

---

# Translating the messages of XMLmind XML Editor

Hussein Shafie, XMLmind Software <xmleditor-support@xmlmind.com>

June 17, 2024

## Abstract

This document explains how to translate the messages (menu labels, button labels, error messages, etc) of XMLmind XML Editor using XMLmind XML Editor itself.

## Table of Contents

1. Using XMLmind XML Editor to translate the messages .....	1
2. Using software other than XMLmind XML Editor to translate the messages .....	3
3. How to correctly translate a message .....	6
3.1. Menu mnemonics .....	7
3.2. Captions found in the main tool bar .....	8
A. The <b>translatexx</b> command-line utility .....	8

## 1. Using XMLmind XML Editor to translate the messages

1. Use **Options** → **Install Add-ons** to install the add-on called "*Translate XMLmind XML Editor*" then restart XMLmind XML Editor (**XXE** for short).
2. Select **File** → **Translation** → **New** to create a new translation from scratch.

Or select "**File** → **Translation** → **Open Message Jar or Add-on Zip**" to update an existing translation. The existing translation is either found in a message Jar (example: `fr_translation.jar`) or in an add-on Zip (example: `fr_translation.zip`, itself containing `fr_translation/fr_translation.jar` and `fr_translation/fr_translation.xxe_addon`).

In both cases, you'll be prompted for a *target language*. This target language is specified using a two-letter ISO 639-1 language code (examples: `fr`, `de`, `es`, `it`). Three-letter ISO 639-2 language codes are *not* supported (examples: `fra`, `deu`), neither are the combinations of a language code and a country code (examples: `fr-CA`, `de-AT`).

Both the two aforementioned menu items create a new<sup>1</sup> XLIFF document. This document cannot be modified, except for the messages in the target language.

---

<sup>1</sup>This means that the first time you'll save the XLIFF document to disk, you'll be prompted for a save filename.



3. Use the following toolbar buttons:  (first empty field),  (last empty field),  (previous empty field),  (next empty field) to navigate between the messages to be translated.

These buttons automatically skip the low-level, technical, messages which are not worth translating. In the above screenshot, section "com/xmlmind/guiutil/Messages.properties", expanded, having a green title, contains messages which are worth translating, while section "com/xmlmind/xml/dtd/Messages.properties", collapsed, having a red title, contains messages which are not worth translating.

4. You may want to use the  (translate selection or message) toolbar button in order to get some help. This button invokes your web browser in order to open the Google Translate page containing the translation of the current message<sup>2</sup> to the target language.
5. Translate all the messages found the "recommended sections". A "recommended section" is expanded and has a green title. In contrast, an "optional section" is collapsed and has a red title. See Section 3, "How to correctly translate a message" [6].
6. After translating the messages, you may want to review what you did using the following toolbar buttons:  (first filled field),  (last filled field),  (previous filled field),  (next filled field).
7. Testing your translation is easy: select "File → Translation → Build and Install the Add-on to Test it" then restart **XXE** as instructed.
8. Once you have finished testing your translation, you may select "File → Translation → Uninstall the Test Add-on" then restart **XXE** as instructed.

Of course, you may want, on the contrary, to keep this installed translation add-on to benefit yourself from your hard work. Note that this so-called "Test Add-on" is in fact a regular add-on which can be removed at any time, normally, using "Options → Install Add-ons".

9. Finally select "File → Translation → Save As Add-on Zip" to save the XLIFF document as a ready-to-deploy add-on Zip (example: fr\_translation.zip).

You'll be prompted for the name of the author of the add-on. A string starting with "mailto:" (example: "mailto:jane@doe.name") gets a special treatment. A string starting with "<html>" and

<sup>2</sup>The current message is the message containing the caret (also called "insertion cursor"). Note that if there is an explicit text selection, then the current message is ignored and instead the text selection is submitted to Google Translate.

followed by well-formed, properly escaped, XHTML (example: "<html><a href='http://www.doe.name/'>Jane Doe & al.</a>") also gets a special treatment. Otherwise, the string is understood to be plain text (example: "Jane Doe & al.").

### Translating *all* messages, that is, not only those contained in the “recommended sections”

- Unchecking the checkbox depicted below and found in the toolbar associated to the "Translate XMLmind XML Editor" configuration allows to visit *all* empty messages fields, whether these fields are contained in the “recommended sections” or in the “optional sections”.



- If you want to translate the messages contained in some add-ons which are not included by default in the stock **XXE** distributions ("RenderX XEP XSL-FO processor plug-in", "Hunspell Spell Checker", etc), then you'll have to first download and install these add-ons<sup>3</sup> prior to using the "Translate XMLmind XML Editor" add-on.

## 2. Using software other than XMLmind XML Editor to translate the messages

As an example, we'll explain how to translate **XXE** messages to French<sup>4</sup>.

1. Use **Options** → **Install Add-ons** to install the add-on called "Translate XMLmind XML Editor" then restart XMLmind XML Editor (**XXE** for short).

After installing the add-on, the **translatexxe** command-line utility is found in the `translate_xxe/` subdirectory of one of the two `addon/` directories [6] scanned by **XXE** during its startup.

Use `translatexxe.bat` on Windows and the `translatexxe` shell script on Linux and on the Mac.

2. Create a new translation from scratch:

```
C:\Prog...\translate_xxe> translatexxe -l fr -e windows-1252 \  
-m fr.txt
```

Or update an existing translation. The existing translation is either found in a message Jar (example: `fr_translation.jar`) or in an add-on Zip (example: `fr_translation.zip`, itself containing `fr_translation/fr_translation.jar` and `fr_translation/fr_translation.xxe_addon`):

```
C:\Prog...\translate_xxe> translatexxe -l fr -e windows-1252 \  
-m fr.txt \misc\fr_translation.jar
```

or:

```
C:\Prog...\translate_xxe> translatexxe -l fr -e windows-1252 \  
-m fr.txt \misc\fr_translation.zip
```

<sup>3</sup>Using **Options** → **Intall Add-ons**.

<sup>4</sup>This is just en example because **XXE** has already been translated to French.

All the above commands generate a text file. If you intend to use a specialized translation editor<sup>5</sup> rather than a text editor, you may want to generate an XLIFF file. In such case, specify a message file having a ".xlf", ".xliff" or ".xml" extension. Example:

```
C:\Prog...\translate_xxe> translategxe -l fr \  
-m fr.xlf \misc\fr_translation.jar
```

Message file `fr.txt` looks like this:

```
+++++++com/xmlmind/guiutil/Messages.properties❶  
#####❷  
  
### ChooseFile ---❸  
  
openFile❹=Open File❺  
openFile=❻  
  
CF.overwriteFile=Overwrite existing file\n\  
{0}?"?  
CF.overwriteFile=  
  
-----com/xmlmind/guiutil/Messages.properties❷  
+++++++com/xmlmind/xml/dtd/Messages.properties❸  
  
invalidDefaultValue=default value "{0}" of attribute "{1}" of "{2}" is invalid: {3}  
invalidDefaultValue=  
  
-----com/xmlmind/xml/dtd/Messages.properties
```

- ❶ Beginning of first section: messages contained in the `com/xmlmind/guiutil/Messages.properties` resource file. As a translator, you can ignore these markers.
- ❷ The title of the section is underlined using the '#' character. This means that it is recommended to translate the messages found in this section.
- ❸ A comment added by the author of the English messages. As a translator, you can ignore these comments.
- ❹ The ID of the message is "openFile".
- ❺ English text of the message having "openFile" as its ID.
- ❻ Placeholder for the French text of the message having "openFile" as its ID.
- ❼ End of first section.
- ❽ Beginning of next section: messages contained in the `com/xmlmind/xml/dtd/Messages.properties` resource file. Notice that the title of this section is not underlined. This means that this section does not contain messages which are worth translating.

---

<sup>5</sup>Example: OmegaT, a free translation memory application written in Java™.

3. Translate all the messages found the “recommended sections” (underlined section title). See Section 3, “How to correctly translate a message” [6].

```
+++++++com/xmlmind/guiutil/Messages.properties
#####

### ChooseFile ---

openFile=Open File
openFile=Ouvrir Fichier

CF.overwriteFile=Overwrite existing file\n\
"{0}"?
CF.overwriteFile=Écraser le fichier existant\n\
"{0}"?

-----com/xmlmind/guiutil/Messages.properties
+++++++com/xmlmind/xml/dtd/Messages.properties

invalidDefaultValue=default value "{0}" of attribute "{1}" of "{2}" is invalid: {3}
invalidDefaultValue=

-----com/xmlmind/xml/dtd/Messages.properties
```

In a text file, in addition to normal characters, it is possible to use the following escape sequences: '\n' (newline), '\t' (tab), '\uxxxx' (where xxxx is the code of a Unicode character in 4-digit hexadecimal notation). '\uxxxx' is especially useful because it's the only way to specify characters not supported by the encoding passed to the `-e` command-line option. Example: '\u2192' (right arrow).

4. Test your translation by first converting the text file to a message Jar:

```
C:\Prog...\translate_xxe> translutexxe -l fr -e windows-1252 \
-j ..\addon\fr_translation.jar fr.txt
```

Make sure that the message Jar is found inside one of the two `addon/` directories [6] scanned by **XXE** during its startup. Then start **XXE**, clear or disable its *Quick Start cache*<sup>6</sup> and restart **XXE**.

If, after restarting **XXE**, you don't see its user interface translated to the target language, then this means that the locale of your computer (example: `en-US`) does not correspond to the target language (example: `fr`). When this is the case, use **Options** → **Preferences, General** section, to change the locale used by **XXE** from "**(default)**" to the locale corresponding to the target language (example: "**français**").

5. Finally convert the text file to a ready-to-deploy add-on Zip:

```
C:\Prog...\translate_xxe> translutexxe -l fr -e windows-1252 \
-p \misc\fr_translation.zip fr.txt
```

You may want to specify for the name of the author of the add-on using the `-a` command-line option. A string starting with "mailto:" (example: "mailto:jane@doe.name") gets a special treatment. A string starting with "<html>" and followed by well-formed, properly escaped, XHTML (example:

---

<sup>6</sup>See **Options** → **Preferences, Advanced|Cached data**.

"<html><a href='http://www.doe.name/'>Jane Doe & al.</a>") also gets a special treatment. Otherwise, the string is understood to be plain text (example: "Jane Doe & al.").

```
C:\Prog...\translate_xxe> translatexxe -l fr -e windows-1252 \  
-a "mailto:jane@doe.name" -p \misc\fr_translation.zip fr.txt
```

### XMLmind XML Editor *addon/* directories

The *addon/* directory is the place where **XXE** finds its extensions, whatever is the type of this extension: configurations, translations to languages other than English, spell checker dictionaries, plug-ins of all sorts, etc.

When **XXE** is started, it scans two *addon/* directories, searching them for extensions.

*xxe\_install\_directory/addon/*

The system *addon/* directory.

*User\_preferences\_directory/addon/*

The user *addon/* directory.

*User\_preferences\_directory* is:

- *\$HOME/.xxe10/* on Linux.
- *\$HOME/Library/Application Support/XMLmind/XMLEditor10/* on the Mac.
- *%APPDATA%\XMLmind\xMLEditor10\* on Windows. Example: *C:\Users\john\AppData\Roaming\xMLmind\xMLEditor10\*.

If you cannot see the "AppData" directory using Microsoft Windows File Manager, turn on **Tools>Folder Options>View>File and Folders>Show hidden files and folders**.

## 3. How to correctly translate a message

Let's use the following message as an example of what to do:

```
Overwrite existing file  
{0}?"
```

If you have used the **translatexxe** command-line utility [8] to generate a text file, the above message will look like this:

```
Overwrite existing file\n\  
{0}?"
```

Let's suppose you have translated it to French as follows:

```
écraser le fichier existant {0}
```

The above translation has several mistakes:

- Word "écraser" must be capitalized.

The word capitalization used in the original message must be reproduced in the translated message.

- Variable {0} must be quoted and the sentence must end with a question mark.

The punctuation used in the original message must be reproduced in the translated message.

- Words "fichier existant" must be followed by a newline character.

If the original message is split into several lines, then the translated message must be split into the same number of lines. Moreover, these lines must have roughly the same length as those of the original message. Failing to do so may cause **XXE** to display very wide dialog boxes.

The correct translation is:

```
Écraser le fichier existant
"{0}"?
```

If you have used the **translatexx** command-line utility [8] to generate a text file, the correct translation is:

```
Écraser le fichier existant\n\
"{0}"?
```

Note that what follows, though less readable, is correct too:

```
Écraser le fichier existant\n"{0}"?
```

### 3.1. Menu mnemonics

Some messages contain one underscore character ('\_'). Menu examples: "\_File", "\_Edit", "\_Search". Menu item examples: "\_New", "\_Open...", "\_Save", "Save \_As...". Such messages are labels for menus, menu items and buttons. The underscore is located just before the letter which is used as a *mnemonic*.

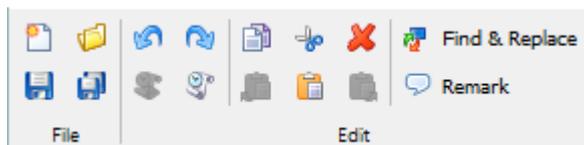
Recommendations:

- The location of the underscore depends on the target language.

Example: in English, **Alt+F** opens the **File** menu because the underscored letter is F (i.e. "\_File"). In German, the **File** menu is called **Datei** and the underscored letter could be D (i.e. "\_Datei").

- Only letters A to Z, a to z and digits 0 to 9 may be underscored. Do not underscore accented letters (examples: é, à).
- Do not underscore the same letter, *no matter its case*, twice in the same GUI component (same menu bar, same menu, same dialog box, etc).
- Because all the above constraints are quite difficult to satisfy, it is better not to use this facility at all rather than using it inconsistently. *In conclusion, if you find that specifying mnemonics this way is too tedious, then do not hesitate to drop the underscore character from all the messages you'll translate.*

## 3.2. Captions found in the main tool bar



The main tool bar contains button groups having captions (e.g. "**File**", "**Edit**"). Some of the buttons have short descriptions (e.g. "**Find & Replace**", "**Remark**").

The corresponding messages must be kept very short — at most two words long (about 20 characters) — and must have all their words capitalized. For example, translate "Find & Replace" to "Rechercher & Remplacer" and not by "Rechercher et remplaceur".

## A. The `translatexxe` command-line utility

After installing the "*Translate XMLmind XML Editor*" add-on, the `translatexxe` command-line utility is found in the `translate_xxe/` subdirectory of one of the two `addon/` directories [6] scanned by **XXE** during its startup.

Use `translatexxe.bat` on Windows and the `translatexxe` shell script on Linux and on the Mac.

The `translatexxe` command-line utility is auto-documented. Execute `translatexxe -?` in order to print a short help text similar to what follows:

```
translatexxe [-v] [-l two_letter_language_code] [-e message_file_encoding]
  [-a author_of_the_translation]

  -m out_message_file [in_jar_file|in_zip_file]
OR
  -j out_jar_file in_message_file
OR
  -p out_zip_file in_jar_file|in_message_file
```

- Use "`-m out_message_file`" to create a XLIFF (`.xlf`) or text (`.txt`) file containing the **XXE** messages to be translated.
- If you have already translated **XXE** messages to, say, Italian, and want to update your translation, use "`-m out_message_file it_translation.jar`" or "`-m out_message_file it_translation.zip`".
- Use "`-j out_jar_file in_message_file`" to convert your message file "`in_message_file`" to message Jar "`out_jar_file`".
- After using option "`-j`" to test your message jar, you may want to use "`-p out_zip_file in_jar_file`" to package the message Jar as a `.zip` file for use by **XXE** add-on manager.

You may want to use option "`-a`" in conjunction with this "`-p`" option. By default, the author of the add-on is taken from your login name.

- In all cases, "`-e message_file_encoding`" may be used to specify the encoding of the input or output message file.

By default, the encoding used for a text file is the native encoding of the platform (e.g. "windows-1252") and the encoding used for an XLIFF file is "UTF-8".

- Use "-l two\_letter\_language\_code" to specify the target language of your translation of **XXE** messages.

By default, this language code is inferred from the first two letters of the basename of the message Jar file. Example: "-j it\_translation.jar msg.txt" implies that you have translated **XXE** messages to Italian.