Data Lister manual

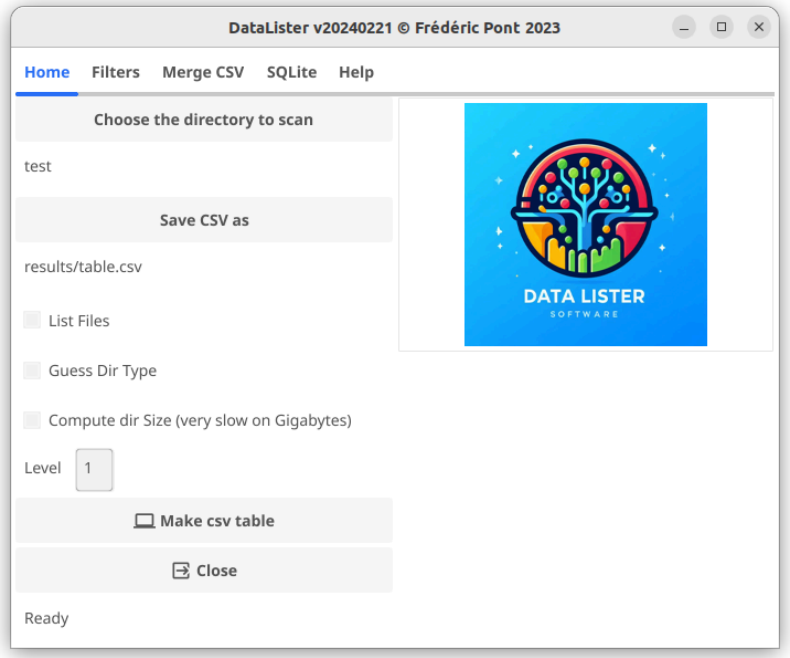
Version 1.0, 2024-02-23.
Frédéric Pont

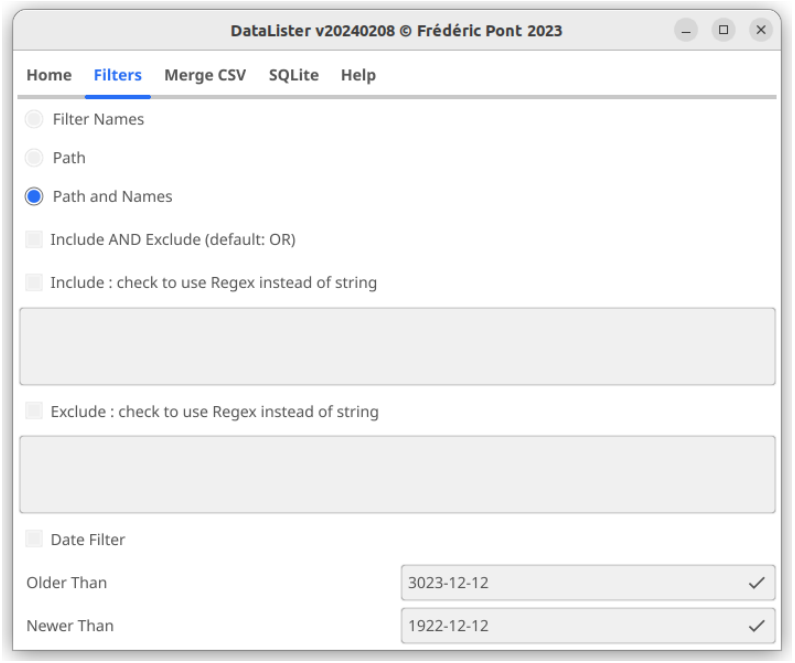
Index

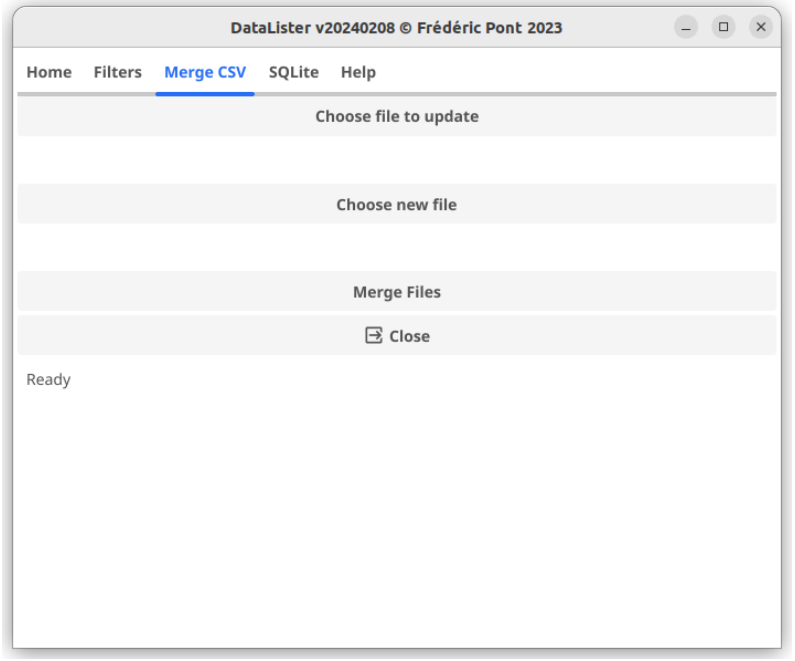
1. DATA Lister presentation

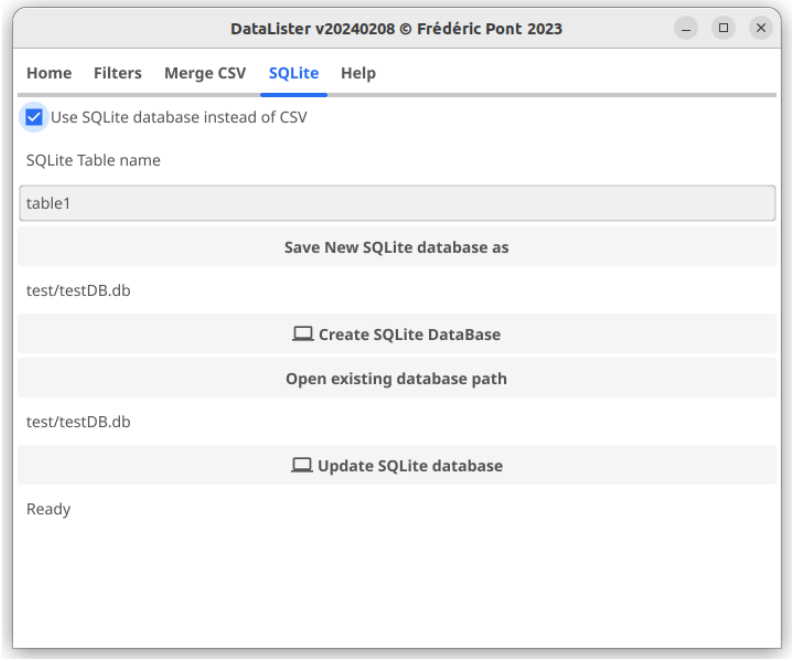
DATA Lister is a software to list relevant directories/files from a file system in a table for data management.

1.1. ScreenShots




DataLister v20240208 © Frédéric Pont 2023
Home
Filters
Merge CSV
SQLite
Help
Filter Names
Path
Path and Names
Include AND Exclude (default: OR)
Include : check to use Regex instead of string
Exclude : check to use Regex instead of string
Date Filter
Older Than
3023-12-12
Newer Than
1922-12-12


DataLister v20240208 © Frédéric Pont 2023
Home
Filters
Merge CSV
SQLite
Help
Choose file to update
Choose new file
Merge Files
Close
Ready



#	Path	Name	Size	LastAccessDate	DirType	TypeScore	SampleType	Project_ID	RelatedProject	content	Delete_Date
0	test/sequences	sequences	0	2023-11-30	BCL2	1	Cells	Project_1	Project_2	MyExperiments	2028-01-01
1	test/images1	images1	0	2023-11-30	VideoMicroscope	1	Cells	Project_1	Project_2	MyExperiments	2028-01-01
2	test/analyses	analyses	0	2023-11-30	Cytometry	1	Cells	Project_1	Project_2	MyExperiments	2028-01-01
3	test/sequences_2	sequences_2	0	2023-11-30	Fasta	1	Cells	Project_1	Project_2	MyExperiments	2028-01-01
4	test/level0	level0	0	2023-12-1			Cells	Project_1	Project_2	MyExperiments	2028-01-01
5	test/level0/file1	file1	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
6	test/level0/file1/sofware	sofware	0	2023-12-1	Software	1	Cells	Project_1	Project_2	MyExperiments	2028-01-01
7	test/level0/dir1	dir1	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
8	test/level0/dir1/dir2	dir2	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
9	test/level0/dir1/dir2/dir3	dir3	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
10	test/level0/level1	level1	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
11	test/level0/level1/level2	level2	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
12	test/level0/level1/level2/level3	level3	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
13	test/dir1	dir1	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
14	test/dir1/dir2	dir2	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
15	test/dir1/dir2/dir3	dir3	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
16	test/dir1/dir2/dir3/dir4	dir4	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
17	test/level1	level1	0	2023-12-1			Cells	Project_1	Project_2	MyExperiments	2028-01-01
18	test/level1/level2	level2	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01
19	test/level1/level2/level3	level3	0	2023-12-1			Cells	Project_1	Project_2	MyExperiments	2028-01-01
20	test/level1/level2/level3/level4	level4	0	2023-11-30			Cells	Project_1	Project_2	MyExperiments	2028-01-01

Caution : set the directory level to a reasonable value before starting DATA Lister on a large file system to avoid producing a huge result table.

2. Key characteristics

- List directories and optionally files.
- TSV table output/update
- SQLite output/update
- Tunable dir level
- Try to guess dir content based on footprint (level must be set to dir +1 to allow the dir content analysis)
- Unlimited number of customizable dir footprint
- Filters by name, path, name and Path
- Include/Exclude filter list by unlimited number of string or regex
- Filter by date
- Unlimited number of custom (pre-filed) columns
- Compute directory size (slow on terabytes of data because 100% of the files are parsed)
- Merge tool to update old table with the new rows from a new analysis

2.1. Installation

No installation required the code is statically compiled.

- Download the zip file from the "<>Code" green button and unzip it
- or `git clone https://github.com/FredPont/Data_Lister.git`

Start the software in a terminal (Linux/Mac) using the ./Linux_DataLister.bin command, Win_DataLister.exe (Windows) or double click on Win_DataLister.exe (Windows)

2.2. Quick start

2.2.1.1. CLI

- Edit config/settings.json to set root directory and options

```
{
    "InputDir": "test",
    "OutputCSVFile": "results/table.csv",
    "OutputSQLFile": "test/testDB.db",
    "ListFiles": false,
    "GuessDirType": false,
    "CalcSize": false,
    "Level": 3,
    "filterName": false,
    "filterPath": false,
    "filterPathName": true,
    "IncludeRegex": false,
    "Include": [
        ""
    ],
    "IncludeAndExclude": false,
    "ExcludeRegex": false,
    "Exclude": [
        ""
    ],
    "OlderThan": "3023-12-12",
    "NewerThan": "1922-12-12",
    "DateFilter": false,
    "UseSQLite": false,
    "SQLiteTable": "table1",
    "CompiledIncludeRegex": null,
    "CompiledExcludeRegex": null
}
```


Use absolute path in “InputDir”, “OutputCSVFile” or “OutputSQLFile”.

Note : for the command line version, backslashes must be escaped in regex in the settings.json file (this is not necessary in the GUI).

Example : to exclude names starting with a dot use “^\\..+”

- The filter priority is Date > Include > Exclude
- If more than one string/regex is given they are cumulated (reg1 OR reg2)
- If Include and Exclude are used simultaneously, they are cumulated (Include OR Exclude) if “Include AND Exclude” is not checked
- Edit config/DirSignatures.json to set the directory patterns (strings, no regex)


```
{
    "Software": {
                "content": [".go", ".git", ".DLL", ".dll", ".r", ".jl", ".pl"],
                "scoreThreshold": 0.2
    },
    "Fasta": {
            "content": [".fasta", ".FASTA", ".fasta.gz"],
            "scoreThreshold": 0.8
    }
}
```


- Edit config/columns.tsv to add custom columns and their optional default values

```
ColumnName  DefaultValueswork in progress...
SampleType  Cells
Project_ID  Project_1
RelatedProject  Project_2
content MyExperiments
Delete_Date 2028-01-01
```

- Start the software using the precompiled binaries for Linux, Mac or Windows

```
Usage :
  -c  Start DataLister directories analysis in command line.
  -g  Start DataLister directories analysis in graphic mode.
  -m  Start DataLister merging tool.
  -i string
      New result file path. Only new files/dir are added to the old file
  -o string
      Old result file path.
  -s  Create a new SQLite database. Example : DataLister -s

  Examples :
```

```
Start the analysis of the directories in command line (-c):
./Linux_DataLister.bin -c

To add new data from newfile to oldfile :
./Linux_DataLister.bin -m -o oldfile.csv -i newfile.csv

if "UseSQLite": true in the config/settings.json file, then
./Linux_DataLister.bin -c
will update the SQLite database indicated in "OutputSQLFile"
```

2.2.2. GUI

By default the software start in GUI mode For a basic usage all the settings are in the "home" tab. Choose the directory to scan, the CSV output, the deepness level and click on the "Make csv table" button.

3. Making CSV tables

DATA Lister can list files/directories in a CSV table with a TAB separator

3.1. GUI

3.1.1. Home settings

In the "home" tab, choose the directory to scan, the CSV output, the deepness level and click on the "Make csv table" button.

Verify that the "Use SQLite database instead of CSV" is not checked in the SQLite tab. This button enable SQLite database update and disable CSV output.

To list files : click on the "List Files" checkbox

Data Lister can guess the directory type of directories at a level n, according to the config/DirSignatures.json file. The directory type can be guessed only if the deepness level is at least n+1 (the software need to explore the n+1 level to guess the type of a directory at level n) If "Guess Dir Type" is checked, the scan will stop at the level n for the identified directories

Data Lister can compute the directories sizes by checking the "Compute dir size" checkbox. To compute the size of a directory, the size of all the files inside this directory have to be computed. This is very time consumming. So it is not recommended to use this option on a large file system.

3.2. Filters

The filter priority is Date > Include > Exclude

It is possible to filter directories/files names, path, or both using an include/exclude list of strings or regular expressions. Include/exclude lists can contain an unlimited number of strings/regex on per line By default the include list is parsed first and then the exclude list. If a file satisfy the include list, it is preserved even if it is rejected by the exclude list. If the "Include AND Exclude" checkbox is checked, then a file is preserved only if it satisfy both lists

If the "Date filter" is checked, a file is preserved only if it satisfy both "Older Than" and "Newer Than" constraints.

3.3. CLI

Data Filter can be used in command line. All the settings are in the config/settings.json file.

Edit config/settings.json to set :

1. the directory to scan : "InputDir" (absolute path)
2. the CSV output file : "OutputCSVFile" (absolute path)
3. list files and directories : "ListFiles"
4. guess the directory type of directories : "GuessDirType"
5. compute the directories sizes : "CalcSize"
6. the deepness level : "Level"
7. filter by names, path or both : "filterName", "filterPath", "filterPathName" (must be set to "false" except one)
8. use regular expressions in include/exclude lists : "IncludeRegex", "ExcludeRegex"
9. use both include and exclude lists "IncludeAndExclude"
10. string/regex must be entered in the "Include" or "Exclude" arrays. Backslashes must be escaped in regex in the settings.json file. Example : to exclude names starting with a dot use "^\\..+" instead of "^\..+". Multiples string/regex must be separated by ",". Example :

```
"Include": [
        "string1",
```

```
            "string2"
        ],
```

11. date filters : “OlderThan”, “NewerThan”. Date format is “yyyy-mm-dd”
12. “UseSQLite” must be set to false

Example of config/settings.json file :

```
{
    "InputDir": "test",
    "OutputCSVFile": "results/table.csv",
    "OutputSQLFile": "test/testDB.db",
    "ListFiles": false,
    "GuessDirType": false,
    "CalcSize": false,
    "Level": 3,
    "filterName": false,
    "filterPath": false,
    "filterPathName": true,
    "IncludeRegex": false,
    "Include": [
        ""
    ],
    "IncludeAndExclude": false,
    "ExcludeRegex": false,
    "Exclude": [
        ""
    ],
    "OlderThan": "3023-12-12",
    "NewerThan": "1922-12-12",
    "DateFilter": false,
    "UseSQLite": false,
    "SQLiteTable": "table1",
    "CompiledIncludeRegex": null,
    "CompiledExcludeRegex": null
}
```

4. Update CSV tables

4.1. GUI

4.2. CLI

5. Working with SQLite database

Data Lister can create/update SQLite databases. In the “SQLite” tab, enter the table name.

5.1. GUI

5.1.1. Create a new database

1. Click on the “Save New SQLite database as” button and enter a file name, for example “MyDataBase.db”
2. Click on the “Create SQLite DataBase” button. A new database with a primary Key and an empty table will be build.

5.1.2. Update a database

1. Check the “Use SQLite database instead of CSV” checkbox. This button enable SQLite database update and disable CSV output.
2. Click on the “Open existing database path” and select the database file.
3. Click on the “Update SQLite DataBase” button. The selected database will be updated.

5.2. CLI

5.2.1. Create a new database

Edit config/settings.json to set :

1. the database file path : “OutputCSVFile” (absolute path)
2. the database table name : “SQLiteTable”
3. start the software with “-s” option : ./Linux_DataLister.bin -s

5.2.2. Update a database

Edit config/settings.json to set :

1. the database file path : “OutputCSVFile” (absolute path)
2. the database table name : “SQLiteTable”
3. enable the database update by setting “UseSQLite” to true
4. start the software with “-c” option : ./Linux_DataLister.bin -c

6. Tips

It is possible to extract relevant directories in a single scan using an include filter. To avoid listing non relevant directories :

1. the relevant directories should be placed at the same deepness level
2. the relevant directories should be placed in directories with a name that is not used else where. For example, if the relevant directories are in “RawData/” directories, then use an include filter by path set to “RawData”.