How to translate Termite

The international language support in Termite is based on the "Rosette" library. There are three sections in this guide. It starts with instructions for translation without any additional tools (except for a plain text editor, but your operating system is guaranteed to come with a suitable one). For translators, there are environments that are more efficient —the second section covers how the Rosette catalog can be converted to and from the formats used by these special environments. The third section contains general remarks that apply to both ways of working.

With no help

A Rosette catalog contains the translations for all languages in a single file. It is called the "catalog" and it is a text file. It can be edited with a plain text editor, such as Notepad. Obviously, we advise you to use a more powerfull text editor, but if Notepad is all that you have, it is perfectly suitable.

A translation block starts with a line in the "key language" (this is usually English), followed by a list of translations in other languages. All languages are indicated by a 2-letter code, "en" for English, "nl" for Dutch, "fr" for French, etc.

Each message must be contained on a single line (or row). To break the line, insert the "\n" character pair at the location where the break must occur. If a backslash appears in the text, you should double it. This is called *escaping* a backslash in Rosette. The most common *special characters* in Rosette are:

\n	A line break ("newline")
\r	A carriage return
\t	A TAB character
\\	The backslash character itself (\)
ddd	A character code of 1 to 3 digits; the code must be in decimal
\x*dd*	A character code of 1 or 2 digits; the code must be in hexadecimal

If a message is formulated differently depending on whether there are "plural forms", you give both the "general" message and the exceptions —the exceptions have a *count* in square brackets behind the language code. For European languages, the *general message* is usually the plural form and the exception is the *singular form*.

With some help

There exist various programs that are specifically designed for translating text. However, only very few are suitable for translating the strings in software products —mainly because of a lack of consensus on how internationalizing of software should be done.

There are no known Computer Aided Translations (CAT) tools that work directly on Rosette catalogs. You can use CAT tools, however, by first exporting part of the catalog in a new format, and importing it back into the catalog after translation. Rosette provides tools to perform the export, import and verification operations —and the essential tools are now freely available. These tools are available both as command line utilities, and as a windowed utility. This document only covers the windowed utility.
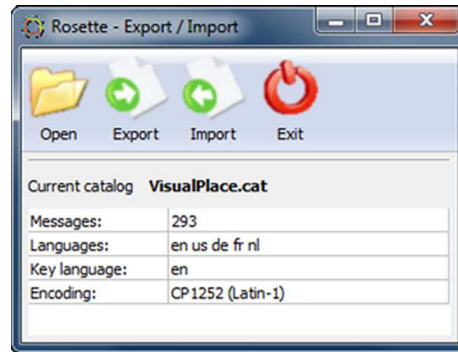
FIGURE 1: *The Rosette export/import utility*

The windowed utility has four buttons for the main operations: opening a Rosette catalog, exporting a single translation (or template for translation) from that catalog in a selected format, importing a translated partial catalog/message file, and closing the utility.

Before starting, it is always a good idea to ensure that you have a backup of the original catalog, and of any other files that you work on. Since Rosette catalogs (and intermediate files that it uses) are all plain text files, a *better* idea is to use a version control system.

After opening a catalog, the utility shows some statistics about the catalog, among which the currently supported languages (the languages that are already in the catalog) and the key language.

When exporting a file, the utility shows a dialog where you need to set two options: the format for the output file and the language to export. The format depends on the translation editor (CAT tool) that you will use (more on this below). The language to export can either be one of the languages that are already in the catalog or, more likely, a new language. To export a new language, you need to type in the two letter code for the language. So for example, if you wish to make a translation for Spanish, you would specify "`es`" as the language to export.

When importing a translated file back into the catalog, no options need to be set. The Rosette utility detects the appropriate parameters.

Below are two example scenarios for translating Rosette catalogs: with a PO editor and using Google's on-line "Translator Toolkit".

• Using a PO editor

`gettext` is a library and toolkit for internationalizing software (its use is restricted to "open-source" programs, because of its license). A few programs exist to edit the translation files used by `gettext`. These files are called "PO" files, software to edit PO files are "PO editors".

A difference with Rosette catalogs is that a PO file contains a translation for only one language, whereas a catalog contains the translations for *all* supported languages. Therefore, a single language must be extracted from the catalog, before starting the translation.

With `gettext`, it is common practice to use the two-letter language code as the filename, with extension "`.po`". For the Rosette "import" functionality, this naming convention is mandatory, as the PO format does not have a header or message fields that encode the key and/or target languages.

When adding a translation for a new language, you have to type in the two-letter code for the new language and choose the format "PO". The export operation now creates a file with only the strings in the key language and empty translations. In `gettext` terminology, this is a PO Template, or "POT" file. Rosette does not make the distinction between PO and POT files —a POT file is just a PO file that happens to contain zero translated strings.[1]

Issues like "escaping" characters and handling plural forms (see the previous section) are often handled by the PO editor and/or the conversion tools.

Some PO editors offer automatic conversion (or compilation) to "MO" format. This is not needed and not useful for Rosette. The MO format is a binary representation of the PO format —and it is incompatible with Rosette's binary format.

• Using Google Translator Toolkit

Google provides a "Translator Toolkit" for translating texts or web pages —an on-line service that combines computer translations and translation memory. The Translator Toolkit uses a side-by-side editor for the original and translated texts. The editor requires that the source file contains strings in only a single language; the output file will also contain only a single language (i.e. the translation). These two files (source and output) are synchronized internally in the Translator Toolkit.

To use Google's Translator Toolkit with Rosette, the first step is to export the key language to a temporary file. This file is then uploaded to the Translator Toolkit and translated. The output from the translation can be imported back into the catalog.

To export the temporary file, choose the key language as the language to export and the "HASH" format. This will give you a file with messages in the key language (usually English), where each message is preceded by an eight-digit hexadecimal code —the hash. This is the file that you will upload and translate.

After uploading, in the left panel of the editor of the Translator Toolkit, you will see lines with:

```
#127FA68E
en: This is not a pipe
```

In the right panel, there are the same strings, already machine translated into the target language. If the target language is French, the string might be:

```
#127FA68E
en: Ce n'est pas une pipe
```

You can correct "Ce" into "Ceci". With a global search & replace operation, you should also replace all occurrences of "en:" to "fr:". The comment with the hash code (`#`) should not be altered or removed; this code is required to to merge the translated file back. The corrected snippet is:

```
#127FA68E
fr: Ceci n'est pas une pipe
```

---

[1] gettext makes this distinction because its "xgettext" utility always creates a single empty POT file to send to the translators, and translators then use "msginit" to merge previously made translations in the template to create a PO file. Rosette's "rsverify" utility, the equivalent of "xgettext" always *modifies* the catalog rather than creating an empty one from scratch. Therefore, what you send to the translators already contains the translations from previous versions.

General remarks

The text strings may contain placeholders for values, amounts or names. These placeholders have the form of "`%1s`": they start with a percentage sign, optionally followed by a number, a letter or both. At run-time, the application replaces the placeholders by actual data, such as a name, a file path, or a number. When translating messages, you should copy the placeholders as they are. When the percentage sign is followed by a number, you may optionally rearrange the placeholders in the translation —that is, in the translation, placeholder "`%2s`" may become before "`%1s`". The message in the key language will give an indication as to what each placeholder stands for.

A few translations of the software are already provided. When starting a translation, you can learn by example, by looking how the existing translations handled things. Some background information, including tips and examples, can be found in a freely available white paper on Rosette, which you can find on the CompuPhase web site.[2]

After adding your translations, you can test the results immediately by launching Termite and choosing your language. Strings that you have not yet translated will appear in the key language (English).[3]

- **The messages and menu text**

All messages, menus and dialog texts are in the file "`Termite.cat`". This file is in the "`doc`" subdirectory below where Termite is installed.

Near the top of the file is a string that starts with the keyword "`$Languages`". You should add the new language to this list. For the language, you must give both the language name and a two-letter abbreviation. For example, if a catalog already contains English (as the key language) and German, and you wish to add a French translation, the catalog will contain:

```
en: $Languages English (en); Deutsch (de)
de: *
```

You then adjust this to:

```
en: $Languages English (en); Deutsch (de); Français (fr)
de: *
fr: *
```

The reason for the lines "`de: *`" and "`fr: *`" is that for this particular message, Termite will *only* use the message in the key language. There should be strings present for the other languages as well (to avoid warning messages from Rosette), but their contents are never used. It has therefore become common practice to translate this particular message to a "`*`".

For all other messages in the file, you should add another message in the new language. The 2-letter code for the new language must be the same as the one that you gave at the `$Languages` list.

Never change the string in the key language ("`en`").

---

[2] *Rosette — Internationalization through message catalogs*, available on the CompuPhase web site.

[3] Depending on your environment, Rosette may fail to import partially translated files. This applies especially to PO editors. That is, you may need to complete the translation before being able to test-drive it.

Apart from the escape characters described on , the strings describing menu items and buttons for the user interface may contain a "&" prefixed to the word or even in the middle of the word. This & character serves to indicate the *shortcut* letter. In the translation, there should preferably also be a shortcut letter, but it does not need to be on the same letter. For standard menu items and dialog fields, you may look at how other applications define the shortcuts, for application-specific menu items/fields, you can choose any letter that does not cause conflicts with other items/fields. If no suitable letter can be found, you can leave the shortcut out of the translation.