

1. Complete the translations

Note.

The author of `tutodoc` manages the French and English versions of the translations.

Caution.

Although we're going to explain how to translate the documentation, it doesn't seem relevant to do so, as English should suffice these days.^a

^aThe existence of a French version is simply a consequence of the native language of the author of `tutodoc`.

The translations are roughly organized as in figure 1 where only the folders important for the translations have been “opened”.¹ **A little further down, the v. section explains how to add new translations.**

i. The `fr` and `en` folders

These two folders, managed by the author of `tutodoc`, have the same organization; they contain files that are easy to translate even if you're not a coder.

ii. The `changes` folder

This folder is a communication tool where important changes are indicated without dwelling on minor modifications specific to one or more translations.

iii. The `status` folder

This folder is used to keep track of translations from the project's point of view. Everything is done via well-commented `YAML` files, readable by a non-coder.

iv. The `README.md` and `LICENSE.txt` files

The `LICENSE.txt` file is aptly named, while the `README.md` file takes up in English the important points of what is said in this section about new translations.

v. New translations

Important.

The `api` folder contains translations relating to the functionalities of `tutodoc`. Here you'll find `TXT` files for editing with a text or code editor, but not with a word processor. The content of these files uses commented lines in English to explain what `tutodoc` will do; these lines begin with `//`. Here's an extract from such a file, where translations are made after each `=` sign, without touching the preceding, as this initial piece is used internally by the `tutodoc` code.

```
// #1: year in format YYYY like 2023.
// #2: month in format MM like 04.
// #3: day in format DD like 29.
date = #1-#2-#3

// #1: the idea is to produce one text like
```

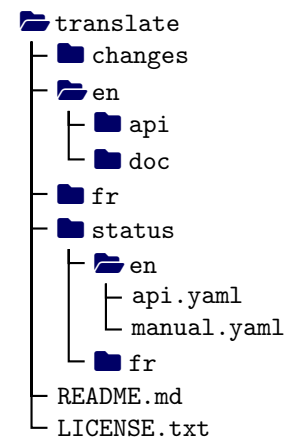


Figure 1: Simplified view of the translation folder

¹This was the organization on October 5, 2024, but it's still relevant today.

```
//      "this word means #1 in english".  
in_EN = #1 in english
```

Note.

The `doc` folder is reserved for documentation. It contains files of type `TEX` that can be compiled directly for real-time validation of translations.

Warning.

Only start from one of the `fr` and `en` folders, as these are the responsibility of the `tutodoc` author.

Let's say you want to add support for Italian from files written in English.²

Method 1 : git.

1. Obtain the entire project folder via <https://github.com/bc-tools/for-latex/tree/tutodoc>. Do not use the `main` branch, which is used to freeze the latest stable versions of projects in the single <https://github.com/bc-tools/for-latex> repository.
2. In the `tutodoc/contrib/translate` folder, create a `it` copy of the `en` folder, with the short name of the language documented in the page "*IETF language tag*" from Wikipedia.
3. Once the translation is complete in the `it` folder, you'll need to communicate it via <https://github.com/bc-tools/for-latex/tree/tutodoc> using a classic `git push`.

Method 2 : communicate by e-mail.

1. By e-mail with the subject "*tutodoc - CONTRIB - en FOR italian*", request a version of the English translations (note the use of the English name for the new language). Be sure to respect the subject of the e-mail, as the author of `tutodoc` automates the pre-processing of this type of e-mail.
2. You will receive a folder named `italian` containing the English version of the latest translations. This folder will be the place for your contribution.
3. Once the translation is complete, you will need to compress your `italian` file in `zip` or `rar` format before sending it by e-mail with the subject "*tutodoc - CONTRIB - italian*".

²As mentioned above, there is no real need for the `doc` documentation folder.