

Translate Files for Xojo with AI

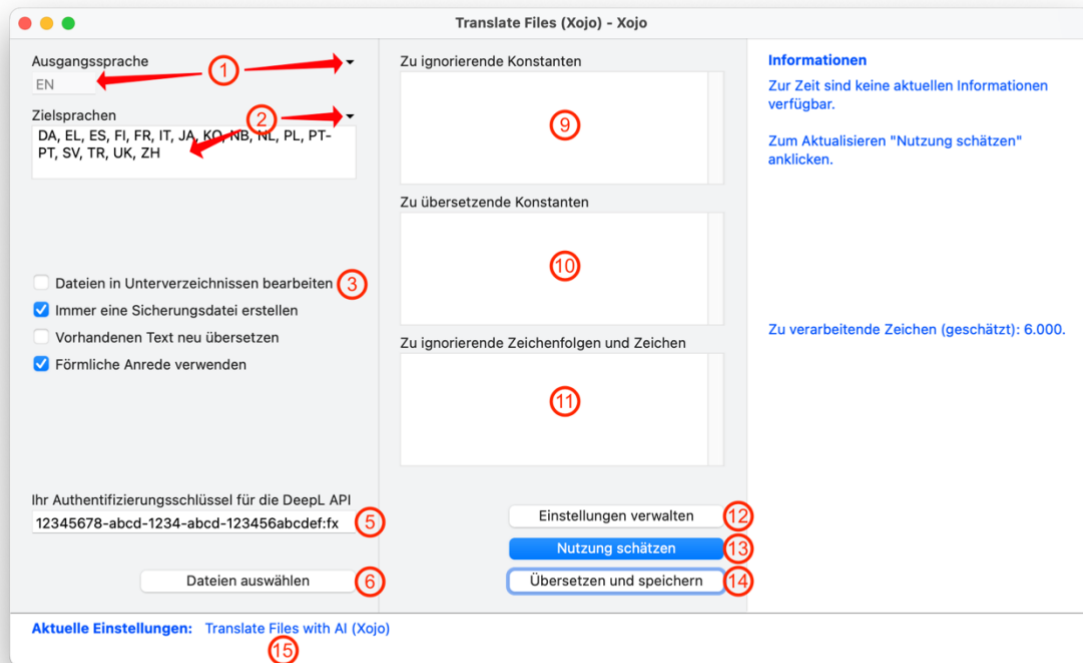
Table of Content

Important notes	1
The program window	2
General and preparations	2
Quick start	3
Languages	3
Selecting the project folder	4
Note	4
Constants to be ignored and translated ⁽⁹⁾ , ⁽¹⁰⁾	4
Special characters ⁽¹¹⁾	4
Note	4
Formal salutation	5
Notes	5
Settings	5
Sets of settings ⁽¹²⁾ , ⁽¹⁵⁾	5
Regular expressions	6
Log file	6
Contact	6

Important notes

- The latest version of this manual can be found at this link in the download section: translate.tomsidede.de.
- Some screenshots are not translated yet. This will be made up for. However, the app itself is already fully translated into your language.
- Make sure you always have a backup copy of your data.
- **Interesting use cases and examples can be found on my blog at translate.tomsidede.de.**

The program window



The markings refer to the annotations in the text.

General and preparations

Translate Files with AI is an app for the automated translation of localizable constants in Xojo source files of type `.xojo_code` and `.xojo_window` into one or more languages. This software uses DeepL to translate the texts. DeepL is currently the world's best translation service and provides meaningful, understandable translations.

It is important to know that the software (the app) itself does not perform translation. The software serves as an interface between the texts in your text files and DeepL. The translation itself is done by DeepL on their servers.

Therefore, you need an account at DeepL. When you register in the developer area of [deepl.com](https://www.deepl.com), you need to make sure that you get a new API key. This is the key that allows this software to use the services of DeepL for translation. Without this key, no text can be translated.

When you first open an account on DeepL, you can get a free API key that allows you to translate up to 500,000 characters per month. If you want to translate more characters within a month, you need to purchase another key and a certain number of characters to translate. This key must be entered into this app instead of the free key.

A free key is easy to recognize because it ends with ":fx". For more information and to get a free and/or paid API key, click the following link:

www.deepl.com/en/pro#developer

Enter the key in the corresponding field of the app ⁽⁵⁾.

Quick start

This section is intended for those who want to use the app quickly without having to read long explanations.

1. Select the code of the language to be translated ⁽¹⁾ and the code of the target language(s) ⁽²⁾ using the two drop-down menus. You can reach these menus by clicking on the small triangle or by right-clicking on the corresponding field ⁽¹⁾, ⁽²⁾.
2. In the target language selection field ⁽²⁾, you can select multiple target languages by holding down the ⌘ key on MacOS or the control key on Windows if you want to translate your texts into multiple languages. You can also select a single language by double-clicking on it.
You exit the selection field by clicking on an area outside the input field or by pressing the Enter key or the Tab key.
3. In the **Open** menu or with ⌘+O (MacOS) or ^+O (Windows), select the folder that contains the file(s) with the constants to be translated, i.e. normally the folder in which your Xojo project is located. Alternatively, you can drag a source folder onto the app.
4. 4. if there are also files with constants to be translated in subfolders of the selected project folder, check the **Edit files in subdirectories** checkbox ⁽³⁾.
5. The number of characters to be translated is estimated with the button ⁽¹³⁾.
6. Start the translation itself with the button ⁽¹⁴⁾.



It is best to try out a translation with a short text first in order to minimize the load on your quota at DeepL.

Languages

The following languages are supported:

BG, CS, DA, DE, EL, EN, EN-GB, EN-US, ES, ET, FI, FR, HU, ID, IT, JA, KO, LT, LV, NB, NL, PL, PT, PT-BR, PT-PT, RO, RU, SK, SL, SV, TR, UK, ZH

Selecting the project folder

Files of type `.xojo_code` and `.xojo_window` must be located in and below the project folder. These are the files that contain the constants to be translated.

Note

If there are no such files in your project folder, the project was probably not saved as Xojo Project but as Xojo Binary Project or Xojo XML Project. In this case, start Xojo and save the project as Xojo Project. Only in this case will the project be saved in the format in which it can be edited by Translate Files with AI. Only correctly licensed versions of Xojo can save a project in this format. Free versions cannot do this.

Constants to be ignored and translated ⁽⁹⁾, ⁽¹⁰⁾

These fields can be used to describe the names of constants that are edited according to the field name. It may be useful to use the placeholders for regular expressions for the beginning and end of lines. The simple specification of the constant `kEdit` would not only describe the constant itself but also all other constants that contain this name. For example, the constant with the name `kEditHelp` would also be affected. The specification `^kEdit$` would avoid this conflict.

Special characters ⁽¹¹⁾

It may happen that special characters or strings embedded in the text are corrupted during translation. Such values can be specified in the **Characters and strings to ignore** field to exclude them from translation.

It is also possible that certain words or terms should not be translated. These can also be specified here.

The translated text should definitely be checked for translation errors in the source program.

Note

Some special characters are translated and/or interpreted differently in different languages. This can lead to unexpected results in the translated constants. This applies in particular to languages that use completely different characters, e.g. Chinese, Japanese, Korean.

The following list contains some of these characters, but is not necessarily complete.

! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { | }
 ~ ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾
 ¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý
 Þ ß à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü
 ý þ ÿ Œ œ Š š Ÿ ƒ – — ` ’ , “ ” „ † ‡ • … ‰ € ™

Formal salutation

In some languages, it may be important to distinguish between formal and informal salutations. This applies to DE (German), FR (French), IT (Italian), ES (Spanish), NL (Dutch), PL (Polish), PT-PT, PT-BR (Portuguese) and RU (Russian). It can be useful to make button labels and help texts (tooltips) informal, since in these cases it is not the user who is addressed, but the program, so to speak. It can make a difference whether a button is labeled *Save file* or *Save the file*. The same applies to information for the user, which in many cases can be provided with a formal salutation.

Notes

It often makes sense to use German instead of English as the source language, since the infinitive is more clearly recognizable in German and is therefore translated more unambiguously.

Give the constants names that indicate their intended use. For example, informative texts could contain the word *Info* to mark them for translation in the *Constants to ignore* or *Constants to translate* field for formal translation.

Settings

The settings in the program window are self-explanatory. They are automatically saved at the end of the program.

The **Do not translate online, only simulate** setting is used to easily check the translation process. It displays the original text and the target text side by side in one window. In order not to unnecessarily load the character contingent of DeepL, the text for the simulation is not translated, but is also displayed in the window of the target file as original text. The text to be translated is highlighted in color for better display.

Sets of settings (12), (15)

In many cases, it is convenient to be able to save settings for different tasks in different sets or profiles in order to recall them later.

This button opens a self-explanatory dialog for managing such sets.

The bottom of the program window shows which set is currently being used.

Regular expressions

Regular expressions are used in the input fields of this app. A regular expression is a string that is used to describe a set of strings according to certain syntactic rules. Regular expressions are used as filter criteria in text search by comparing the text with the regular expression pattern. For example, it is possible to find all words in a word list that begin with *T* and end with *S* without having to explicitly specify the letters in between or their number.

The Internet offers a wealth of information on the structure and use of regular expressions.

If you want to enter normal text in such a field that should not be interpreted as a regular expression, start the text with an apostrophe. Example: `'Not a regular expression`. If you want to use an apostrophe as the first character in a regular expression, you must precede it with a backslash. Example: `\'A meaningless regular expression`.

Log file

Translate Files with AI maintains a log file that keeps track of each transaction. The log file has the standard format for .log files and is located in the *logs* folder below the user folder.

Contact

tomsidee

Thomas Osthege

Siegstraße 24

D-51570 Windeck-Rosbach

Germany

Fon: +49 2292 9599730

Fax: +49 2292 9599733

contact@eurocomp.info

translate.tomsidee.de

If you have any questions or miss a feature, please email me using the app's contact function. I appreciate feedback and of course positive reviews in the App Store. Please do not criticize the software with negative reviews. I can't respond to that and it only frustrates. It also does not contribute to the further development or improvement of the app.