

Media Library Assistant

Internationalization and Localization Guide 2015/03/03

1 INTRODUCTION

Internationalization is the process of developing your plugin so it can easily be translated into other languages. Localization describes the subsequent process of translating an internationalized plugin. Internationalization is often abbreviated as i18n (because there are 18 letters between the "i" and the "n") and localization is abbreviated as l10n (because there are 10 letters between the "l" and the "n").

Media Library Assistant (MLA) has been significantly internationalized. As of this writing:

- All of the text strings in the PHP source code have been wrapped in a call to one of the Gettext library functions.
- Text strings have been removed from most of the HTML template files (.tpl files), moved to the PHP source code and wrapped in a Gettext call.
- The Poedit “cross-platform gettext catalogs (.po files) editor” (Free version 1.7.4) has been used to extract translation information from the source code and build the Portable Object Template (.pot) file.
- A directory (/languages) has been added to the plugin file structure to hold the template file, the Portable Object (.po) file(s) and the Machine Object (.mo) file(s).
- Code has been added to load the Machine Object (.mo) file(s) containing the run-time translations.

At this time, the HTML templates in the plugin’s /tpl directory for on-screen help (“help-for-tpl”) and for plugin documentation (“documentation-settings-tab.tpl”) have not been internationalized. Instead, a scheme has been implemented to load translated versions of these files from custom and language-specific directories.

2 CREATING A NEW TRANSLATION

There are several translation applications that work with the GNU gettext files. The MLA localization work has been done with the popular, free Poedit application. You can begin the translation process by downloading and installing Poedit from their web site, <http://poedit.net/>.

The examples and instructions in this Guide are based on Poedit Free version 1.7.4. Older versions of Poedit may not present the same dialog boxes and other screens shown in this Guide.

2.1 PREPARING FOR TRANSLATION

Before you begin your work, take a moment to set up a directory for the files you will be using.

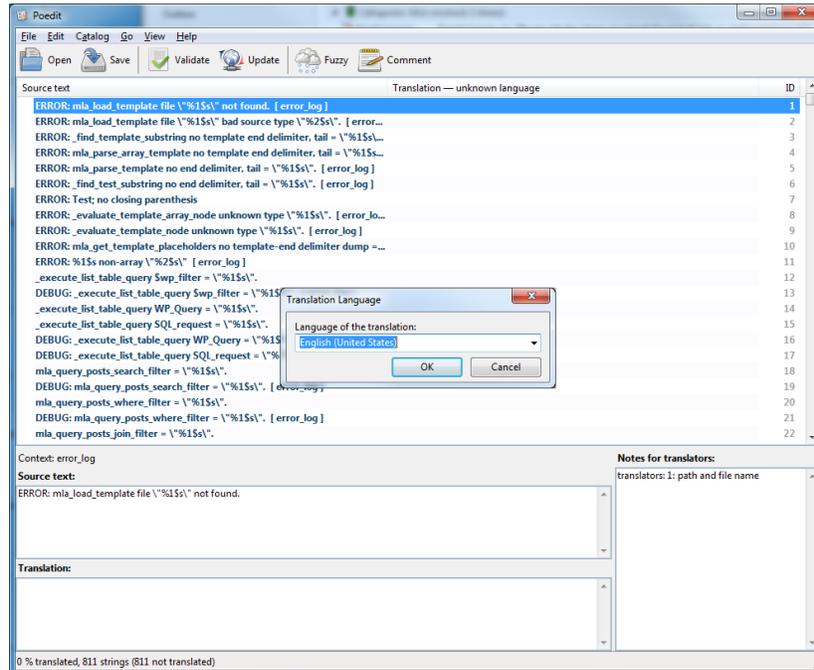
Media Library Assistant looks for its “standard” language files in the /media-library-assistant/languages/ directory. You can also find the standard language POT template in this directory.

Important: When a new MLA version is installed it will replace the contents of this directory. If you store your own translated files here they will be lost.

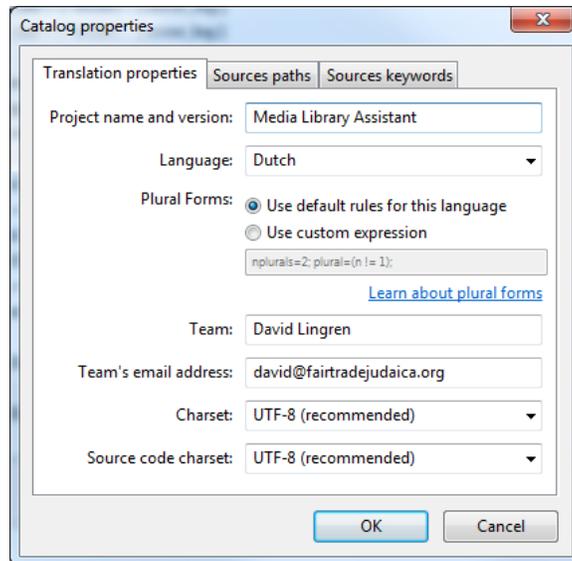
If your site is hosted remotely you should create a directory on the system where you install/installed Poedit. You can copy the MLA POT file (and the Guide) to this directory for easy access. If your site is hosted on the same system you can access the POT file from its original location, but you should still create a separate directory for your own .po file.

2.2 BEGINNING THE TRANSLATION

1. Assuming you use Poedit, start the process by opening the program. Then click the "Create new translation" link in the center of the welcome screen or pull down the File menu and select "New from POT/PO file..." Navigate to the /media-library-assistant/languages/ directory (or wherever you have a local copy of the POT file), select the "media-library-assistant-en_US.pot" file and click "Open". You will then see the "Translation Language" dialog box. In the "Translation Language" dialog box, select the appropriate "Language of the translation" from the dropdown list and click OK.



2. If the Translation Language dialog box does not appear, pull down the "Catalog" menu and select "Properties..." Change the "Language:" dropdown to your target language, then click "OK."

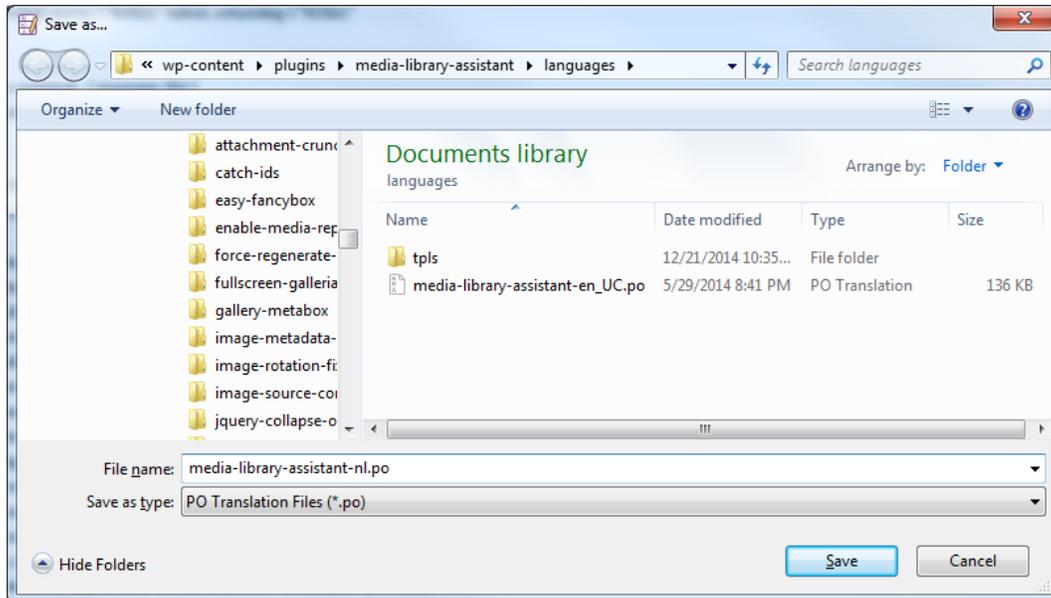


3. Before you get started on your translation, save the new file with a name that conforms to the WordPress conventions. There are two important conventions you must follow:
 - a. You must add the plugin slug, "**media-library-assistant-**", before the language/locale name. This allows WordPress to manage translation files for all the plugins on your site.
 - b. You must use the full locale code, not just the language code. For example, Poedit will suggest file name "nl.po", but you must add the country/specific translation code to get the full locale, "nl_NL".

The full Portable Object (PO) file name for a Dutch translation becomes:
media-library-assistant-nl_NL.po

4. Pull down the "File" menu and click "Save as..." When the "Save as..." dialog box opens, add "**media-library-assistant-**" before the language/locale name, make sure you have the complete locale name, then click "Save". A new .po file will be created in the /languages/ directory and given the appropriate locale name for your selected language. For example, the "English (United States)" language file will be called "**media-library-assistant-en_US.po**".
5. If you skip step 3, when you do save your work you must rename the file to conform to the WordPress file naming convention, adding the plugin name and the complete locale, e.g., "media-library-assistant-en_US.po".
6. Pull down the Catalog menu and select "Properties...". Fill in your "Team" information in the Translation Properties tab of the Catalog Properties dialog box. Depending on your language, you may need to adjust the Plural forms information as well. You can find more information about Plural Forms at the [Localization Guide](#) web site. You should not need to change the Sources paths or Sources keywords information.

7. Do as much translation work as you need.
8. To save your work, click the “Save” toolbar icon or pull down the File menu and click “Save”. The first time you do this you will be asked for the name and location of the file:



You should change the name of the translated file to conform to the WordPress conventions, i.e., add "media-library-assistant-" before the locale portion of the file name, as shown in the example above.

9. You can return to the .po file for additional work by pulling down the File menu and clicking “Open...” or clicking on the file name in the “recent files” area at the bottom of the menu.
10. Upload the .mo file for your new translation to your site and test your work. More information on upload location choices is given in a later section, “Activating your Translation”.
11. Consider translating one or more of the template (.tpl) files for on-screen help and documentation. See the “Translating Template Files” section for more information.

If you have done an extensive or complete translation for a language not yet a part of the MLA distribution, please consider sharing your work!

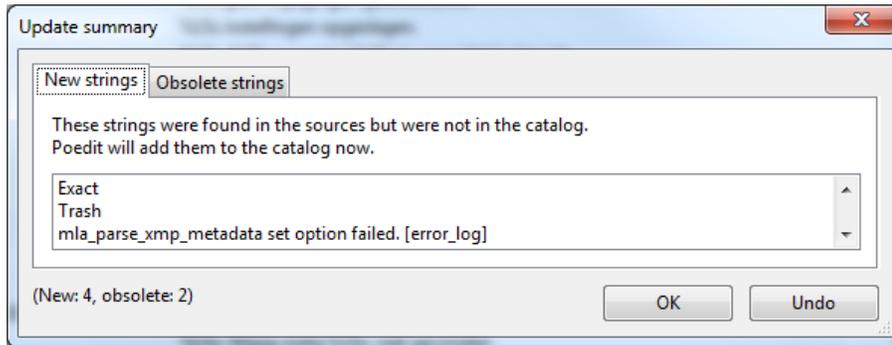
3 UPDATING A TRANSLATION

When a new version of MLA is released it will often contain changes to the .pot file. You can easily update your own .po file to reflect the changes.

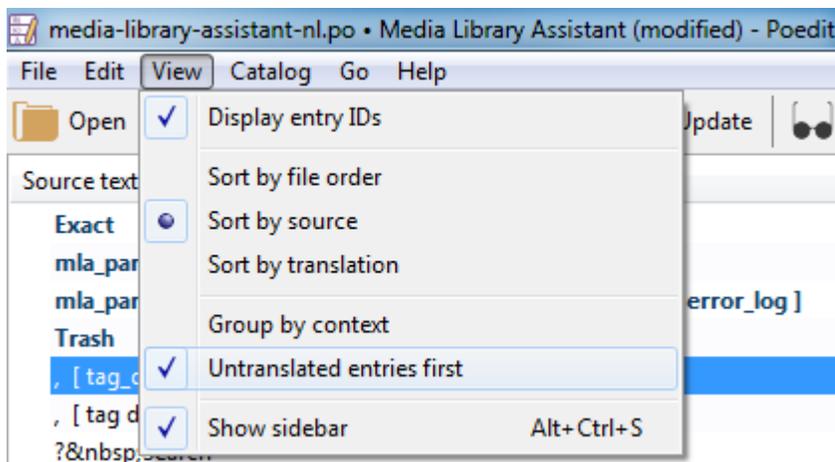
1. Assuming you use Poedit, start the process by opening the program, by pulling down the File menu and clicking “Open...” or clicking on the file name in the “recent files” area at the bottom

of the menu.

2. Pull down the Catalog menu and select "Update from POT file...". Navigate to the /media-library-assistant/languages/ directory, select the "media-library-assistant-en_US.pot" file and click "Open". The two files will be compared and the .po file changes will be displayed in the Update Summary dialog box:



3. Click "OK" to save the changes to your .po file, or "Undo" to leave your file unaltered.
4. If you accept the changes, you can easily find any new strings by pulling down the "View" menu and clicking "Untranslated entries first" as shown below:



5. Once you have completed whatever translation work they require, save your work and update your site with the new .po and .mo files.
6. Review the template (.tpl) files and update any of the files you've translated in past versions.

4 ACTIVATING YOUR TRANSLATION

Media Library Assistant looks for its “standard” language files in the `/media-library-assistant/languages/` directory.

Important: When a new MLA version is installed it will replace the contents of this directory. If you store your own translated files here they will be lost.

You can override any or all of the standard translation values by preparing your own “custom” `.mo` file and putting it in a second directory that MLA will search and load before its own standard files are loaded. This way, any translations in your custom file will be used in preference to the standard translations.

WordPress defines a “languages directory” for its own use. By default, this is `“/wp-content/languages/”`, but you can override that by defining `WP_LANG_DIR` in your `wp-config.php` file.

To add a custom file for MLA, create a subdirectory in your languages directory named “media-library-assistant”, e.g., `“/wp-content/languages/media-library-assistant/”`. Put your custom file(s) in this subdirectory and MLA will find them. For example, a custom “Dutch” language file would be found here:

`/wp-content/languages/media-library-assistant/media-library-assistant-nl_NL.mo`

Adding “media-library-assistant” to the file name seems redundant, but makes the file names conform to the new WordPress 3.7 scheme for automatically distributing and updating language files. The approach to custom language files is more fully described in this blog post:

[How to load theme and plugin translations](#)

5 TRANSLATING TEMPLATE FILES

MLA uses template files for many of its pages, mostly in the Settings/Media Library Assistant tabs. The template files separate HTML markup from the MLA source code and make it easier to maintain the plugin.

Most of the template files are internationalized; the source text values have been replaced by substitution parameters and values moved into the source code where they can be added to the `.pot` file. The on-screen help and documentation templates have not been internationalized. They are almost pure text and are too large for conventional translation.

If you want to translate a template file, you can copy the file, update it and put the copy in a separate place that MLA will look in before it loads the standard template file. MLA uses the locale name to separate template files for different translations while retaining the file name. There are two locale-specific directories in addition to the standard directory. In the examples below, “`{locale}`” is replaced by the locale name, such as “`en_US`” for “English (United States)”. MLA searches for template files in these three directories (in the order shown):

1. The “custom” directory; use this directory for files unique to your site. Files here will not be disturbed when a new MLA version is installed. Custom template files are stored within the

“media-library-assistant” subdirectory in the WordPress Languages directory, e.g., “/wp-content/languages/media-library-assistant/tpls/{locale}/”

2. The “language-specific” directory; use this directory for files that are part of the official MLA plugin versions. Files here will be replaced when a new MLA version is installed. Language-specific template files are stored within the “media-library-assistant” plugin “languages” subdirectory, e.g., “/wp-content/plugins/media-library-assistant/languages/tpls/{locale}/”
3. The standard directory; this is the place for the untranslated files that are part of the official MLA plugin versions. Files here will be replaced when a new MLA version is installed. Since the files are untranslated, no locale name is needed; the directory is “/wp-content/plugins/media-library-assistant/tpls/”

6 SHARING YOUR TRANSLATION

If you would like to contribute your translation to the MLA user community you can send it to me for inclusion in future MLA releases.

1. Contact me through our web site, [Fair Trade Judaica/Contact Us](http://fairtradejudaica.org/our-story/contact-us/) (<http://fairtradejudaica.org/our-story/contact-us/>). I will reply with an e-mail address where you can send your .mo and/or .po files.
2. Send the .mo file, the template files and the .po file. I will give it a quick test and if no problems emerge I will add it to the plugin distribution package.
3. Tell me how and with what links you would like to be acknowledged in the MLA documentation in appreciation for your efforts.

Thank you!

7 TRANSLATION TIPS

Context information and translator notes have been added to the .pot file entries to help you make effective translations.

7.1 NOTES FOR TRANSLATORS

These appear in the lower-right portion of the main Poedit window. They explain the content of any variable parts of the source text, i.e., arguments of the form “%1\$s” that will be filled in at run-time with specific values. The “Notes for translators:” that accompany these entries give you the content of the argument. You can re-arrange the order of these variable parts as needed; the first part of each argument, e.g., “%1”, will match the argument to the right substitution text.

7.2 CONTEXT INFORMATION

Sometimes one term is used in several contexts and although it is one and the same word in English it has to be translated differently in other languages. Context can also indicate words or phrases used in a special way, such as table column headers.

In Poedit you can see context indicators in square brackets next to the source text list entries and above the “Source text:” label in the lower-left part of the main window. The MLA .pot file has context labels for these special text entries:

- `error_log` – Error and debug messages written to the web server error file. Consider leaving these untranslated.
- `message_list` - Debug messages displayed only during plugin development. Consider leaving these untranslated.
- `tag_delimiter` – The delimiter between “flat taxonomy” items such as Tags and Att. Tags.
- `list_table_column` – Column headers for tables such as the Media/Assistant submenu table.
- `post_mime_types`
`post_mime_types_singular`
`post_mime_types_plural` – The labels used for the “views” list above the Media/Assistant submenu table, e.g., “Image/Images”.
- `post_mime_types_description` – The text describing the Post MIME Types in the Settings/Media Library Assistant Views tab.
- `upload_list_table_view_singular`,
`upload_list_table_view_plural` – The text describing the “views” list above the Settings/Media Library Assistant Uploads tab table.
- `taxonomy_name_plural`
`taxonomy_name_singular` – The plural and singular forms of the “Att. Tags” and “Att. Categories” taxonomies.

There is one additional “informal context” label to look for. Several of the Source text values begin with the string “mla_debug”. These are messages displayed when the `[mla_gallery]/[mla_tag_cloud]` “mla_debug” parameter is set to true. Consider leaving these untranslated.

8 INTERNATIONALIZATION AND LOCALIZATION RESOURCES

Here are links to the resources I found useful in planning and executing my internationalization work:

- [How to Localize a WordPress Plugin and Make it Translation Ready](http://premium.wpmudev.org/blog/localize-a-wordpress-plugin-and-make-it-translation-ready/)
<http://premium.wpmudev.org/blog/localize-a-wordpress-plugin-and-make-it-translation-ready/>
- [How to Translate a WordPress Plugin](http://premium.wpmudev.org/blog/how-to-translate-a-wordpress-plugin/)
<http://premium.wpmudev.org/blog/how-to-translate-a-wordpress-plugin/>
- [Localization Guide](http://docs.translatehouse.org/projects/localization-guide/en/latest/index.html)
<http://docs.translatehouse.org/projects/localization-guide/en/latest/index.html>

- [GNU 'gettext' utilities](http://www.gnu.org/software/gettext/manual/html_node/)
http://www.gnu.org/software/gettext/manual/html_node/
- [i18n for WordPress Developers](http://codex.wordpress.org/i18n_for_WordPress_Developers)
http://codex.wordpress.org/i18n_for_WordPress_Developers
- [Internationalization \(WordPress Codex\)](http://make.wordpress.org/docs/plugin-developer-handbook/plugin-components/internationalization/)
<http://make.wordpress.org/docs/plugin-developer-handbook/plugin-components/internationalization/>
- [How to load theme and plugin translations](http://ulrich.pogson.ch/load-theme-plugin-translations)
<http://ulrich.pogson.ch/load-theme-plugin-translations>
- [Video: i18n: Preparing Your WordPress Theme for the World – RTL handling at 31 minutes in](http://www.youtube.com/watch?v=fJfggrzjEis)
<http://www.youtube.com/watch?v=fJfggrzjEis>
- [Right-to-Left Language Support \(WordPress Codex\)](http://codex.wordpress.org/Right-to-Left_Language_Support)
http://codex.wordpress.org/Right-to-Left_Language_Support
- [Understanding _n_noop\(\)](http://kovshenin.com/2013/_n_noop/)
http://kovshenin.com/2013/_n_noop/
- [Poedit](http://www.poedit.net/)
<http://www.poedit.net/>
- [Create a .pot or .po File using Poedit](http://betterwp.net/wordpress-tips/create-pot-file-using-poedit/)
<http://betterwp.net/wordpress-tips/create-pot-file-using-poedit/>
- [WordPress – Poedit: Translation Secrets](http://www.cssigniter.com/ignite/wordpress-poedit-translation-secrets/)
<http://www.cssigniter.com/ignite/wordpress-poedit-translation-secrets/>
- [5.1.6 Language specific options – defines Poedit Sources keywords format](http://www.gnu.org/software/gettext/manual/html_node/xgettext-Invocation.html#xgettext-Invocation)
http://www.gnu.org/software/gettext/manual/html_node/xgettext-Invocation.html#xgettext-Invocation
- [Codestyling Localization \(WordPress Plugin\)](http://wordpress.org/plugins/codestyling-localization/)
<http://wordpress.org/plugins/codestyling-localization/>
- [WP Native Dashboard \(WordPress Plugin\)](http://wordpress.org/plugins/wp-native-dashboard/)
<http://wordpress.org/plugins/wp-native-dashboard/>