Localization

- What is localization?
- Translating WordPress, Plugins and Themes
- Translating Themes and Plugins
  - Localization files
  - POT (Portable Object Template) files
  - PO (Portable Object) files
  - MO (Machine Object) files
  - Generating POT file
    - WP-CLI
    - Poedit
    - Grunt Tasks
  - Translate PO file
  - Generate MO file
    - Command line
    - Converting multiple PO files at once
    - Poedit
    - Grunt task
- Tips for Good Translations
  - Don’t translate literally, translate organically
  - Try to keep the same level of formality (or informality)
  - Don’t use slang or audience-specific terms
  - Read other software’s localizations in your language
- Using Localizations
- Resources

What is localization?

Localization describes the process of translating a software and adapting its strings to a specific location. Localization is abbreviated as `l10n` (because there are 10 letters between the l and the n.)

The process of localizing software has two steps. The first step is when the developers provide a mechanism and method for the eventual translation of the program and its interface to suit local preferences and languages for users worldwide. This process is internationalization (i18n). WordPress developers have done this already, so in theory, WordPress can be used in any language.

The second step is the actual **localization** (l10n), the process by which the text on the page and other settings are translated and adapted to another language and culture, using the framework prescribed by the developers of the software. WordPress has already been localized into many languages (see the list of polyglots teams for more information).

Translating WordPress, Plugins and Themes

If you want to help translating WordPress, or any Theme or Plugin hosted in WordPress.org themes and plugins directories, you should go to Translate WordPress.

There you will get to know all the translators teams and learn about translate.wordpress.org, where you can work on translations online and in collaboration with thousands of translators around the world.

Translating Themes and Plugins

If you want to translating plugins and themes that are not hosted on WordPress.org, or if, for any reason, you want to translate themes or plugins offline and directly in the plugins/themes files, you can do this creating and editing Localization Files.

Localization files

There are three types of Localiztion files you need in order to translate your plugin/theme:

- POT files: a template file with all your original strings
- PO files: editable file with the translations to one language (one file per language)

- MO files: compiled versions of the PO files, actually used by the application

POT (Portable Object Template) files

This file contains the original strings (in English) in your plugin/theme. Here is an example `POT` file entry:

```
#: plugin-name.php:123
msgid "Page Title"
msgstr ""
```

PO (Portable Object) files

Every translator will take the `POT` file and translate the `msgstr` sections in their own language. The result is a `PO` file with the same format as a `POT`, but with translations and some specific headers. There is one PO file per language.

MO (Machine Object) files

From every translated `PO` file a `MO` file is built. These are machine-readable, binary files that the gettext functions actually use (they don't care about .POT or .PO files), and are a "compiled" version of the PO file. The conversion is done using the `msgfmt` tool. In general, an application may use more than one large logical translatable module and a different `MO` file accordingly. A text domain is a handle of each module, which has a different `MO` file.

Generating POT file

The POT file is the one you need to hand to translators, so that they can do their work. The POT and PO files can easily be interchangeably renamed to change file types without issues. It is a good idea to offer the POT file along with your plugin/theme, so that translators won't have to ask you specifically about it. There are a couple of ways to generate a POT file for your plugin:

WP-CLI

Install WP-CLI and use the `wp i18n make-pot` command according to the documentation.

Open command line and run the command like this:

```
wp i18n make-pot path/to/your-plugin-directory
```

Poedit

You can also use Poedit locally when translating. This is an open source tool for all major OS. The free Poedit default version supports manual scanning of all source code with Gettext functions. A pro version of it also features one-click scanning for WordPress plugins. After generating the po file you can rename the file to POT. If a mo was generated then you can delete that file as it is not needed. If you don't have the pro version you can easily get the Blank POT and use that as the base of your POT file. Once you have placed the blank POT in the languages folder you can click "Update" in Poedit to update the POT file with your strings.

internationalization-localization-03

Grunt Tasks

There are even some grunt tasks that you can use to create the POTs. grunt-wp-i18n & grunt-pot To set it up you need to install node.js. It is a simple installation. Then you need to install grunt in the directory that you would like to use grunt in. This is done via command line. An example Grunt.js and package.json that you can place in the root of your plugin. You can the grunt tasks with a simple command in the command line.

Translate PO file

There are multiple ways to translate a PO file.

The easiest way is to use Poedit when translating. This is an open source tool for all major OS. The free Poedit default version supports manual scanning of all source code with Gettext functions. A pro version of it also features one-click scanning for WordPress plugins and themes.

You can also use a text editor to enter the translation. In a text editor the strings will look like this.

```
#: plugin-name.php:123
msgid "Page Title"
msgstr ""
```

You can enter the translation between the quotation marks. For the German translation, the final result would look like this.

```
#: plugin-name.php:123
msgid "Page Title"
msgstr "Seitentitel"
```

A third option is to use an online translation service. The general idea is that you upload the POT file and then you can give permission to users or translators to translate your plugin. This allows you to track the changes, always have the latest translation and reduce the translation being done twice.

Here are a few tools that can be used to translate PO files online:

- Transifex
- WebTranslateIt
- Poeditor
- Google Translator Toolkit
- GlotPress

The translated file is to be saved as `my-plugin-{locale}.mo`. The locale is the language code and/or country code you defined in the constant `WPLANG` in the file `wp-config.php`. For example, the locale for German is `de_DE`. From the code example above the text domain is 'my-plugin' therefore the German MO and PO files should be named `my-plugin-de_DE.mo` and `my-plugin-de_DE.po`. For more information about language and country codes, see Installing WordPress in Your Language.

Generate MO file

Command line

A program `msgfmt` is used to create the MO file. `msgfmt` is part of Gettext package. Otherwise command line can be used. A typical `msgfmt` command looks like this:

Unix Operating Systems

```
msgfmt -o filename.mo filename.po
```

Windows Operating Systems

```
msgfmt -o filename.mo filename.po
```

Converting multiple PO files at once

If you have a lot of `PO` files to convert at once, you can run it as a batch. For example, using a `bash` command:

Unix Operating Systems

```
# Find PO files, process each with msgfmt and rename the result to MO
for file in `find . -name "*.po"` ; do msgfmt -o ${file/.po/.mo} $file ;
done
```

Windows Operating Systems

For Windows, you need to install Cygwin first.

Create a file called potomo.sh with the following content:

```
#! /bin/sh # Find PO files, process each with msgfmt and rename the result to MO
for file in `/usr/bin/find . -name '*.po'` ; do /usr/bin/msgfmt -o ${file/.po/.mo} $file ;
done
```

You can then run this command in the command line.

```
cd C:/path/to/language/folder/my-plugin/languages
C:/cygwin/bin/bash -c /cygdrive/c/path/to/script/directory/potomo.sh
```

Poedit

`msgfmt` is also integrated in Poedit allowing you to use it to generate the MO file. There is a setting in the preferences where you can enable or disable it.

internationalization-localization-04

Grunt task

There is grunt-po2mo that will convert all of the files.

Tips for Good Translations

Don’t translate literally, translate organically

Being bi- or multi-lingual, you undoubtedly know that the languages you speak have different structures, rhythms, tones, and inflections. Translated messages don’t need to be structured the same way as the English ones: take the ideas that are presented and come up with a message that

expresses the same thing in a natural way for the target language. It’s the difference between creating an equal message and an equivalent message: don’t replicate, replace. Even with more structural items in messages, you have creative license to adapt and change if you feel it will be more logical for, or better adapted to, your target audience.

Try to keep the same level of formality (or informality)

Each message has a different level of formality or informality. Exactly what level of formality or informality to use for each message in your target language is something you’ll have to figure out on your own (or with your team), but WordPress messages (informational messages in particular) tend to have a politely informal tone in English. Try to accomplish the equivalent in the target language, within your cultural context.

Don’t use slang or audience-specific terms

Some amount of terminology is to be expected in a blog, but refrain from using colloquialisms that only the “in” crowd will get. If the uninitiated blogger were to install WordPress in your language, would they know what the term means? Words like pingback, trackback, and feed are exceptions to this rule; they’re terminology that are typically difficult to translate, and many translators choose to leave in English.

Read other software’s localizations in your language

If you get stuck or need direction, try reading through the translations of other popular software tools to get a feel for what terms are commonly used, how formality is addressed, etc. Of course, WordPress has its own tone and feel, so keep that in mind when you’re reading other localizations, but feel free to dig up UI terms and the like to maintain consistency with other software in your language.

Using Localizations

Place the localization files in the language folder, either in the plugin `languages` folder or as of WordPress 3.7 in the plugin `languages` folder normally under `wp-content`. The full path would be `wp-content/languages/plugins/my-plugin-fr_FR.mo`.

As of WordPress 4.0 you can change the language in the “General Settings”. If you do not see any option or the language that you want to switch to then do the following steps:

- Define WPLANG inside of wp-config.php to your chosen language. For example, if you wanted to use french, you would have:

```
define ('WPLANG', 'fr_FR');
```

- Go to `wp-admin/options-general.php` or “Settings” -> “General”
- Select your language in “Site Language” dropdown
- Go to `wp-admin/update-core.php`
- Click “Update translations”, when available
- Core translations files are downloaded, when available

Resources

- Creating .pot file for your theme or plugin
- How To Internationalize WordPress Plugins
- Translating Your Theme
- Blank WordPress POT
- Improved i18n WordPress tools
- How to update translations quickly
- Workflow between GitHub/Transifex
- Gist: Complete Localization Grunt task
- WordPress.tv tags: i18n, internationalization and translation

---

□□□ #2
□ □□□ □□□ 2026-10-31 18:07:37 UTC
□ □□□ □□□ 2026-10-31 18:07:38 UTC