Foreign Language Text Reader (FLTR)

A Handy App For Language Learning

Current FLTR Version: **1.4.1 (2025-10-17)**
Documentation last changed on 2025-10-17
Website:
https://github.com/hapepo23/foreign-language-text-reader

Table of Contents

Abstract and Introduction

FLTR (= Foreign Language Text Reader) helps you to do both extensive and intensive reading as part of your foreign language acquisition in an easy and pleasant way.

While reading, you look up unknown words in dictionaries via a web browser or via locally installed dictionary applications, and save vocabulary terms (words and multi-word expressions) with translations, romanization (like Pinyin, Hiragana, etc., optional) and example sentence (optional). Each term has also a learning status (1/"Unknown" to 5/"Known", plus Statuses "Ignored" and "Well Known") with an associated color.

The saved words will automatically show up with all their data and status in all texts in the same language. They can be easily imported into flashcard software like **Anki** or **Mnemosyne** via the export function.

FLTR is in many areas similar to the Public Domain software **"Learning With Texts" (LWT)**. But FLTR is easier to install and run - no personal web server and no database is needed. **"LingQ.com"** offers a web service with a huge text library, tutoring services and a forum, and is also quite similar to FLTR, but much slower because it is a web service. You must be online to use LingQ, and it costs money ($10 per month). What's more, you are completely dependant on the availability of the LingQ service, and you don't own your vocab data. If you stop paying, you lose nearly everything. See also **"Compare Similar Software (FLTR, LWT, LingQ)"**.

FLTR is programmed in Java 1.8, also known as Java 8. To run FLTR, you need an installed Java Runtime Engine (JRE) from **http://java.com**. The program runs on macOS, Windows, and Unix/Linux. There is no need to be online while using FLTR, however you must be online if you use online dictionaries.

FLTR reads and writes its data (language definitions, texts, terms, export, general settings) in UTF-8 text or TAB-delimited CSV files, which makes it easy to add, change, or extract data with simple to use programs like a text editor or a spreadsheet program.

Motivation for this software

From the **Wikipedia article "Extensive reading":**

> *Extensive reading (or Free voluntary reading) is an aid to language learning, including foreign language learning, by means of a large amount of reading. The learner's view and review of unknown words in specific context will allow the learner to infer and thus learn those words' meanings. ... Extensive reading is contrasted with intensive reading, which is slow, careful reading of a small amount of difficult text – it is when one is "focused on the language rather than the text". Extensive and intensive reading are two approaches to language learning and instruction, and may be used concurrently; intensive reading is however the more common approach, and often the only one used.*

Stephen Krashen has published more than 350 papers and books, contributing to the fields of second language acquisition, bilingual education, and reading.

He promotes the use of "*free voluntary reading*" during second language acquisition, which he says "*is the most powerful tool we have in language education, first and second*".

Watch **this video**.

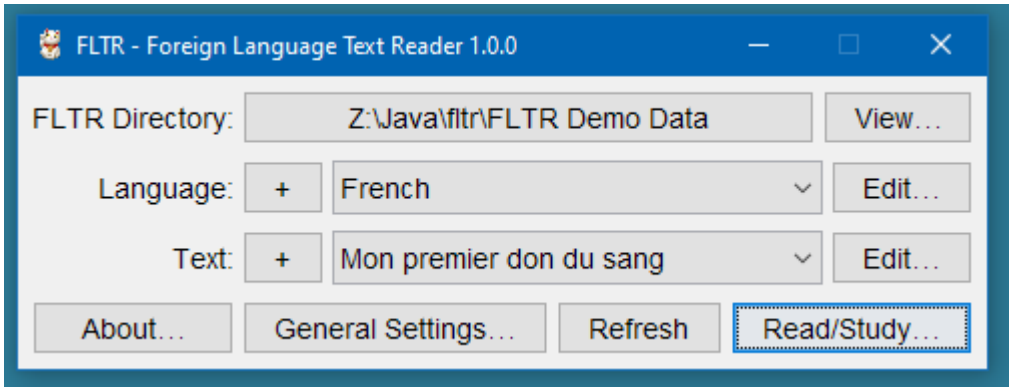
Steve Kaufmann has created more than 1,400 videos about the importance of reading (and listening) in language learning.

Watch **his videos on YouTube.**

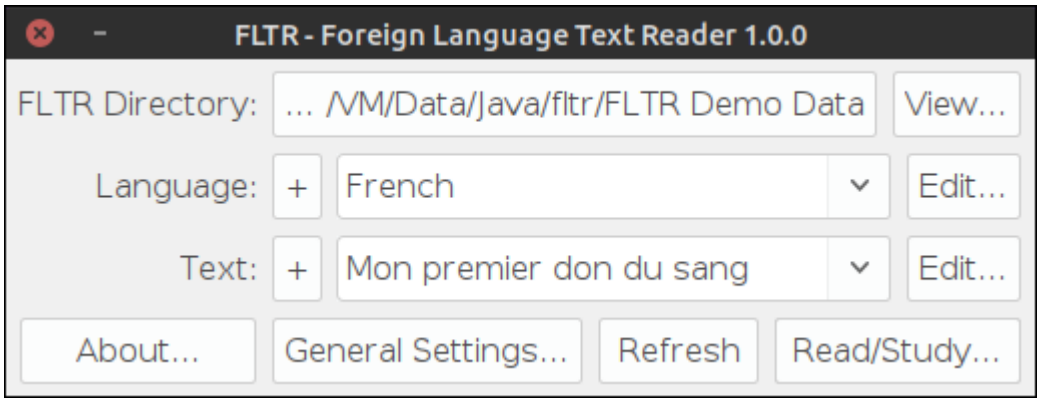
Screenshots

Please note that some screenshots were created with older versions of the software and have not been updated.

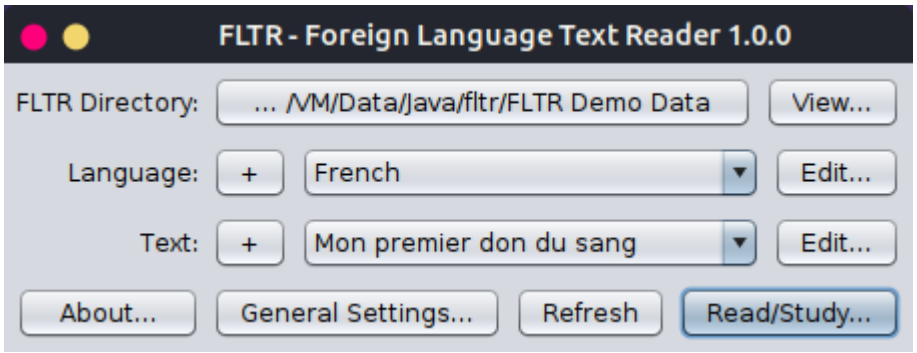
Main Window (Windows 10, System Look & Feel, 130 %)



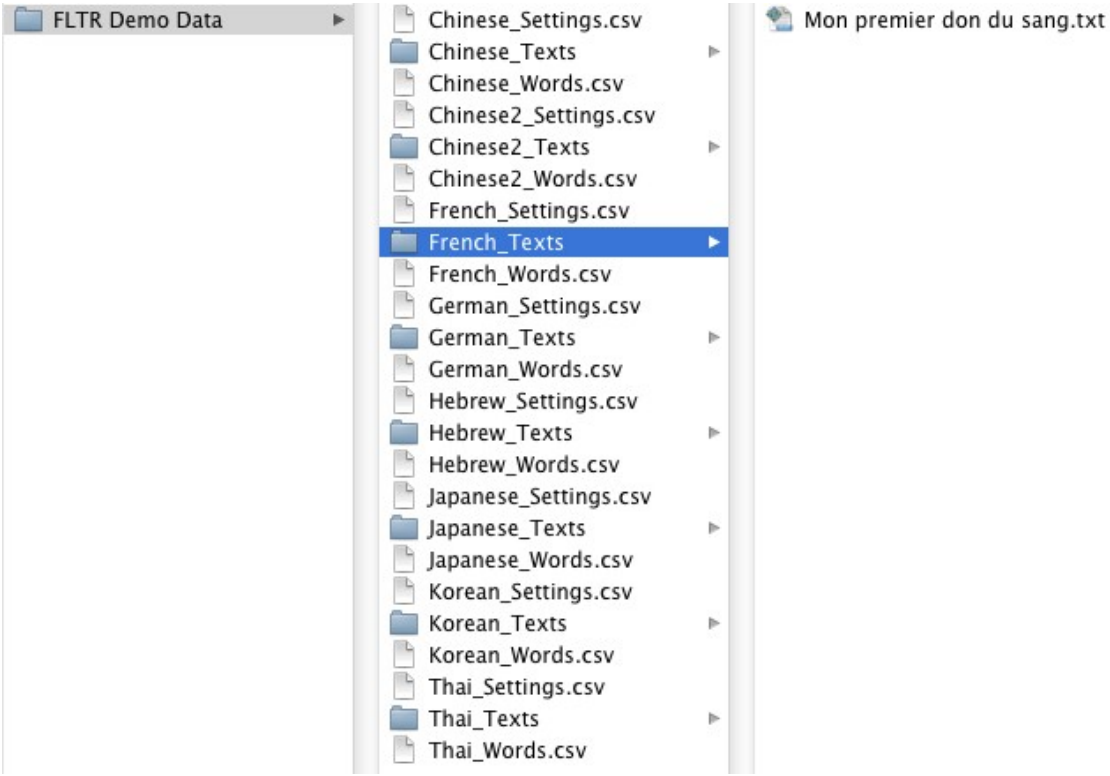
Main Window (Linux Mint, System Look & Feel, 120 %)



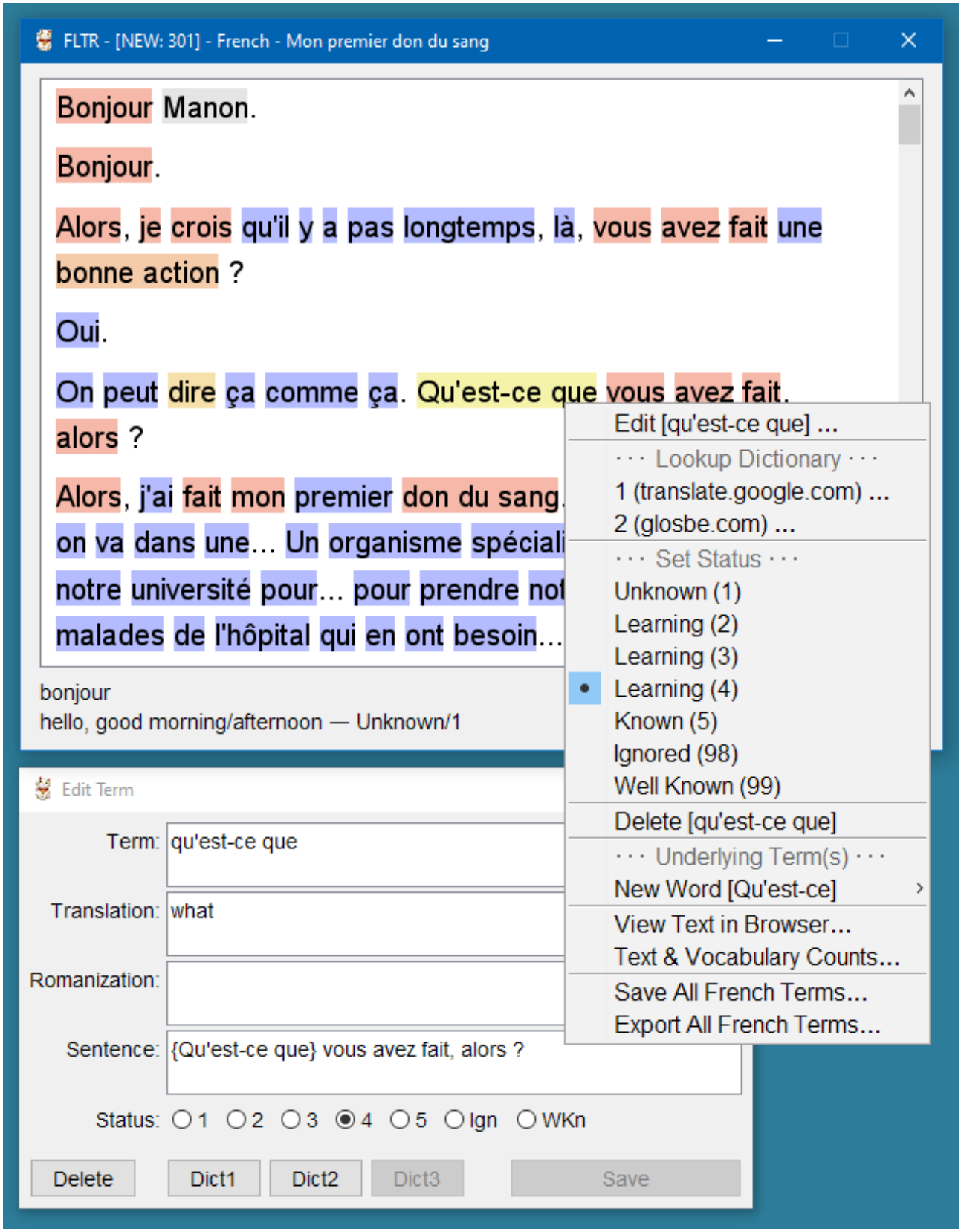
Main Window (Linux, Nimbus Look & Feel, 100 %)



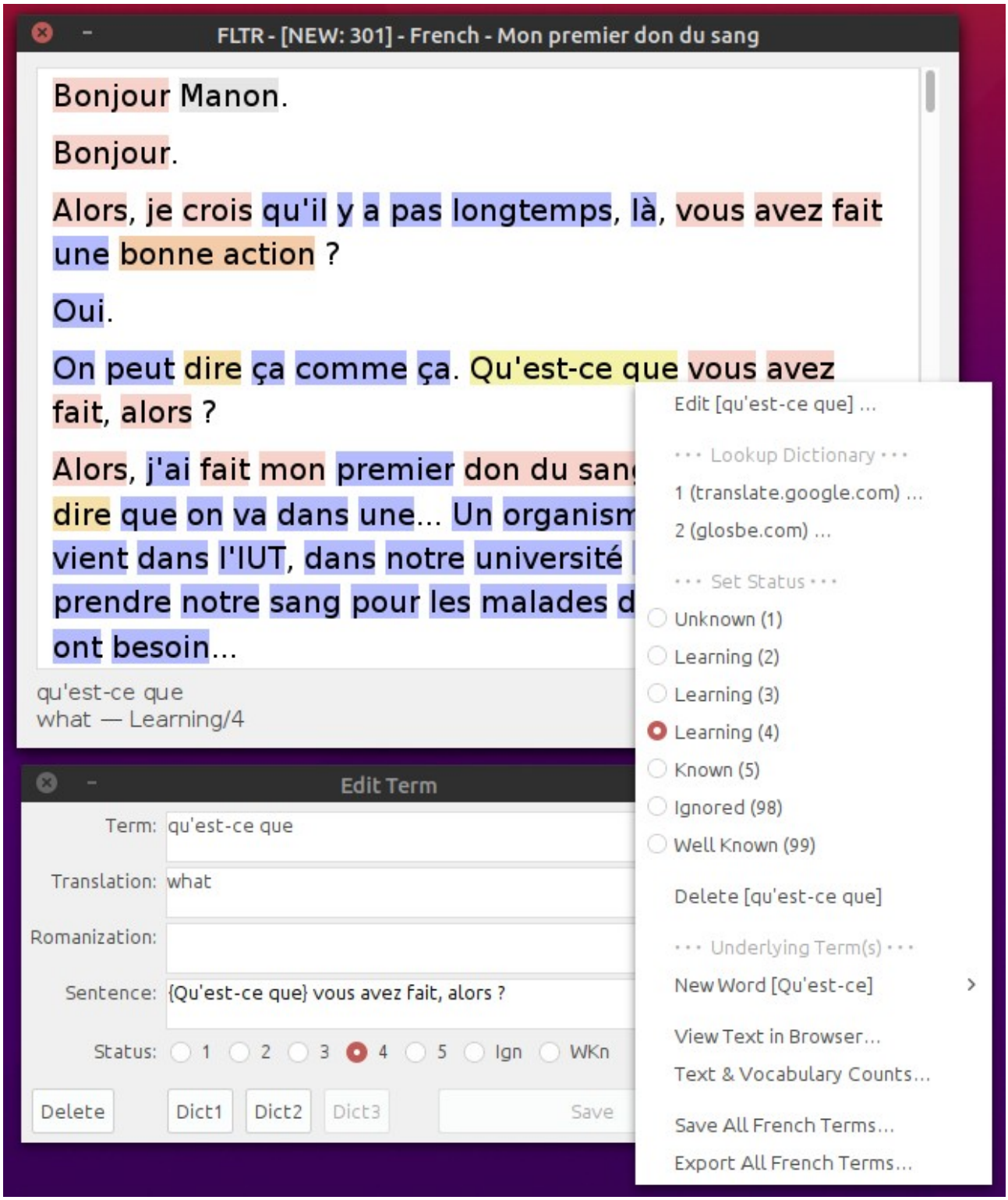
Directory & Files Structure



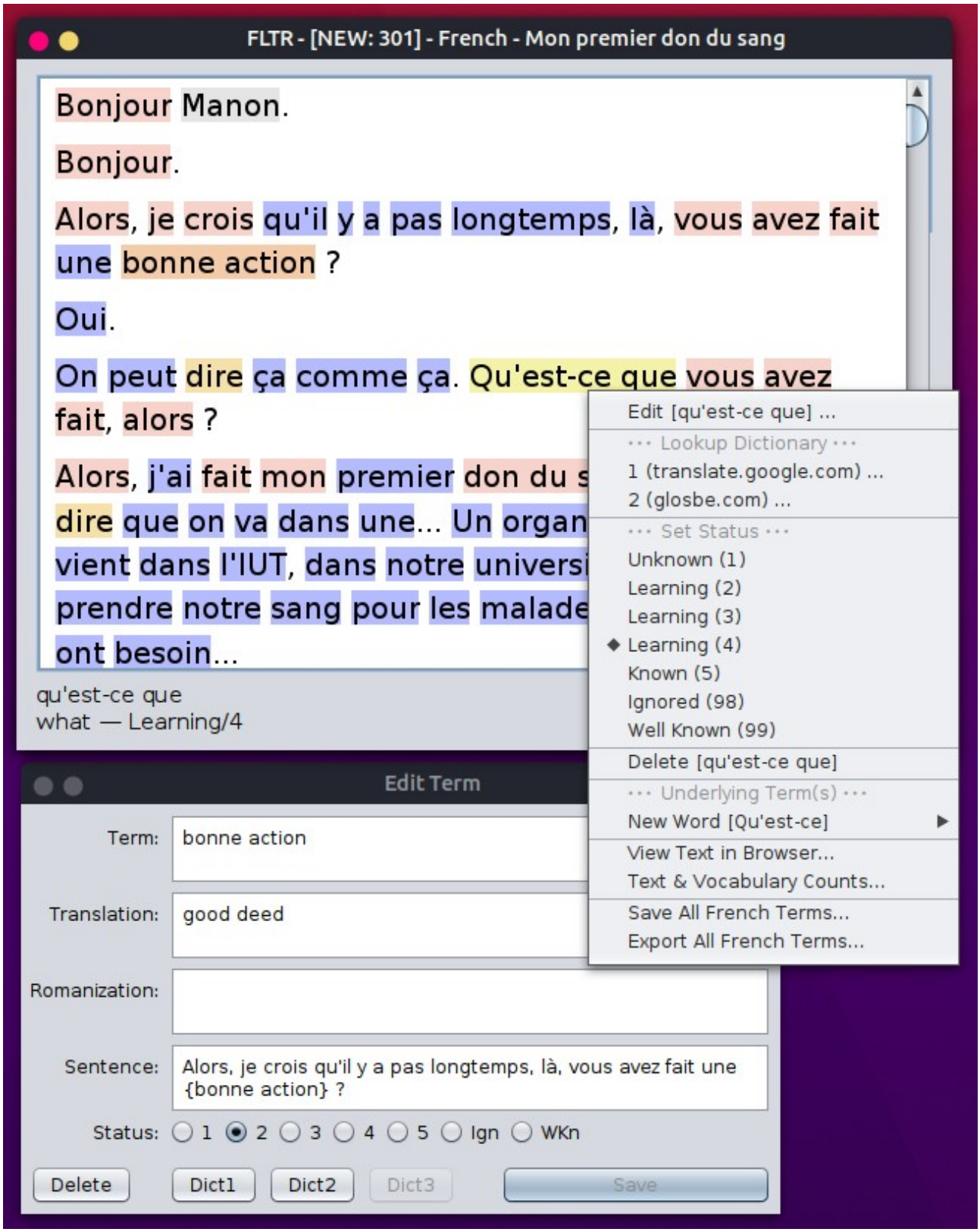
Text & Vocabulary Windows (Windows 10, System Look & Feel, 130 %)



Text & Vocabulary Windows (Linux Mint, System Look & Feel, 100 %)



Text & Vocabulary Windows (Linux, Nimbus Look & Feel, 100 %)



Overview

- First things first: **"Download and Installation"**.
- Now you must do the **"FLTR Initial Configuration"**.
- The language settings will be saved in "XXX_Settings.csv" (where XXX is the name of a language). These settings must be adjusted/changed for exotic languages and in order to specify dictionary URLs or program calls, font name, font size, etc. See: **"FLTR Language Settings"** and **"Integration with dictionary lookup software"**.
- General FLTR Settings like window sizes, etc. are saved in a file named ".prefs" in the user's home directory. See: **"General Settings"**.
- Now the fun begins: **"Reading Texts, Creating & Editing Words or Expressions"**.
- The vocabulary will be saved in "XXX_Words.csv". The optional Vocabulary Export will be saved in "XXX_Export.txt". Details see: **"Vocabulary File"**.
- You can work with your saved vocabulary: Just select <Vocabulary> in the text combo box, then filter, sort, and limit the terms, and decide whether to review as a text, or to review in your browser, or to export the data. See: **"Review And Edit Vocabulary"**.
- Not all questions answered? Read the **"Frequently Asked Questions (FAQ)"**.
- Finally: **"Compare Similar Software (FLTR, LWT, LingQ)"**
  and **"Version History"**.

Download and Installation

Step 1:

Check whether you have already Java installed or not:

- Enter **java -version** in a command or terminal window to see the installed Java version number, or an error message, if Java is not installed.
- FLTR needs **Java version 1.8 (aka Java 8)** or higher.

Step 2:

If you don't have Java on your computer - download and install Java:

- Go to **http://www.java.com/en/download/manual.jsp**
- Download the version for your operating system.
- Install it.
- Repeat step 1 to check whether Java is correctly installed.
- It is recommended to install the **newest** available version.

Step 3:

Download and install FLTR:

- Go to **https://github.com/hapepo23/foreign-language-text-reader**
- Click on " <> Code" and download the latest zip archive.
- Unzip it in a directory of your choice.

Step 4:

Start FLTR:

- Double-click on **FLTR.jar** (for all operating systems)
- Important: On Linux, UNIX, macOS, etc., **FLTR.jar** must be made "executable" to be run by the Java Runtime System. (Just run **chmod a+x FLTR.jar** in a Terminal.)
- You can also start FLTR via command line: **java -jar FLTR.jar**

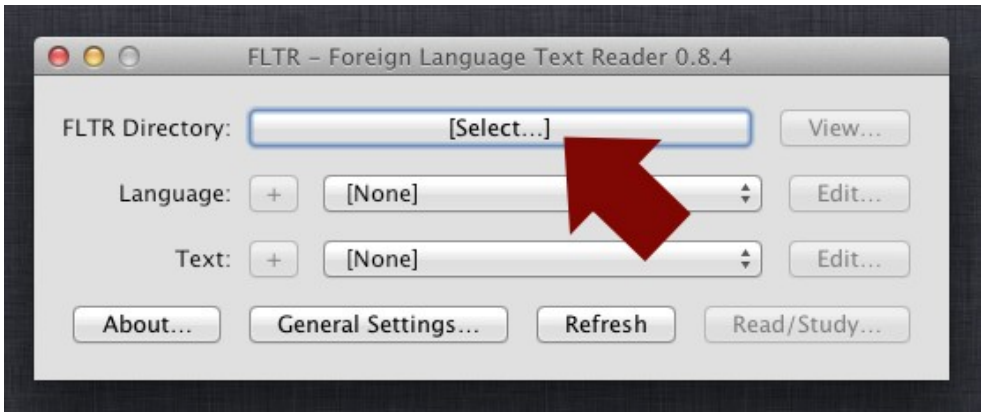
Read more about starting to use and to configure FLTR: **"FLTR Initial Configuration"**

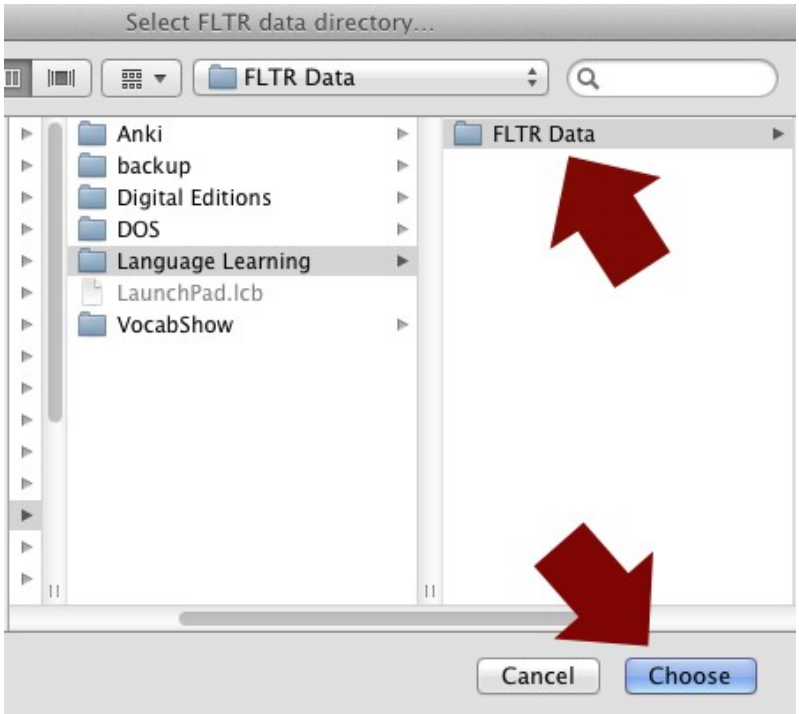
FLTR Initial Configuration

Create an FLTR data directory.

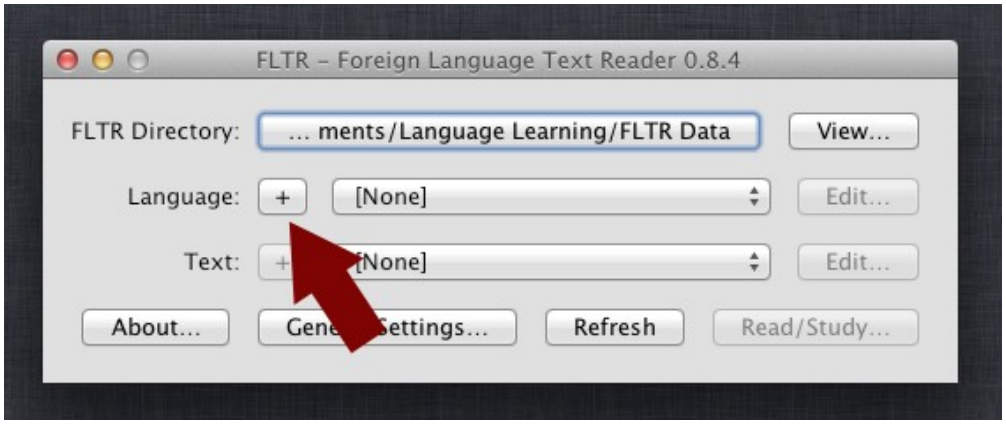
Start FLTR by double-clicking on FLTR.exe or FLTR.jar.

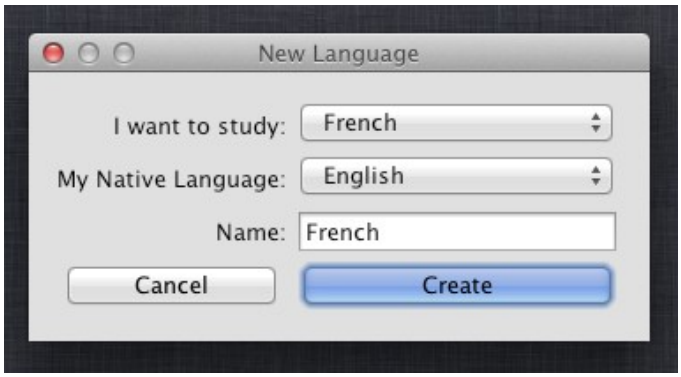
Choose the above created FLTR data directory.

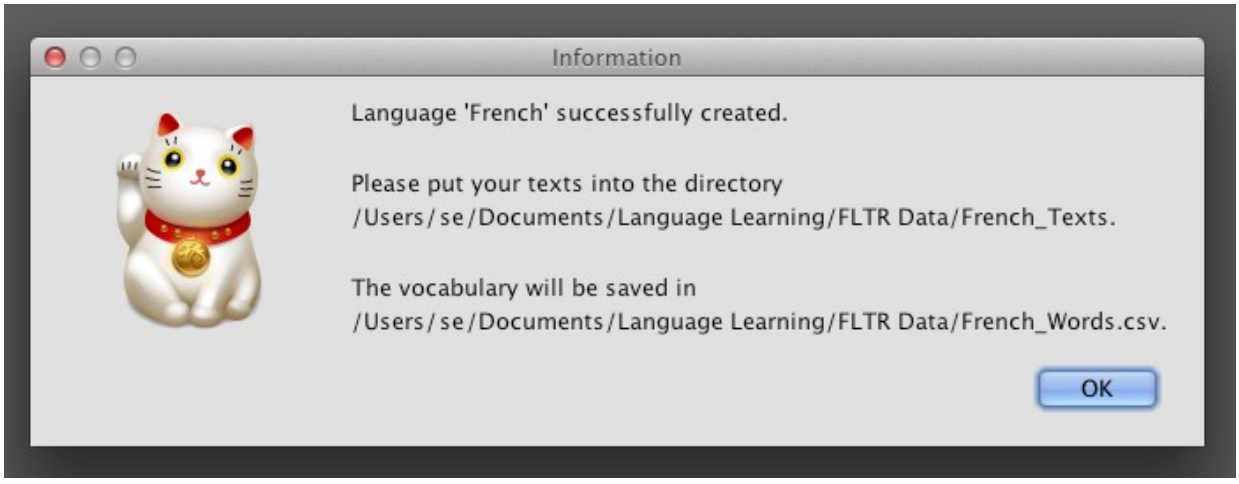


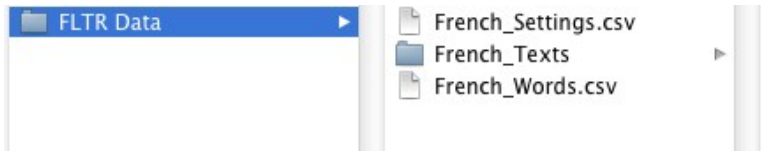


Now create a new language: (+). In the following text we call this language "XXX". Language Settings will be saved in "XXX_Settings.csv".









Now do the **FLTR Language Settings**.

FLTR Language Settings

Language Settings will be saved in "XXX_Settings.csv" where XXX is the language.

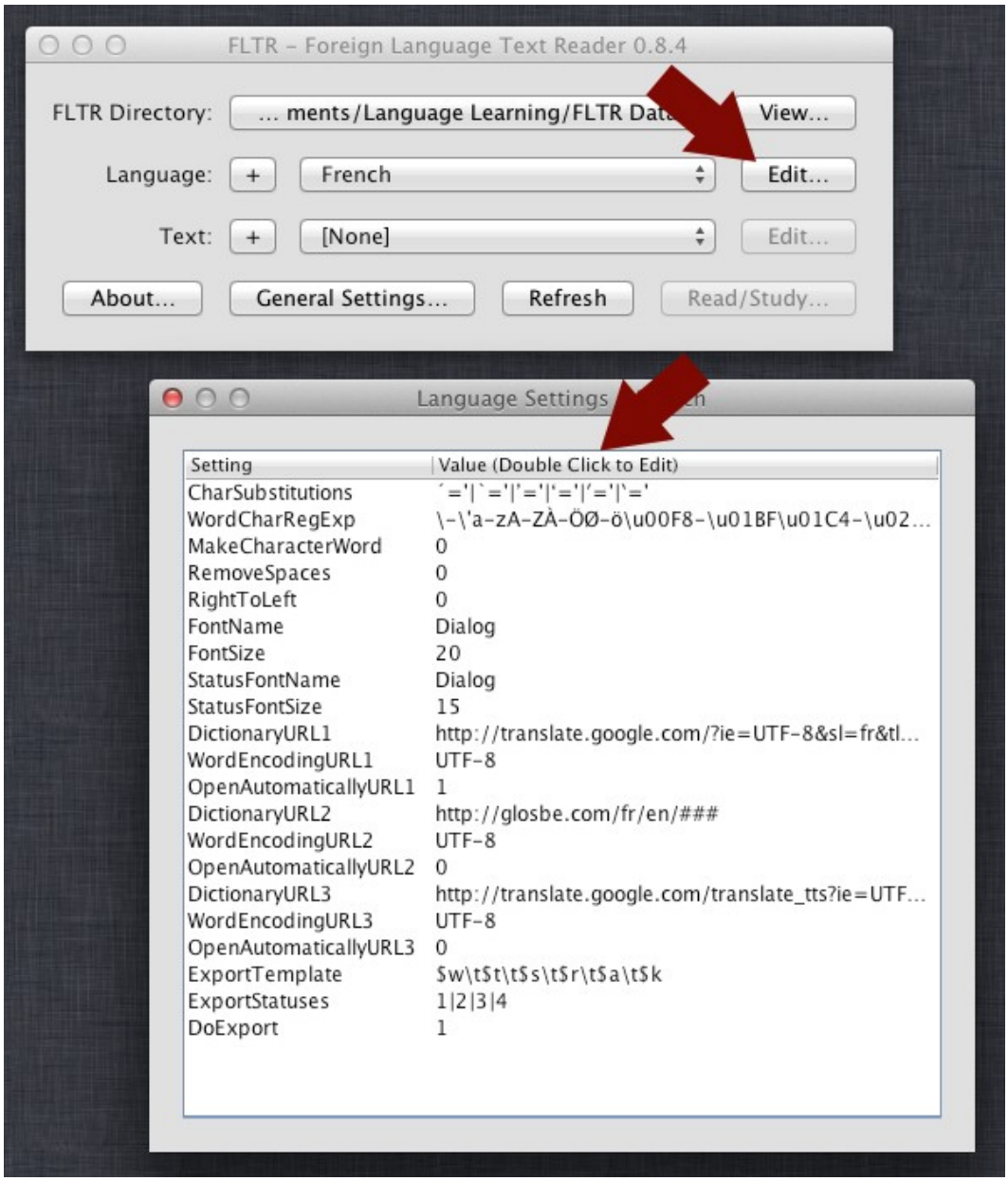
Format: keyword-value pairs, TAB delimited, without quotes, UTF-8 encoding.

If you create a new language, a "XXX_Settings.csv" file with default settings will be automatically created. This default is OK for most European languages including Greek and Cyrillic script. The DictionaryURL1 should be corrected or changed. A second and third dictionary can be added.

```
FLTRLANGPREFS
charSubstitutions ´='|`='|’='|‘='|'='|`='
wordCharRegExp    ⬋
         \-\'a-zA-ZÀ-ÖØ-ö\u00F8-\u01BF\u01C4-\u024F\u0370-\u052F\u1E00-\u1FFF
makeCharacterWord 0
removeSpaces      0
rightToLeft 0
fontName    Dialog
fontSize    20
statusFontName    Dialog
statusFontSize    15
dictionaryURL1    http://translate.google.com/?ie=UTF-8&sl=xx&tl=yy&text=###
wordEncodingURL1  UTF-8
openAutomaticallyURL1   1
dictionaryURL2    http://glosbe.com/xx/yy/###
wordEncodingURL2  UTF-8
openAutomaticallyURL2   0
dictionaryURL3
wordEncodingURL3  UTF-8
openAutomaticallyURL3   0
exportTemplate    $w\t$t\t$s\t$r\t$a\t$k
exportStatuses    1|2|3|4
doExport    1
```

The file must start with a line containing "FLTRLANGPREFS".

The settings can easily be modified via "Language Settings..."



Key	Default	Explanation
charSubstitutions	´='|`='|’='|‘='|'='|`='	List of character substitutions. These are mainly to ensure that similar characters like apostrophes cause problems in your vocabulary
wordCharRegExp	\-\' a-z A-Z À-Ö Ø-ö \u00F8-\u01BF \u01C4-\u024F \u0370-\u052F \u1E00-\u1FFF	A list of characters OR character ranges "x-y" that defines all characters in a word, e.g. English: "a-zA-Z", German: "a-zA-ZaöüÄÖÜß", Chinese: "一-龥" or "\u4E00-\u9FA5". If you don't want ' and - as part of words, remove \- and/or \'. See also: http://tinyurl.com/cbpndlt (Important: In LWT, Unicode ranges are defined: \x{....}-\x{....}, in FLTR you must specify: \u....-\u.... !)
makeCharacterWord	0	1=for Chinese, Japanese, otherwise=0
removeSpaces	0	1=for Chinese, Japanese, otherwise=0
rightToLeft	0	1=for Arabic, Hebrew, Urdu, etc., otherwise=0
fontName	Dialog	Font name for text display. You must choose a font that displays the language's characters properly.
fontSize	20	Font size for text display
statusFontName	Dialog	Font name for the text status line.
statusFontSize	15	Font size for the text status line
dictionaryURL1	http:// translate.google.com/? ie=UTF- 8&sl=xx&tl=yy&text=###	Dictionary URL No. 1 or command with parameter(s) to call an installed dictionary lookup program (see **next chapter**). Word placeholder must be

Key	Default	Explanation
		given as ###.
wordEncodingURL1	UTF-8	Encoding of word ### in URL 1
openAutomaticallyURL1	1	1=Link 1 or Cmd 1 opens automatically when clicking on word
dictionaryURL2	http://glosbe.com/xx/yy/###	Dictionary URL or command No. 2
wordEncodingURL2	UTF-8	Encoding of word ### in URL 2
openAutomaticallyURL2	0	1=Link 2 or Cmd 2 opens automatically when clicking on word
dictionaryURL3		Dictionary URL or command No. 3
wordEncodingURL3	UTF-8	Encoding of word ### in URL 3
openAutomaticallyURL3	0	1=Link 2 or Cmd 3 opens automatically when clicking on word
exportTemplate	$w\t$t\t$s\t$r\t$a\t$k	Template that defines how the vocab export XXX_Export.txt or individual vocab exports should be written, see VocabularyFile and/or ReviewEditVocabulary
exportStatuses	1|2|3|4	List of statuses, only the words with these statuses are exported into XXX_Export.txt
doExport	1	1=write always a vocab export file XXX_Export.txt, 0=no export

Integration with dictionary lookup software

Warning:
This chapter is for experienced users only.
You should know how to call a program in a terminal with path and arguments.
Backup your data before you start. Test everything thoroughly.

In the previous chapter you learned how to open dictionary entries from a website via a browser.

There are other sources besides calling a website to get the translation of foreign word or expression, e.g. dictionary lookup software like GoldenDict, translate-shell, etc. These applications must be downloaded and installed separately.

Well, how do you launch such a dictionary lookup software automatically and display the translation of a word you are looking for?

We must differentiate two types of software:

- GUI (Graphical User Interface) software that displays its results in a window, e.g. GoldenDict

- Software that writes its results as text in a terminal window, e.g. translate-shell

Both can be launched with the word(s) in question from FLTR.

Before you start defining a new dictionary lookup software in FLTR, you should **backup yor data**. So if something goes wrong or does not work, you can always go back.

You should also **not define more than one** dictionary lookup software for a language. The other two may be website URLs.

You must know **how to call your dictionary lookup software for a word** in a certain language.

Some examples (WORD is the word searched for) :

GoldenDict on Linux:
`goldendict "WORD"`

GoldenDict on Windows:
`"C:\Program Files (x86)\GoldenDict\GoldenDict.exe" "WORD"`

translate-shell on Linux (French → English, brief results):
`trans -b fr:en "WORD"`

translate-shell on Linux (French → English, detailed results):
`trans -no-ansi fr:en "WORD"`

You define the launch command in the language definitions in the Fields dictionaryURL1, dictionaryURL2, or dictionaryURL3.

If the contents of these fields starts with `http://` or `https://`, FLTR will launch your browser and opens the given website URL.
If not, it will interpret the contents as a command to launch a dictionary lookup software.

The definition must be in a special format:

`program path | parameter-1 | parameter-2 | … | … | parameter-n [&]`

The format may look a bit strange, but it has the advantage that you don't have to fiddle around with quotes and/or escape characters. Don't use single or double quotes or escape spaces, etc.!

The **word searched for** must be given as **###** without quotes.

The optional **& at the end** must be given if the program is a **GUI program** without text output (i.e. output will be displayed in a window of the GUI program). The result is also that FLTR does not wait until the program ends. It runs side-by-side with FLTR.

If you **don't give the & at the end**, the program should write the translation as text (e.g. in a terminal window) and must end afterwards. This program output is displayed in a special results window within FLTR, and **the translation can easily be marked (e.g. double click) and copied to the FLTR term window translation field via a "Mouse middle click".**

If the program does not end, it will block the following requests. Then you must click on the **KILL** button to solve this situation.

The above examples must specified as follows in the language settings "dictionaryURL1, -2, -3":

GoldenDict on Linux (GUI program):
```
goldendict | ### &
```

GoldenDict on Windows (GUI program):
```
C:\Program Files (x86)\GoldenDict\GoldenDict.exe | ### &
```

translate-shell on Linux (French → English, brief results, text output):
```
trans | -b | fr:en | ###
```

translate-shell on Linux (French → English, detailed results, text output):
```
trans | -no-ansi | fr:en | ###
```

Final notes:

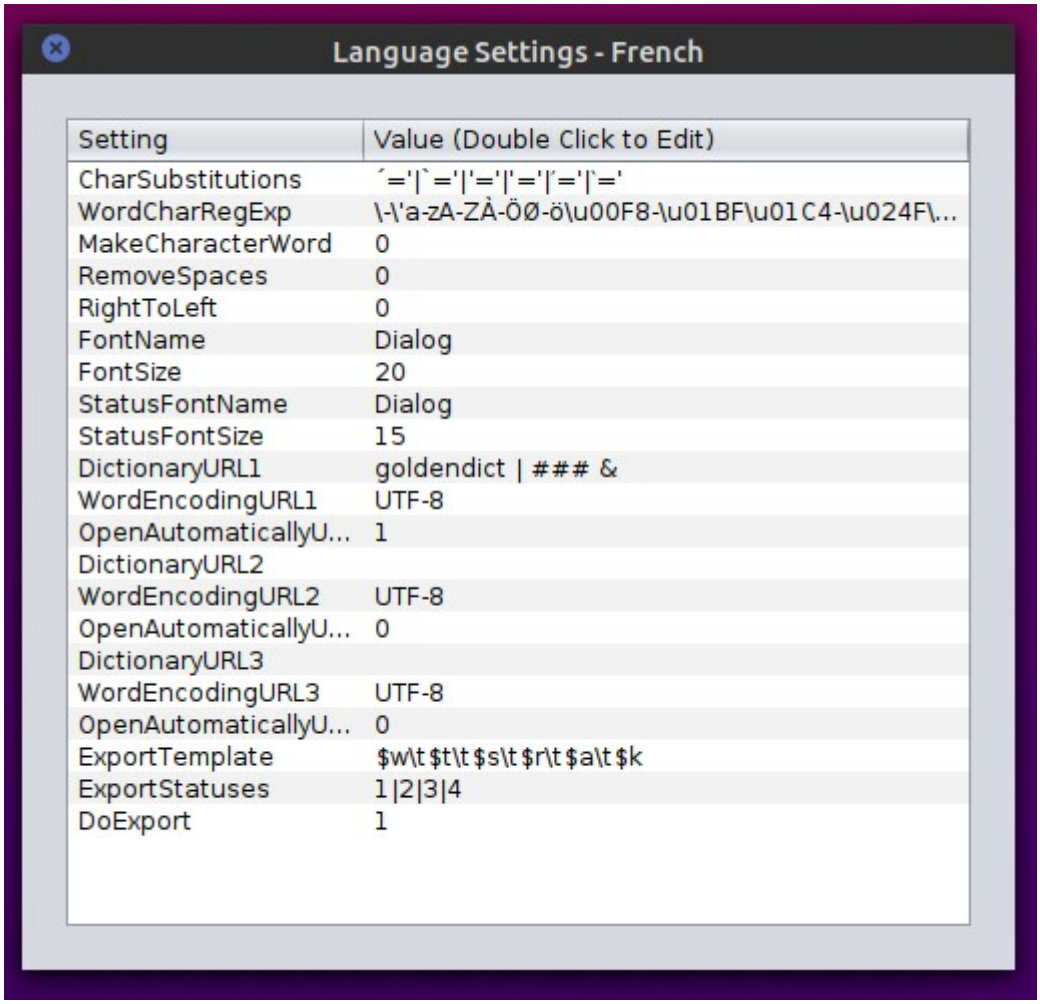
Due to unavailibility of Apple machines, this feature could not be tested under macOS.

translate-shell is not available under Windows but may be runnable in the Windows Subsystem for Linux (WSL). This has not been tested. In general, non-GUI programs under Windows that must run within the Windows Terminal, have many encoding and other problems.

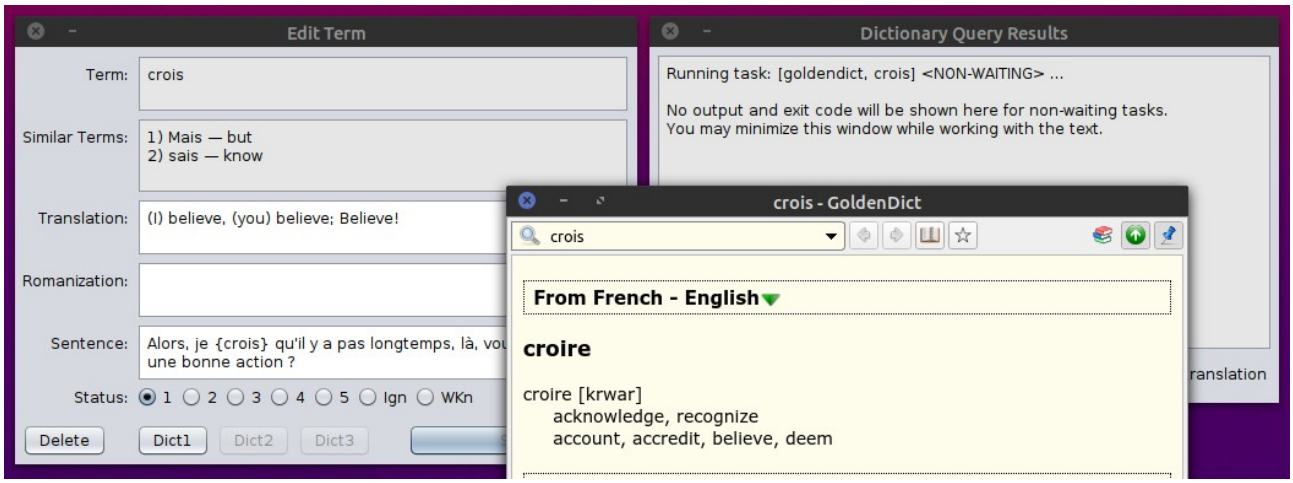
Again:
Make backups of your data before experimenting with the integration of dictionary lookup software into FLTR.

Example Screenshots (Linux):

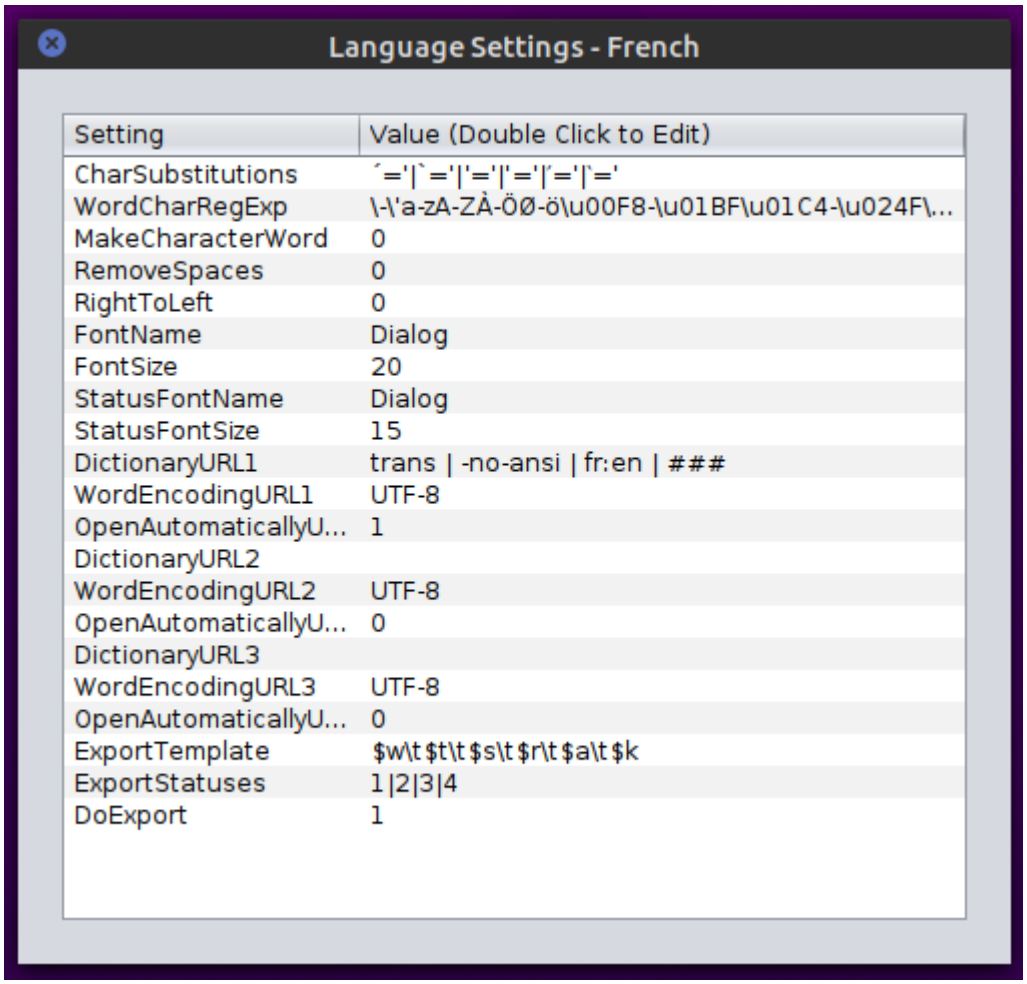
Screenshot of the definition – GoldenDict:



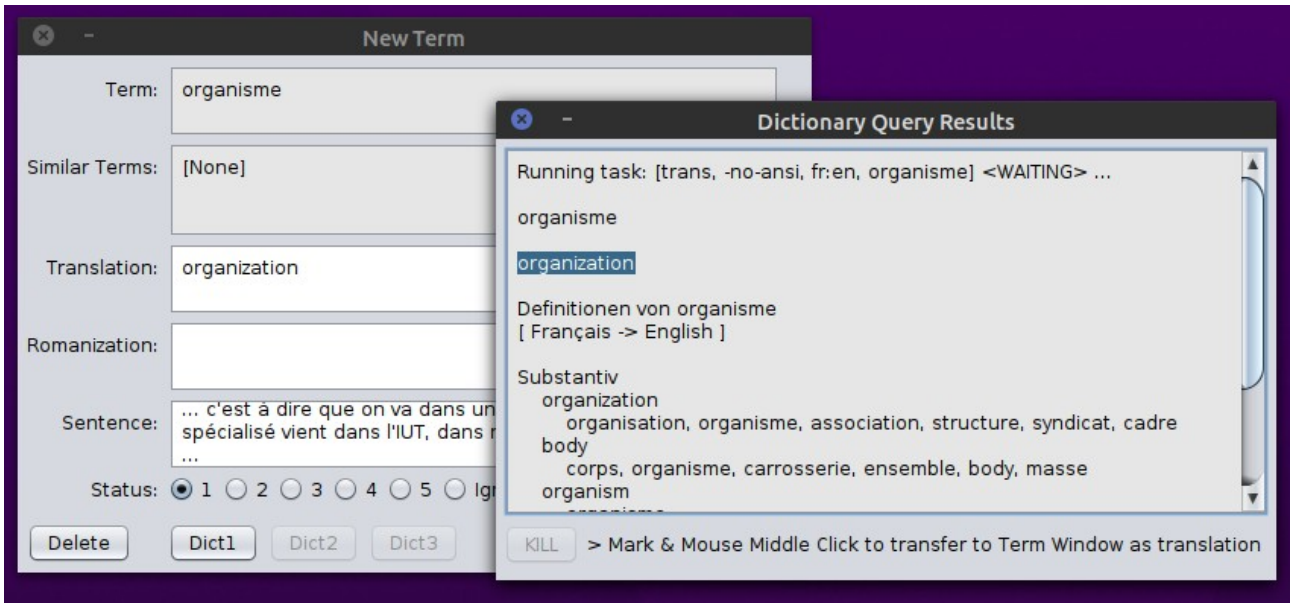
Screenshot of the usage – GoldenDict:



Screenshot of the definition - translate-shell:



Screenshot of the usage – translate-shell:

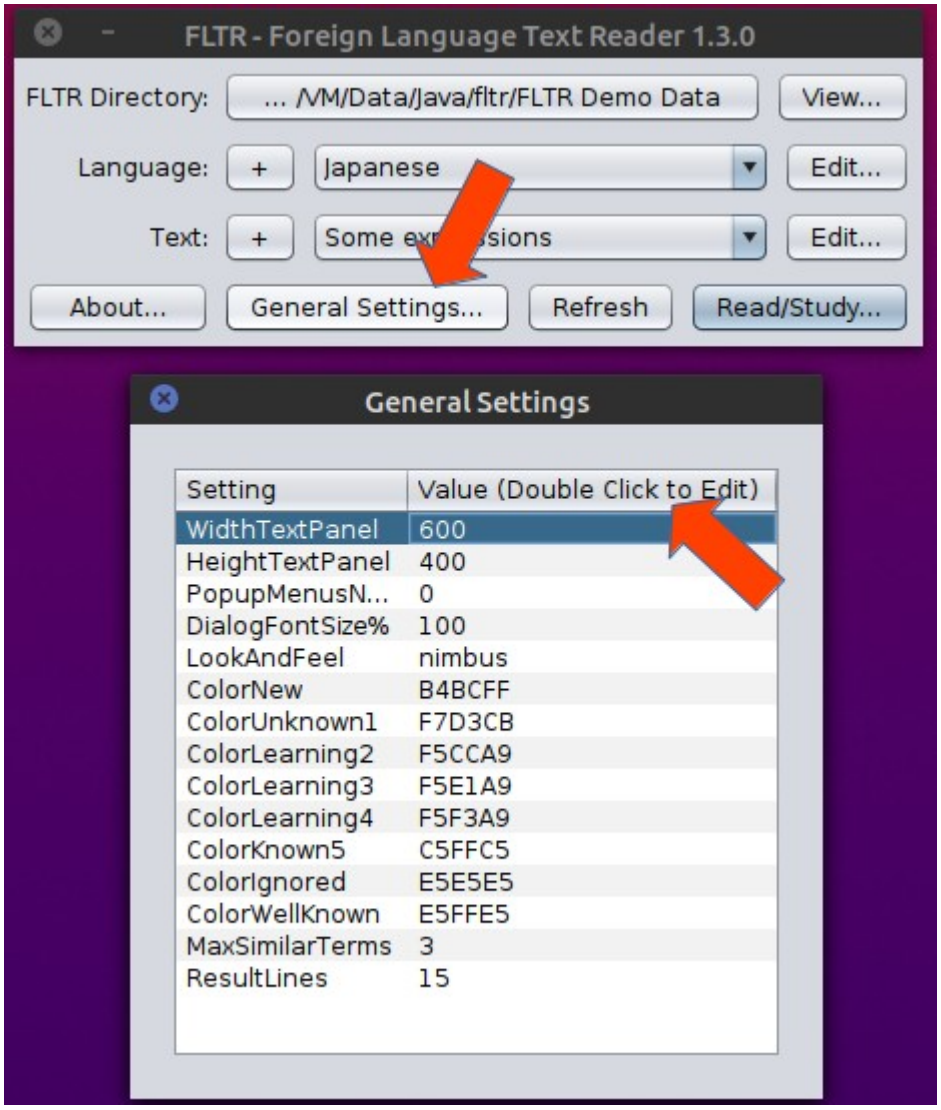


General Settings

General FLTR settings (text area size, window positions, look & feel, font size factor, popup menu type, colors) are saved in a file ".fltrprefs" in the current user's home directory (this file is hidden on Mac/Linux/Unix).

Format: keyword-value pairs, TAB delimited, without quotes, UTF-8 encoding. The file must start with a line "FLTRPREFS".

In the moment the only interesting data to be changed by a user (just click on "General Settings...") are, see screenshot:



- Width and Height of the text panel in the text window. Defaults are 600 x 400. It is set rather small because of the small screen sizes of netbooks. If you want a larger text area, just make these numbers larger, e. g. 1000 x 700. But keep in

mind that you need also screen space for the browser window, the data entry window, and a media player program window.

- Text area popup menus can be nested (old version = 1) or not (new version, =0, default).

- Dialog Font Size in percent, default = 100, allowed value range: 75 ... 150 %. On some platforms and certain look and feels, this value may be ignored. Find out yourself. Especially on Windows: 110 or 120 % are a good value with Look & Feel = system. Program must be restarted after changes.

- Look and Feel. Possible values: **system** (default), **nimbus** (font size must be 100 %), or **metal**. Use whatever you like. Program must be restarted after changes.

- HTML (hexadecimal or hex) color codes (without a "#" in front) of the background colors of the different term statuses. You may find out the code via https://htmlcolorcodes.com/. If you delete a color code, the default will be set automatically.
  Defaults:

New	B4BCFF
Unknown1	F7D3CB
Learning2	F5CCA9
Learning3	F5E1A9
Learning4	F5F3A9
Known5	C5FFC5
Ignored	E5E5E5
WellKnown	E5FFE5

- The number of similar terms displayed in the term window "MaxSimilarTerms" can be set from 0 to 10. This function can be switched off by setting the value to zero. The maximum number of similar terms is 10, the default value is 3.

- The number of lines in the Result Window (= results from calls to external dictionary lookup software) "ResultLines" can be set between 5 and 50, default: 15.

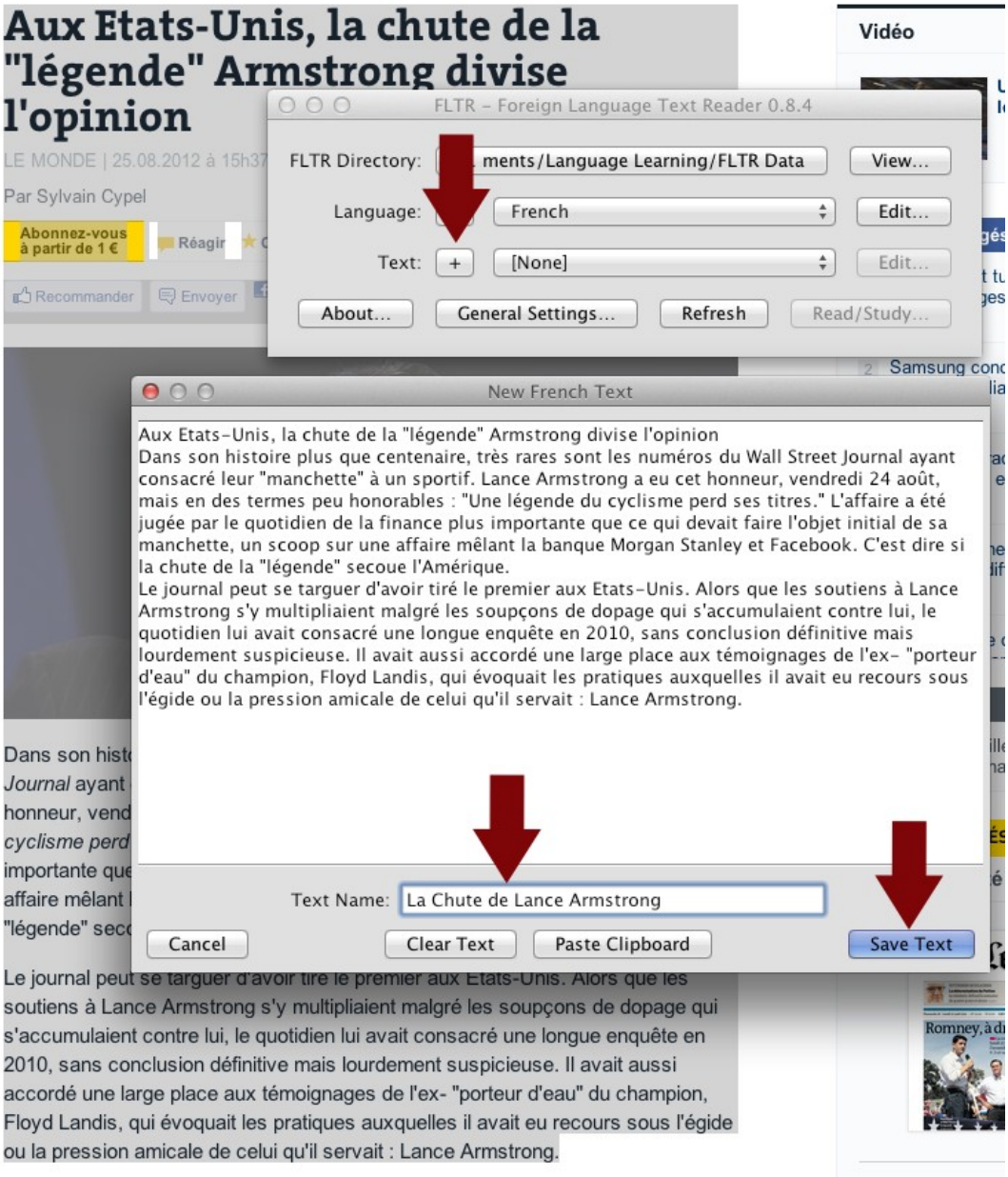
Reading Texts, Creating & Editing Words or Expressions

You have two possibilities to save a text in the language you are studying.

Method 1:

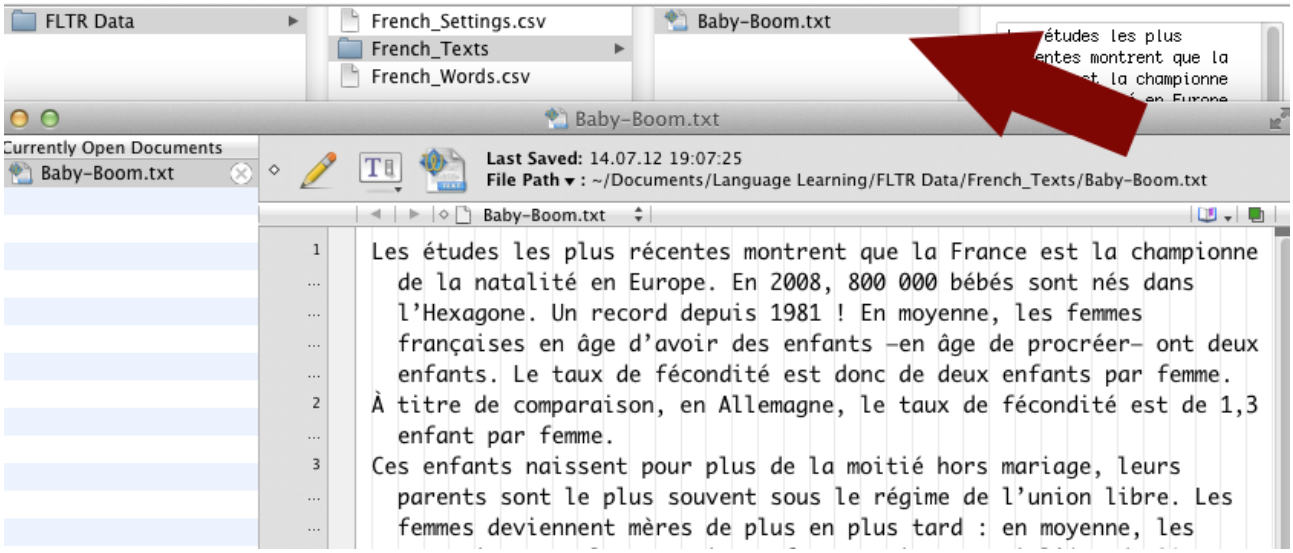
Copy some text (e.g. from the internet) into the clipboard (via Ctrl+C or Cmd+C).

Now click on (+) right to "Text:". The Text will be pasted from the clipboard automatically. You must supply a text name and click on (Save Text).

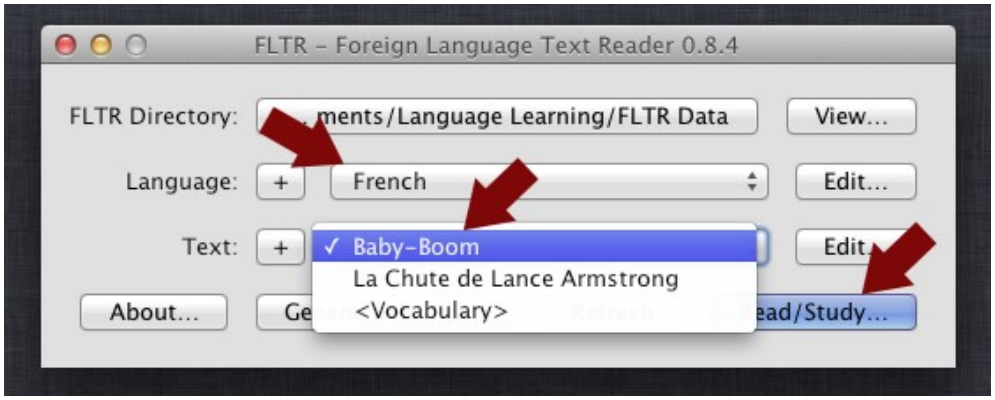


Method 2:

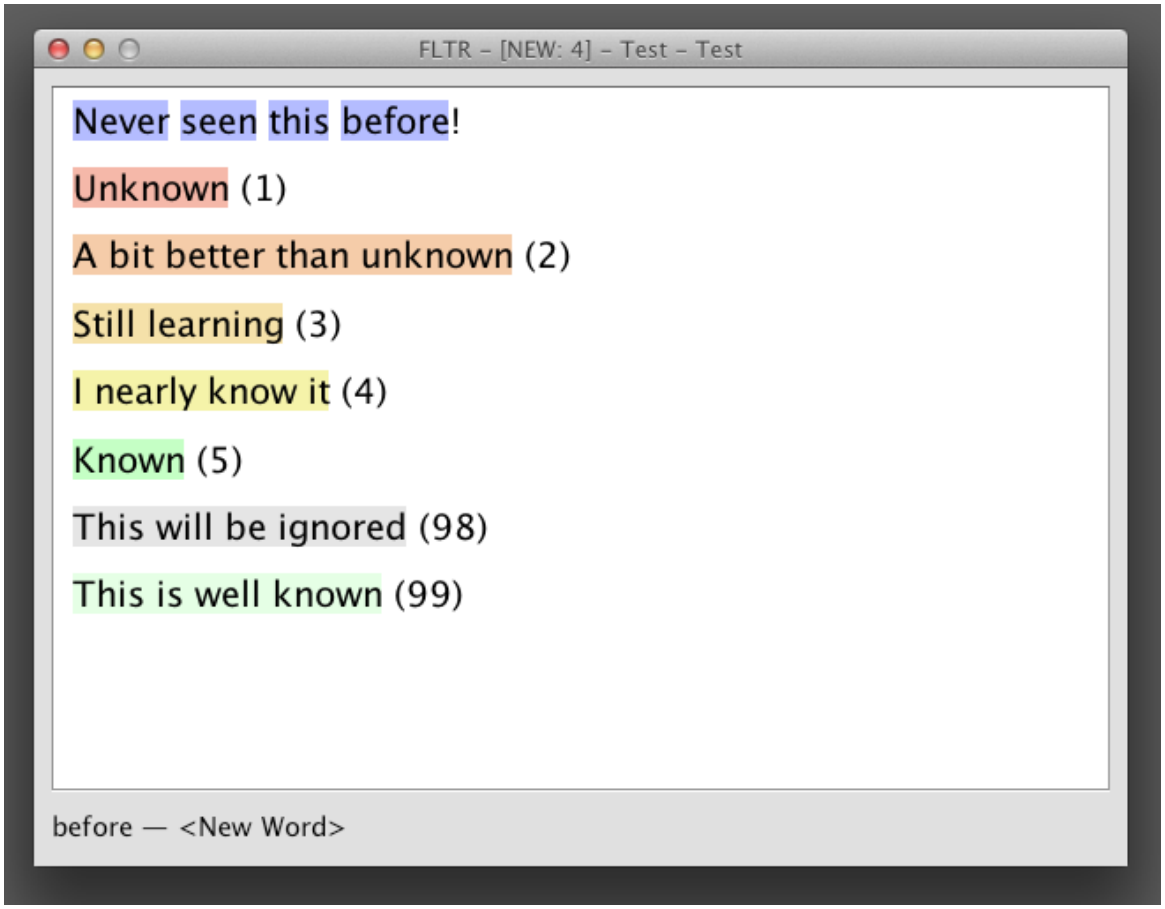
Save or copy some texts you want to study into the new directory "XXX_Texts" below the FLTR data directory. Please use a ".txt" file extension and save as UTF-8 encoding in a text editor like Notepad++ (Win) or TextWrangler (Mac). If you use a text processor like OpenOffice, LibreOffice, etc., save as "encoded text" and choose UTF-8 encoding and CRLF line breaks.



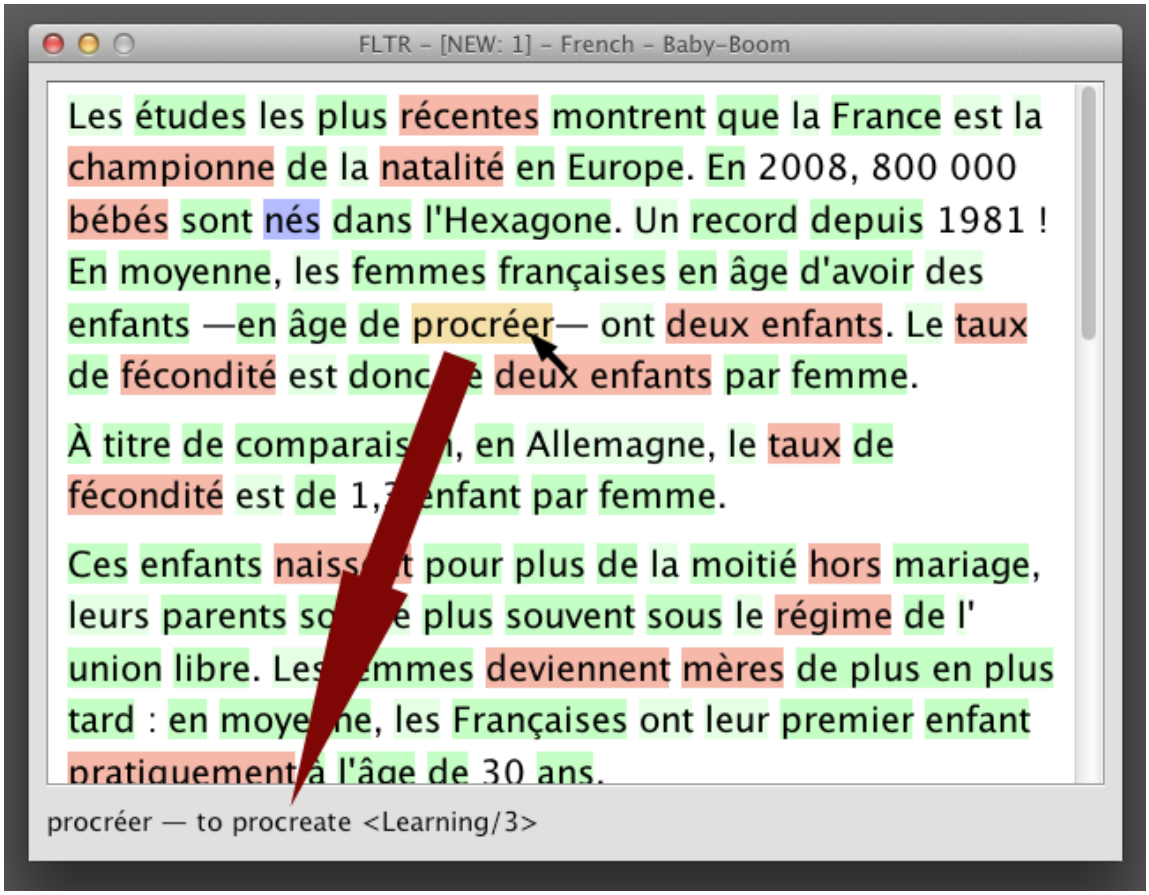
Click (Refresh) so that the program knows about the new saved text, then select a language, and select a text and click (Start Studying...).



Every word or multi-word expression (all called "terms") has a **Status**, designated by a color:



Hovering over a term: You see the translation, romanization (if any) and status of the term:



LEFT-click on a blue (new) term: The web dictionary (with openAutomaticallyURL.. = 1) will open, and you can create a new term or vocabulary item.

LEFT-click on a non-blue (existent) term: The web dictionary (with openAutomaticallyURL.. = 1) will open, and you can edit the term.

RIGHT-click on a word: A popup menu pops up, and you can create or edit the term, lookup dictionaries, change the term status, work with the underlying (hidden) terms, open the text in a browser, display vocab counts per text and per current language, save or export all terms of the current language.

LEFT-click and HOLD on first word, MOVE to last word and RELEASE: A new multi-word expression will be created, or if it already exists, it can be edited.
If the end of a paragraph is within such a multi-word expression, it can be created and saved, but will not show up in the text. But it will appear in a text with such a multi-word expression, but without paragraph ending(s) within.

Limitations: The length of a term or multi-word expression is limited: 200 characters max. The same applies to the translation and the romanization. The example sentence may be up to 400 characters long.

Vocabulary File

The Vocabulary will be saved in "XXX_Words.csv" where XXX is the language.

The previous version is available in "XXX_Words.bak".

Format: TAB delimited, without quotes, UTF-8 encoding. It's the same format like the LWT TSV export, so you can download your LWT terms (TSV export) and replace the file "XXX_Words.csv".

Only fields 1 to 5 will be read. Field 2 to 5 are optional, see defaults.

If the same word appears more than once while reading the vocab file, the last appearance wins.

Field	**Contents**	**Export File Code (as Text)**	**Export File Code (as HTML)**	**Default if missing**
1	Term (Word, Expression)	%w	$w	must not be empty
2	Translation	%t	$t	"?"
3	Sentence Fragment	%s	$s	""
4	Romanization	%r	$r	""
5	Status (1,2,3,4,5,98,99)	%a	$a	1
6	Term in Lowercase (=Unique Key)	%k	$k	Will always be the term (field 1) in lowercase

After you end working with your text, and you did some changes to your vocabulary, you will be asked whether you want to save your changes. If you click on (Yes), the complete vocabulary (and the export file, if "doExport" is set to 1) will be saved to disk (fields 1 to 6).

Export File

The above codes (see table) are placeholders for the "Export Vocabulary" template "exportTemplate".

The Export file will be created if "doExport" is set to 1 in the Language Settings. It contains all vocab items that have a current status of the list of statuses in the language setting "exportStatuses".

The export file XXX_Export.txt can be easily (re)imported into Anki (use the $.. placeholders!).

The %w, %t, %s, %r, %a, %k are placeholders for the raw data, see table.

The $w, $t, $s, $r, $a, $k are placeholders for the data with escaped HTML special characters (< = <, > = >, & = & " = ", see table.

To enter a TAB character, use \t.

To enter a NEWLINE character, use \n.

To enter a dollar, percent character or backslash: use $$, %% or \\.

There are special placeholders for cloze tests:

%c = The sentence, with the {...} are replaced by "{***}".
$c = like %c, but escaped HTML special characters.

%d = The sentence, with the {...} are replaced by "{***Translation***}".
$d = like %d, but escaped HTML special characters.

Export File Example 1

exportTemplate = $w\t$t\t$s\t$r\t$a\t$k

EXPORT (escaped HTML special characters):

Word[TAB]Translation[TAB]Sentence[TAB]Romanization[TAB]StatusCode[TAB]Key

Export File Example 2

exportTemplate = $w\n$t\n$s\n$d\n$a\n$k\n---

EXPORT (escaped HTML special characters):

Word
Translation
Sentence With {Word}
Sentence With {***Translation***}
StatusCode
Key

Review And Edit Vocabulary

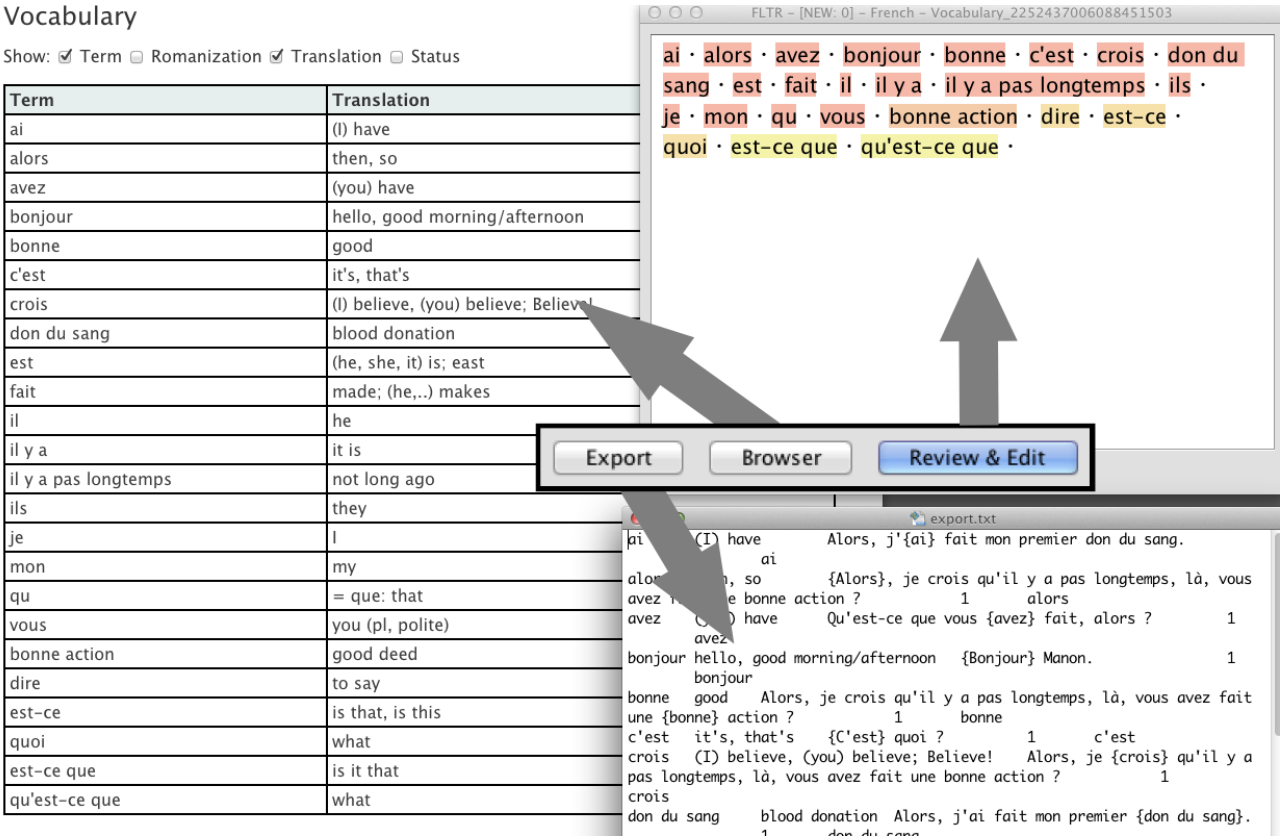
When selecting a text, you will see an entry <Vocabulary>.

If you select this special entry, and click (Start Studying...), you will see the "Vocabulary Filter and Sort Options".

You can now filter, sort, and limit your Vocabulary, and this selection of terms can now be ...

- reviewed/edited like a text,
- reviewed/tested in a browser,
- or saved as a text file according to the export template, see **"Vocabulary File"**.

Actions:


Vocabulary
Show: Term Romanization Translation Status
Term
Translation
ai
(I) have
alors
then, so
avez
(you) have
bonjour
hello, good morning/afternoon
bonne
good
c'est
it's, that's
crois
(I) believe, (you) believe; Believe!
don du sang
blood donation
est
(he, she, it) is; east
fait
made; (he,..) makes
il
he
il y a
it is
il y a pas longtemps
not long ago
ils
they
je
I
mon
my
qu
= que: that
vous
you (pl, polite)
bonne action
good deed
dire
to say
est-ce
is that, is this
quoi
what
est-ce que
is it that
qu'est-ce que
what
FLTR - [NEW: 0] - French - Vocabulary_2252437006088451503
ai · alors · avez · bonjour · bonne · c'est · crois · don du sang · est · fait · il · il y a · il y a pas longtemps · ils · je · mon · qu · vous · bonne action · dire · est-ce · quoi · est-ce que · qu'est-ce que ·
Export
Browser
Review & Edit
export.txt
ai (I) have Alors, j'{ai} fait mon premier don du sang. 1 ai
alors then, so {Alors}, je crois qu'il y a pas longtemps, là, vous avez fait une bonne action ? 1 alors
avez (you) have Qu'est-ce que vous {avez} fait, alors ? 1 avez
bonjour hello, good morning/afternoon {Bonjour} Manon. 1 bonjour
bonne good Alors, je crois qu'il y a pas longtemps, là, vous avez fait une {bonne} action ? 1 bonne
c'est it's, that's {C'est} quoi ? 1 c'est
crois (I) believe, (you) believe; Believe! Alors, je {crois} qu'il y a pas longtemps, là, vous avez fait une bonne action ? 1 crois
don du sang blood donation Alors, j'ai fait mon premier {don du sang}. 1 don du sang

Frequently Asked Questions (FAQ)

Q:
Some characters don't show up, or are displayed wrongly.

A:
Change the font in the language settings. Try "Dialog", or set it to a font suitable for the language you are studying, or install a Unicode font and set it in the language settings.
See also **http://en.wikipedia.org/wiki/Unicode_font**

Q:
Where is the audio player?

A:
There is none. Just use your favorite audio player: iTunes, Winamp, VLC, etc. If you are an intermediate to advanced language learner, you will often use FLTR without listening to the audio, just because of the fact that an associated audio file does not exist.

Compare Similar Software (FLTR, LWT, LingQ)

Feature	FLTR	LWT	LingQ
Programmed in	Java	PHP, JavaScript	Unknown, possibly PHP, JavaScript
Application Type	Desktop App	Web App	Web & Mobile Apps
Data Storage	Text & TSV Files	MySQL DB	Unknown, possibly MySQL DB
Open Source	Yes	Yes	No
License	MIT	Public Domain	Proprietary
Prerequisites	Java Runtime	Local or Remote Web Server, Web Browser	Web Browser
Installation	Un-ZIP and Run	Webserver Install., Un-ZIP, Run	None
Speed	Very fast	Locally quite fast	Medium to slow, depending on Internet Connection and Server Load
Languages	All Unicode (also Right-To-Left)	All Unicode (also Right-To-Left)	Currently (2021) 24 languages
Audio Support	No (Use your PC's player)	Yes, HTML5 player	Yes, HTML5 player
Text Tag Support	No	Yes	Yes
Text Archive	No, but just move text(s) in/out of the FLTR text directory	Yes	Yes
Term Tag Support	No	Yes	Yes
Text Management	Text Files in UTF-8	Upload into DB	Upload, must be paying member
Term Import	LWT: easy (via "Export all Terms (TSV)"). Other: e.g. via Spreadsheet	Flexible Import Function	Simple Import Function, must be paying member
Term Export	Flexible, via Template	Anki & TSV	Simple CSV, must be paying member

Feature	FLTR	LWT	LingQ
Max. No. of Terms	Unlimited	Unlimited	Free member: 20, Paying member; unlimited
Max. No. of Words in Multi-Word-Expressions	Unlimited, but 200 characters max.	9	Unknown
Display of Similar Terms	Yes, 10 max.	Yes	No
View/Edit of underlying Terms	Yes, via right click	Yes, after display mode switching	Yes, after clicking
Testing	No, use Anki or other flashcard programs	Yes	Yes
Statistics	Only Word counts	Yes	Yes
Cost	Free	Free	≈10 US$/month

Version History

0.8.9 (2019-03-20)
Converted to Java 1.8.
External dictionaries with secure (https://...) and unsecure (http://...) web protocol are now supported.
"FLTR for macOS + Apple's Java 1.6" is no longer available, as development is now done in Java 1.8. See installation.
Some minor changes.

0.8.10 (2020-07-17)
Logic to open a URL in the default browser has been modified.

1.0.0 (2020-10-03)
Moved to a different Sorceforge project.
Some minor corrections (web links, logic to open external programs, About info).

1.1.0 (2021-05-25)
Editing of the background colors to display the different levels is now possible.
Paste, Copy, Cut operations in the term window are now possible using the mouse.
Displaying similar terms in the term window has been removed (had no value and function in previous releases).
Saving and exporting of all terms while working with a text via right-click popup menu is now possible.
Some improvements in the term window.

1.2.0 (2021-05-30)
The display of similar terms in the term window is possible again and now configurable. The display is now in a non-editable text field with a maximum of 3 lines. This function can be switched off by setting the "MaxSimilarTerms" preference to zero. The maximum number of similar terms is 10, the default value is 3. You can copy and paste text from this text box.
The term in the term window is not editable now.

1.3.0 (2021-06-09)
Integration of FLTR with dictionary lookup software like GoldenDict, translate-shell, etc. is now possible. See Chapter **"Integration with dictionary lookup software"**.
Confirmation dialog box implemented for the "I know all" actions.

1.4.0 (2021-06-14)
The last reading position in a text is now automatically saved and restored when you

open it again. But, after changing the text font, text font size or text panel width/height, the reading position may be wrong after such changes.

1.4.1 (2025-10-17)
Moved to GitHub. Minor corrections.