be separated by a ',). Show me an exemple example and the addresses have to be separated by a ',). Show me an exemple fuzible@fuzible-app.com:select id_sample,it_random_date,nb_random_number,it_random_string,id_group,id_ssgroup	
	from sample1 as MyChart WHERE id_sample> {71}	
	Fuzible Email -> fuzible@fuzible-app.com	
	Query Analyzer > Source Infos	
	> Target Infos   DR/ME : Hubble Email	
	> Query Details MLL SUBJECT + 9 Enail RECIPIENT(5) : fuable#fuable-app.com CHART TITLE : MyChart	
	Indicates:	
	- The subject of the mail (retrieved from the job description)	
	- Recipients (OUTPUT of the query)	
	- The name of the data table (retrieved from the alias of the query:	
	SELECT * from matable AS My_Chart -> Affichera My Chart	
	(Underscores are systematically replaced by a whitespace)	

AD	users:select user as name, param1 as des	scription from user_	_parameters	
	Fuzible Active Directory Dem	o -> users		
	Query Analyzer			
	> Source Infos	•		
	> Target Infos	•	DRIVER : Active Directory CONN. MARE : Fizible Active Directory Demo	
	> Query Details	•	AD QUERY : (&(objectClass=user)(objectCategory=person)(name=[DATA FROM : name])) AD OBJECT TARGET : users	
	Indicates:			
	- The AD object	in which o	data will be written in	
	- The search qu	ery that w	vill be performed	

## **QUERY ANALYZER - View Data**

This option opens a new window that will allow you several things:

- Load source data to preview it
- Have information on each source field
- Test the sync. mode

By clicking on "Load Source Data," the software will load the source data and display a 500-row preview (can be changed).

If the Job is a Sync. Job, each tab (Target, Insert, Update, Delete) will show you everything the sync. will do.

🖗 Show Source D	ata							
Data Preview			Data	Succes	sfully Loaded.		With Data Ana	lyzer
						Pre	eview (rows) :	500
Source Data	Target Data To Inser	t To Update To Delete						
DataTable Name : AXP Namespace : A Row Count : 2503 Fields Count : 8	PROD	Additional Properties : [AX_dim_plancomptabl	e] SYNCH	RO_PRIMA	RY_KEY : id_compt	ecomptable		
id_comptecomptable	id_racine_comptecomptable	li_comptecomptable	id_type	li_type	SYNCHRO_TAG	DBNAME	DTLOAD	
101000	1	CAPITAL APPELE	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
10100000	1	Capital souscrit (Sociétés de capitaux -	3	Résultat	L	AXPROD	5/22/2020 11:44:59	AM
101100	1	CAPITAL SOUSCRIT NON APPELE	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
104100	1	PRIME D EMISSION	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
104200	1	PRIME DE FUSION	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
104300	1	PRIME D APPORT	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
106100	1	RESERVE LEGALE	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
106110	1	RESERVE LEGALE	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
106200	1	Réserves indisponibles.	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
106300	1	RESERVES CONTRACTUELLES	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
106400	1	Réserves réglementées	3	Résultat		AXPROD	5/22/2020 11:44:59	AM
106480	1	RESERVE SPEC ART 238B CGI	3	Résultat		AXPROD	5/22/2020 11:44:59	AM
106800	1	AUTRES RESERVES	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
109000	1	ACTIONNAIRE CAPITAL SOUSCRIT NON AP	3	Résultat		AXPROD	5/22/2020 11:44:59	AM
110000	1	REPORT A NOUVEAU BENEFICE	3	Résultat	1	AXPROD	5/22/2020 11:44:59	AM
119000	1	REPORT A NOUVEAU PERTE	3	Résultat		AXPROD	5/22/2020 11:44:59	
120000	1	RESULTAT EXERCICE BENEFICE	3	Résultat		AXPROD	5/22/2020 11:44:59	
121000	1	RESULTAT PROVISOIRE	3	Résultat		AXPROD	5/22/2020 11:44:59	
129000	1	RESULTAT EXERCICE PERTE	3	Résultat	1	AXPROD	5/22/2020 11:44:59	
129100	1	ACOMPTES SUR DIVIDENDES EN ATTENTE D'AFFECTATION	3	Résultat		AXPROD	5/22/2020 11:44:59	
13100000	1	Réserve légale	3	Résultat	1	AXPROD	5/22/2020 11:44:59	
13810000	1	Autres réserves disponibles	3	Résultat	1	AXPROD	5/22/2020 11:44:59	_
13821000	1	Réserve pour l'impôt sur la fortune	3	Résultat	1	AXPROD	5/22/2020 11:44:59	
14110000	1	Résultats reportés en instance d'affectation	3	Résultat	1	AXPROD	5/22/2020 11:44:59	_
14120000	1	Résultats reportés (affectés)	3	Résultat		AXPROD	5/22/2020 11:44:59	
145000	1	AMORTISSEMENTS DEROGATOIRES	3	Résultat		AXPROD	5/22/2020 11:44:59	
151100	1	PROVISION POUR LITIGES	3	Résultat	1	AXPROD	5/22/2020 11:44:59	
151110	1	LABOR TERMINATION COSTS	3	Résultat		AXPROD	5/22/2020 11:44:59	_
151120	1	SOC - Divers provisions litige	3	Résultat		AXPROD	5/22/2020 11:44:59	
121120	1	SOC - Divers provisions litige	3	Résultat		AXPROD	5/22/2020 11:44:59	_
151130								

It is possible to define the number of rows to be displayed in the preview window, but also to make a quick and simple export of data in CSV format (useful for making quick comparisons of data)

In addition, by clicking "With Data Analyzer", you can see the details of each field from the Source. Useful for understanding how Fuzible interprets data types.

Data Preview				Data Successfully Loaded. 🛛 🕒 With Data Ar					
Source Data	Target Data	To Inser	t To Update To Delete						
DataTable Name : AX8 Namespace : A Row Count : 2502 Preview : 500 Fields Count : 8	PROD								
id_comptecomptable	id_racine_compted	omptable	li_comptecomptable	id_type	li_type	SYNCHRO_TAG	DBNAME	DTLOAD	Γ
101000	1	Name	: id_racine_comptecomptable	•		•	PROD	5/19/2020 7:22:03 PM	t.
10100000	1		ginal Definition : LEFT(A.MAINACCOUNTID,1	) as id rad	ine cor	notecomotabl	PROD	5/19/2020 7:22:03 PM	
101100	1	-	. Type : INT	/ 03 10_100	.inc_cor	nprecomptabl	PROD	5/19/2020 7:22:03 PM	Γ
104100	1		a Type : Int64				PROD	5/19/2020 7:22:03 PM	
104200	1		w Null : False				PROD	5/19/2020 7:22:03 PM	Ľ
104300	1						PROD	5/19/2020 7:22:03 PM	Γ
106100	1		nique : False				PROD	5/19/2020 7:22:03 PM	Г
106110	1	- Prin	nary Key : [id_comptecomptable]				PROD	5/19/2020 7:22:03 PM	Γ
106200	1	]					PROD	5/19/2020 7:22:03 PM	Г
106300	4		RESERVES CONTRACTUELLES	3	Résultat	1.	AXPROD	5/19/2020 7:22:03 PM	-

Note that each sub-query, each cross-query appear in the contextual menu and the data from each of them, loaded independently of the rest. In the example below, the Source query contains a sub-query, a UNION, and an cross-query.



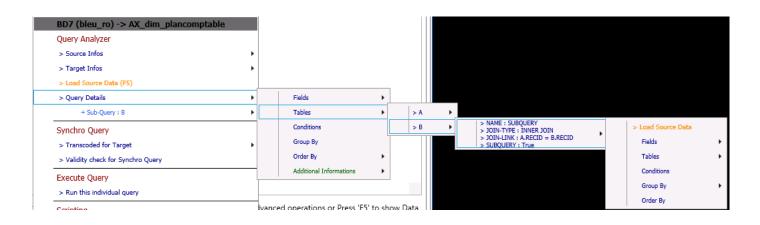
#### **QUERY ANALYZER - Query Details**

Allows you to fully deconstruct a query to verify that it is compliant, and that it has no syntax errors.

#### For example, you can go through all the fields...

BD7 (bleu_ro) -> AX_dim_plancomp Query Analyzer	aubic.					
> Source Infos	•					
> Target Infos	•					
> Load Source Data (F5)						
> Query Details	•	Fields	•	> id_comptecomptable	÷	> RAW : (A.MAINACCOUNTID) as id_comptecomptable > FIELD : (A.MAINACCOUNTID)
+ Sub-Query : B	+	Tables	•	> id_racine_comptecomptable	•	> TABLE : MAINACCOUNT > TABLE ALIAS : A
Synchro Query		Conditions		> li_comptecomptable	_ • h	> INDEX : 0
> Transcoded for Target	•	Group By		> id_type	- +	
> Validity check for Synchro Query		Order By	•	> li_type	- +	
Execute Query		Additional Informations	•			

### All tables and understand their joins...

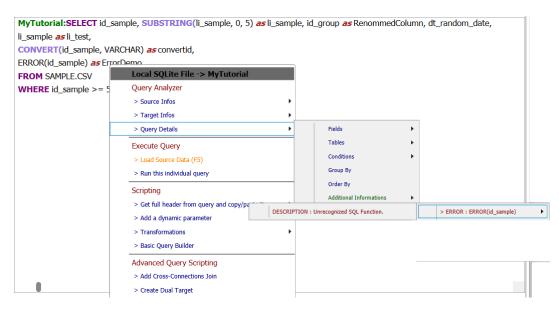


## *View sub-queries and try them...*

BD7 (bleu_ro) -> AX_dim_plancompta	ble	
Query Analyzer		
> Source Infos	•	
> Target Infos	•	
> Load Source Data (F5)		
> Query Details	•	
+ Sub-Query : B	> Load Source Data	
Synchro Query	Fields	
> Transcoded for Target	Tables	
> Validity check for Synchro Query	Conditions	
Execute Query > Run this individual query	Group By Order By	

### Check syntax errors...

#### In this example, the query contains an unknown transformation: ERROR(id\_sample)



The syntax errors detection is especially useful for "non-SGBD" queries, which allows you, if the query fails, to understand why it didn't work.

In the case of a query on a SGBD, the detection is essentially informative, as Fuzible does not know all the twisted cases that a query may contain.

## **EXECUTE QUERY - Run this individual Query**

You can only execute a specific query rather than the entire Job. By choosing this option, only the query on which you are positioned will be executed.

In this mode, the "LOG" tab does not fill up and the graphical interface is "blocked" for the time of execution. Once the processing is done, a LOG screen appears and shows the result.

## **SYNCHRO-QUERY - Transcoded for Target**

When you have written a query for a synchro. job, you may want to test how well the Source query transcoding is working on the Target. This menu lets you see the query as it will be performed in the Target.



## SYNCHRO-QUERY - Validity check for Synchro Query

This option simply checks the validity of the synchro. query.

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

It can be tedious to manually enter the entire header of an SQL table or file (if you want, for example, not to do a SELECT \* but a SELECT with the name of the fields). By clicking here, Fuzible will retrieve all the fields from the Source and if it has joins, you can choose from which table/file/webservice... you want the header back:



Once the header is retrieved, the software copies the header in the clipboard, you are then asked to paste it in your query:

mytargettable:select id\_sample,li\_sample,dt\_random\_date,nb\_random\_number,li\_random\_string,id\_group,id\_ssgroup
from sample1

#### **SCRIPTING - Transformations**

In case your Source connection is not a database, the Transformations menu reminds you of all the SQL syntax available to manipulate the data. This is obviously also available during writing : the assistant makes suggestions based on what you type.

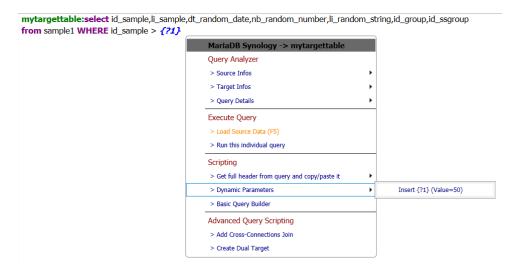
		CASE field WHEN 'value1' THEN 'value2' () ELSE 'value3' END
		COALESCE(field, 'replacementValue')
SHS Fuzible Data Re	plicator. Synchronizer	CHARINDEX(field, 'expToFind')
ile Configuration To		LENGTH(field)
Job Selection		CONCAT(field1, field2,)
		CONVERT(field, SQL type)
GUIZM	[10] Demo Job 001	DISTINCT (ex : SELECT DISTINCT * FROM [])
		TOP (ex : SELECT TOP 100 * FROM)
lob Configuration S	ource : CSV File Target : SQLite Database Que	LIMIT (ex : SELECT * FROM myTable LIMIT 100)
Job Queries Sand	20X	AVG(field)
		SUM(field)
A Query must start with the t ex : MyTable:SELECT ()	able name in which data will be written in (will be created if non-existe	MIN(field)
MyTutorial:SELEC	CT id_sample, SUBSTRING(li_sample, 0, 5) as	MAX(field)
li_sample <i>as</i> li_test	,	COUNT(field)
CONVERT(id_same		GROUP BY (ex : SELECT id_client, li_client, SUM(nb_amount) FROM [] GROUP BY id_client, li_client)
FROM SAMPLE.	Local SQLite File -> MyTutorial	ISNULL(field, 'replacementValue')
WHERE id_samp	Query Analyzer	INNER JOIN
	> Source Infos	LEFT JOIN
	> Target Infos	RIGHT JOIN
	> Query Details	NIOL
	Execute Query	OUTER JOIN
	> Load Source Data (F5)	LPAD(field, paddedLength, 'padString')
	> Run this individual query	LTRIM(field)
	Scripting	LOWER(field)
	> Get full header from query and copy/paste it	ORDER BY (ex : SELECT * FROM [] ORDER BY field ASC
	> Add a dynamic parameter	RPAD(field, paddedLength, 'padString')
	> Transformations	RTRIM(field)
	> Basic Query Builder	REPLACE(field, 'ValueToReplace', 'ReplacementValue')
	Advanced Query Scripting	SUBSTRING(field, startIndex, length)
	> Add Cross-Connections Join	UPPER(field)
	> Create Dual Target	WHERE (ex : SELECT * FROM [] WHERE field = 'value')
	-	AS (ex : SELECT id_client AS myClient [])
Right-clicl	c on a query for advanced operations or press 'F5' to	UNION (ex : SELECT field FROM table1 UNION SELECT field FROM table2)
		* (ex : SELECT * FROM [])

#### **SCRIPTING - Add a Dynamic Parameter**

You can add a Dynamic Parameter to your query. For example, I want to make the "WHERE id\_sample" filter dynamic :

mytargettable:select id_sample,li_sa	ample,dt_random_date,nb_random_number	,li_rando	m_string,id_group,id_ssgro	up
from sample1 WHERE id_sample >				
	MariaDB Synology -> mytargettable			
	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			
	> Basic Query Builder			
	Advanced Query Scripting			
	> Add Cross-Connections Join			
	> Create Dual Target			

#### After entering the desired value, Fuzible will add the dynamic setting in the query:



#### ... then add the parameter in the "Job Parameters" menu :

Dynamic Parameters	50/		?
	ilable command. Each parameter must be separated by a semicolon ( ; eries, text fields, connections) by referencing them like this : {?1}, {?2}	) View Job With Replaced Values	
			~

#### If Dynamic Parameters have already been set, the menu will be as follows:

<pre>mytargettable:select id_sample,li_sample,dt_r from sample1 WHERE id_sample &gt; {?1}</pre>	andom_date,nb_random_number,li_rand	om_st	ring,id_group,id_ssgroup
	MariaDB Synology -> mytargettable		
	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	•	
	> Dynamic Parameters	÷	Insert {?1} (Value=50)
	> Basic Query Builder		
	Advanced Query Scripting		
	> Add Cross-Connections Join		
	> Create Dual Target		

## **SCRIPTING - Basic Query Builder**

A simple assistant to create a query. It is understood that this mode does not allow for advanced queries.

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

This is where you can open the menu for a cross-query.

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

This is where you can open the menu for a multi-target query.

# Log Viewer tab

Depending on the level of LOG chosen in the "Job Configuration" tab, you will see more or less detail in the Log Viewer.

What you see is systematically referred to log files that are produced by the software.

At the end of a Job, a message will appear on the screen indicating its status.

A round-up of debug possibilities:

- The Query Analyzer and the "Show Source data" screen (contextual query menu)
- The "SIMULATION" mode, which executes the Job without performing any operation in the Target, it merely displays everything it will do there (see below)
- The general LOG: quite verbose, it can help you understand a problem during Job execution, you can also set it up in "Detailed" to get as much information as possible.

Check out the files produced by the Job, you have 3:

- 1) The LOG file, which shows everything you see in the "LOG Viewer" tab
- 2) The "QUERIES" file that shows all the queries that have failed.
- 3) The "DEBUG" file that provides a higher level of information from any error.

b Selection ——			Create/manage	a Multi-steps Job ————
UIZM	[10] Demo Job 001		~	Create new Step
b Configuration	Source : CSV File Targe	et : SQLite Database Queries Lo	og Viewer	
15:19:46][INF][SRC][M 15:19:46][INF][TRG][M 15:19:46][INF][TRG][SC 15:19:47][INF][PRG][Lc	Thread.RepSyncTask_Source] T01 Thread.RepSyncTask_Target] T01 - QLTools.InsertDataInBDD] Insert da IgTools.EndJobLog] RUNNING TIN	JOB STARTED : Demo Job 001 (1/0) > Getting Source Data (SAMPLE.CSV) Target A -> Replication (51 Rows) To MyTutot ta from dataset on MyTutoria(51) E: 00:00:00 - ERRORS : 0 - WARNINGS : 0 OB FINISHED (NO ERRORS) : Demo Job 001	ial	
		STATUS : RanToCompletion -> Destination : MyTutorial -> Rows Source : 51 -> Rows Target : 0 -> Inserted : 51 -> Updated : 0 -> Deleted : 0 -> Processing Date : 19/01/202	× 1 15:19:46	
			ОК	
View Threads				

## Example of a successful execution:

## Example of a failed execution; THE LOG is displayed in bold:

b Selection ——				Create/	manage a Multi-steps J	ob
JIZM	[21] MongoDB -> CSV : Ea	irly Test		$\sim$		Create new Step
b Configuration	Source : MongoDB Database	Target : CSV File	Queries	Log Viewer		
15:21:36][INF][SRC] 15:21:41][ERR][SRC] 15:21:41][INF][SRC] 15:21:42][INF][SRC] 15:21:42][INF][TRG] 15:21:42][INF][TRG] 15:21:42][INF][PRG]	[LogTools.StartJobLog] 19/01/2021 - [INIProgram.ExecutePrePostJobOper [NOSQLTools.ExecuteCommand] Com [MTnerad.RepSyncTask, Source] T01 - [MTnerad.RepSyncTask, Target] T01 - [MTnerad.RepSyncTask, Target] T01 - [LogTools.EndJobLog] RUNNING TIM [LogTools.EndJobLog] 19/01/2021 - J	ttions] Executing Pre-Jo mand drop failed: ns no Getting Source Data ( Target A -> Replication Target A -> Replication E : 00:00:05 - ERRORS :	b Command(abc) to found.(Com fuziblebson) (100 Rows) Tr (100 Rows) Tr (100 Rows) Tr (100 Rows) Thomas (100 Rows) Thomas	) in Source mand drop failed: ns na mongomulti.csv mongomulti_Data.CSV 5:0 oDB -> CSV : Early Test X		
View Threads						

## **Running a Job**

Once the Job is set up, saved, and tested, you can execute it and interrupt it if necessary.

These buttons are at the bottom of the interface. Also, you will find that on the "Job Configuration" tab, you have a little "Simulation Mode" button just above. This will write in the LOG any "write" operation that is going to be executed on the Target, without performing it.

This LOG has the advantage of not actually executing the Job, so not to take any risks and possibly compromise the Target.

In addition, this mode writes all SQL queries, which can be very useful in the case of replication of SGBD data to SGBD: allows you to get all SQL code: "INSERT," "DELETE," "UPDATE" statements.



## Example of "Simulation" output (Job that copies data from a file to a database)

🗐 n	
1	SINULATION MODE WILL FOST-WORK INFUT FILE(S) -RIEN: C:(Vesrs\quizm\source\Workspaces/Fuzible\Fuzible\bin\x64\Debuq\FILES\SAMPLE.CSV
2	SIMULATION MODE) WILL POST-WORK INPUT FILE(S) ON (S) FFP - 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."table", il."to" FROM sqlite_master AS m INNER JOIN pragma_foreign_key_list('h
6	[SIMULATION MODE] SELECT il.name, il.type, '0', 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","li_test")
11	

Also, if you check the associated LOG "Queries" file, you will get all the queries in plain sight, including INSERT. This scenario is especially useful if you've installed the app locally, and you want to send data to a locally inaccessible database.

You can only generate queries through the "simulation" mode, and then connect to the remote server to integrate the data via INSERT code Fuzible produced.

# "Service" Application

The software comes with a background service application: This service works in harmony with the "Client" application (using SQL mode), as well as the Job Orchestrator.

The app automatically creates and purges the SQL table that is used for its operation. As a "console" application, its LOG is written in the « SERVICE\_YYYYMMDD\_LOG.TXT » file.

The application keeps only one file, it systematically erases the one from the day before.

It retrieves the list of Jobs invoked (by the "Client" application or by the Planification) and executes them one after the other (it can run several in parallel, this setting being managed in the main application, configuration menu)

## Setting up the Windows Task Manager

Note: This setting can be done automatically by Fuzible's configuration menu. However, you may need to manually edit/create it on somewhat tricky points, such as the execution account.

Modifier le déclencheur X	🕒 Propriétés de Replicator Service (Ordinateur local)
Lancer la tâche : À l'heure programmée 🗸 🗸	Général Déclencheurs Actions Conditions Paramètres Historique
Paramètres O Une fois Démarrer: 18/04/2018  T 07:00:00 Synch. fuseaux horaires	Spécifiez les conditions qui, avec l'élément déclencheur, détermineront si la tâche doit s'exécuter. Elle ne s'exécutera pas si l'une de ces conditions n'est pas vérifiée. Inactivité
Chaque jour     Chaque semaine     Chaque mois	Démarrer la tâche si l'ordinateur est inactif pendant :     Attendre l'inactivité pendant :     Attendre l'inactivité pendant :     Attendre l'inactivité pendant :     Attendre l'inactif explus inactif     Redémarrer si l'état inactif recommence Alimentation     Ne démarrer la tâche que si l'ordinateur est relié au secteur
Paramètres avancés  Report maximal de la tâche (aléatoire) : 1 heure  Répétre la tâche toutes les : 1 minute  pour une durée de : 1 jour  Arrêter toutes les tâches à l'issue de la durée de répétition Arrêter la tâche si elle s'exécute plus de : 4 heures  Expiration : 19/05/2021  20:42:21  Activée	Ve demarrer la carre que s'i l'ordinateur est reire au secteur Arrêter si l'ordinateur passe en alimentation par batterie Sortir l'ordinateur du mode veille pour exécuter cette tâche Réseau Ne démarrer que si la connexion réseau suivante est disponible : N'importe quelle connexion
OK Annuler	OK Annuler
Modifier une action       ×         Vous devez spécifier l'action que cette tâche effectuera.       Action : Démarrer un programme         Paramètres       Programme/script :         C\Tools\FuzibleService.exe       Pargourir         Ajouter des arguments (facultatif) :       C\Tools\         Commencer dans (facultatif) :       C\Tools\	Propriétés de Replicator Service (Ordinateur local) Général Déclencheurs Actions Conditions Paramètres Historique Spécifiez d'autres paramètres influant sur le comportement de la tâche. Autoriser l'exécution de la tâche à la demande Exécuter la tâche dès que possible si un démarrage planifié est manqué Si la tâche échoue, recommencer tous les : Iminu ~ Tenter de recommencer jusqu'à : 3 fois Arrêter la tâche si elle s'exécute plus de : 4 heure ~ Si la tâche en cours ne se termine pas sur demande, forcer son arrêt Si la tâche s'exécute déjà, la règle suivante s'applique : Exécuter une nouvelle instance en paral ~
OK Annuler	OK Annuler

With each start, the app checks the stack of requested Jobs to run and:

- Checks if the number of Jobs being processed does not exceed the max. value from the settings.
- Sorts out the list of Jobs to be launched according to the priority assigned to it (between 1 and 3)
- Executes the requested Job(s)
- Follows the progress of the job and get its output ; updates the SQL table accordingly so that the user, from the "Client" application, can see the progress (by clicking "Job Status")

The service must be installed on the same machine and path as the Fuzible application.

## Setting up an external job (excluding Fuzible) with the "Service" app

The idea of this option allows you to take advantage of the "client/service" system to perform any other task.

This is quite feasible, and simply requires you to manually enter data into the SQL table "client\_jobs" (which is used by the "Service" application and which is located on its SQL instance) the external Job information that one wishes to be able to trigger (basically, the execution of a BAT file performing certain operations is well-advised)

Field	Description	
User_jobs	Use any username (for example, the person who creates this job)	
Job_id	A Job number, for example "001"	
Job_name	b_name Job name: this is what will be displayed to the user	
Job_description	Description: A few more words to describe the Job	
Job_params	Default Dynamic Parameters (optional)	
Job_queries	You can maybe write a more in-depth description of the Job ?	
Job_haschildren	0	
Job_password	The password that allows the user to launch the Job. Fuzible passwords are encrypted, but for those external commands, you have to enter it "as it" in the table	
Job_priority	Execution priority (1,2,3)	
Application_name	The name of the app to be launched (ex : c:\Tools\MyFile.bat)	
Job_category	Job categorization to optimize user view	

For the user, this is the representation of an external Job in the list of Jobs that will be proposed to him in the "Client" application:

[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
REFUSHARE -> PANAMA : Import RUBPAI RhPI

#### And here is the database representation:

	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	202001	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	3sdigA8JNku0	1	ReplicatorApp.exe	File -> Database
3	SVCSCO	[15]	BLEUSHARE -> BLEU : Véhicules : Imp	Traitement des fic	202004	IMPORT_LEASEPLAN:s	0	y/9i5nSoe2Dy	1	ReplicatorApp.exe	File -> Database
4	SVCSCO	[16]	BLEUSHARE -> BLEU : Véhicules : Imp	Traitement des fic	20200331	IMPORT_TOTAL_SERIS	0	i+CKpDuXSFM	1	ReplicatorApp.exe	File -> Database
5	SVCSCO	[53]	DWH -> FICHIER : Extraction du Grand	Extraction du Gra	>=;%^^^%MM<	v_grand_livre_comptable	0	lyMQTIDiKO+u	1	ReplicatorApp.exe	Database -> File
6	SVCSCO	[62]	MOSAIC -> BLEUSHARE : Extraction d	Réalise l'extractio	20200515	INT_acomptes.CSV:selec	0	57P6RHhtq8X	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	4WnbSiyvugdz	1	ReplicatorApp.exe	File -> Database
9	SVCSCO	[72]	MOSAIC -> RHPI : Webservice Modifier	Exécution des We	id_matricule	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.{?2}	0	HNq7kAZtyKn	1	ReplicatorApp.exe	Database -> Webservice
11	SVCSCO	[81]	MOSAIC -> RHPI : Préparation aux modi	Extractions à partir		01_CONTRAT_MODIFIE	0	07JbyuFDrblq1	1	ReplicatorApp.exe	Database -> File
12	SVCSCO	[85]	MOSAIC-PP -> RHPI : Webservice Prim	Exécution du We		program=SPS010RB:SEL	0	pmx5BF4cXE6	1	ReplicatorApp.exe	Database -> Webservice

# "Client" Application

A lightweight client is provided and allows you to run jobs remotely from his own workstation.

The value of making the ability to trigger a Job by an user is crucial, for example:

- Let users choose the date/time to launch a Job
- Produce reports on the fly

#### Pre-requisite on client workstation:

- ➢ Windows OS (7+)
- > .NET Core 3.1
- > INI file "CLIENTAPP. INI" (available in Fuzible installation path)

Warning: If you change the Client/Service connection string in the app's general settings, you'll need to provide the new "CLIENTAPP. INI" to users because this file contains the connection settings!

USER PC Fuzible Light App Gets Lift of Available Jobs Last Job Invoke a Job - Immediate or delayed Invoke - Job Status - Execution Plantification - Flood Control - Flood Control - Flood Control - Fuzible Service - Fuzible Service

Here's a diagram of how « Client » App works :

It communicates with the Service through a database that is configured from the Main Application.

- Reads the Execution Queue Executes Jobs (from Light App or Planification) - Writes Status Log The light client app uses a "stack" system. When choosing a job to execute, the app writes an SQL row in the "Service" app database instance. The settings of this connection are present in "CLIENTAPP. INI" and are encrypted for security reasons.

The "Client" software does not execute the Job. It loads it into a stack; it's the "Service" app provided that controls and launches the jobs that are invoked.

A Job must be configured beforehand as visible in the "Client" application:

ob Selection				Create/manage a Mult	ti-steps Job	<u>e</u>
GUIZM	[9] Accounting Data To Da	atawarehouse	~	Prev. Step 1 > Accounti	ing Data Tc 🗸 🗸	Next Step
ob Configuration Sc	ource : SQLite Database	Target : XML File Que	ies Log Viewer			
Job Decription						
Creation Date : 05/10/2020 13:16:28 Last Modified : 19/11/2020 16:31:35 Last Execution : 05/10/2020 13:28:22	- RUNNING TIME : 00:00:03 - ERRORS :	0 - WARNINGS : 0	Rename Job Delete Job	Change Passwo Planification		Create New Job
Main Parameters ———						
	Data Replication	~				
	target without any comparison					
Vill copy a source data into a Dynamic Parameters ou can write any text or any a				Vie	w Job With I	Replaced Values
Dynamic Parameters You can write any text or any ay You can use them anywhere (r	target without any comparison wailable command. Each parame queries, text fields, connections)	by referencing them like this : (	1), {?2]	Vie	w Job With I	Replaced Values
Will copy a source data into a Dynamic Parameters You can write any text or any a You can use them anywhere (r	available command. Each parame queries, text fields, connections) Errors + Informations	by referencing them like this : (	g in SQL	Vie te e-mail adress(es) here (se		Replaced Values
Vill copy a source data into a Dynamic Parameters Ou can write any text or any a You can use them anywhere (o OG Level God Level Send Mail When Finished	available command. Each parame queries, text fields, connections) Errors + Informations	by referencing them like this : (	g in SQL			
Vill copy a source data into a Dynamic Parameters Ou can write any text or any a You can use them anywhere (o OG Level God Level Send Mail When Finished	target without any comparison available command. Each parame queries, text fields, connections) Errors + Informations	by referencing them like this : (	g in SQL	te e-mail adress(es) here (si	eparated by a	semicolon)
Vill copy a source data into a Dynamic Parameters ou can write any text or any a You can use them anywhere ( og OG Level Send Mail When Finished	arget without any comparison available command. Each parame queries, text fields, connections) Errors + Informations	v La	g in SQL get) if Job has Errors	te e-mail adress(es) here (si	eparated by a	semicolon)

By opening the "Client"app, you can choose one of the Jobs available from the list. Its password will be required; The person who creates the Job will have had the presence of mind to provide it to the person (or persons) granted to launch the Job.

The list of available Jobs:

#### Fuzible Remote Job Launcher

- 🗆 X

Fuzible Remote Job Launcher

List of Jobs :		Request
Dvnamic Paramete	[DATABASE -> DATABASE] Fuzible (GUIZM)	
synamic rorunete	MySQL -> Postgres : Demo Synchro	
Scheduled Execution	(DATABASE -> FILE)	
nformations	Fuzible (GUIZM)	
	Accounting Data To Datawarehouse	
	[FILE -> FILE]	
	Fuzible (GUIZM)	
	CSV -> XLS : This is a sample Job	
	[MAILBOX -> FILE]	
	Fuzible (GUIZM)	
	WS -> SQLITE : Démo [MAILBOX -> MAILBOX]	
	Fuzible (GUIZM)	
	Gmail-> Hotmail : Transfert Mail	

The person handling the client application can change the dynamic parameters of the selected Job: very useful to produce, for example, a period-specific Reporting, setting a filename to import, etc. everything is possible, it all depends on how the Job is configured!

Dynamic Parameters	>= <mark>;</mark> %YYYY%MM	'<1M	
Scheduled Execution	20/05/2020	15 15	н

It is also possible to delay the launch of the Job, by default, by clicking "Request", the Job will be stripped as quickly as possible by the "Service" application, but it can also be set to a particular date/time (for example to launch "heavy" tasks in the middle of the night without having to wake up in the middle of the night to trigger the launch)

"View Queries" allows you to see the queries associated with the Job, and their translation with dynamic settings. Rather reserved for users with some knowledge of SQL!

After clicking "Request", the software shows the position in the stack:

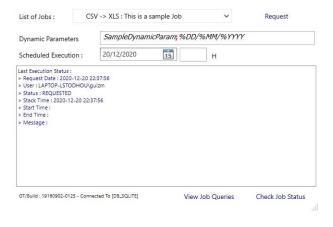
ist of Jobs : C	V -> XLS : This is a sample Job	Request
Dynamic Parameters	SampleDynamicParam;%DD/%MM/%YYYY	
Scheduled Execution :	20/12/2020 15 H	
	[INF] Job Stacked in Positi	on 1
	O	<

The user can follow the status by clicking on "Check Job Status."

The Job has been invoked but not yet handled by the "Service" application:

😴 Fuzible Remote Job Launcher 🦳 🗆 🗙

Fuzible Remote Job Launcher



#### Job executed and completed:

Fuzible Remote . Fuzible Remote Job La					
List of Jobs :	CSV -> XLS : This is a samp	ole Job	~		Request
Dynamic Parameters	SampleDynamicPa	ram <mark>;</mark> %DL	D/%MM/%YYYY		
Scheduled Execution :	20/12/2020	15	н		
[INF] WITH DYNAMIC PAR. [INF] T01 -> Getting Sourc [INF] T01 - Target A -> Rep [INF] T01 -> Getting Sourc [INF] T01 - Target A -> Rep [INF] T01 -> Getting Sourc	22:37:56 22:45:50 2:42:02 RTED : CSV -> XLS : This is a sam AMETRES : SampleDynamicParame Data (SAMPE.CSV) ilication (100 Rows) To SAMPLE_C Data (SAMPE.CSV) ilication (100 Rows) To SAMPLE_C	96DD/96MM, 0UTPUT_SELE 0UTPUT_SELE	/%YYYYY CTALL.XLSX		
GT/Build : 19160902-0125 - 0	connected To [DB_SQLITE]	Vie	w Job Queries	C	Check Job Status

Note: If the user clicks "Check Job Status" before requesting the execution, he will see the LOG of the last run to date, if it has not yet been purged (the retention time is defined in the general parameters of the software)

# **Fuzible SQL: Glossary**

## **Supported functions**

Here is a list of the SQL functions supported by Fuzible's engine for querying a non-SQL Source. For use, internet is your friend (SQL standard). The Query Assistant will also show you how to use them.

	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	Randomizes values to simulate an « anonymization » feature
	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
	(rappel : 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
Math. Operations in functions	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
Sub-queries	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)

# **Unsupported**

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 » or « 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
	<pre>Ex : SELECT * FROM monfichier WHERE annee &gt; {%YYYY}</pre>
Fields as sub-queries	Ex : SELECT (select id FROM monfichier.csv) as id FROM monautrefichier.csv
Math. operations on aggregated data	Ex : SELECT COUNT(*) + 10 FROM monfichier