# LiteMap Language File Translation Help

Below are some rules and instructions about the language files.

**File Format:**

All language files are in UTF8 encoding, so whatever text editor is used it must be able to save the file in UTF8 encoding after editing. Don’t use Microsoft Word please.  On Windows machines use Notepad NOT Wordpad. Use ‘Save As’, and then you will see an ‘Encoding’ option which you can change to UTF8 before you save the file. On the Mac I think TextWrangler may have the same sort of choice.

**Structure:**

The structure of the files is that each entry is a php variable with a name then an ‘=’ then the text to be translated in speech marks. A simple example is:

$LNG->ERROR\_GROUP\_NOT\_FOUND\_MESSAGE = "The required Group could not be found";

The php variable name is ‘$LNG->ERROR\_GROUP\_NOT\_FOUND\_MESSAGE’ then and ‘=’ then the text to be translated, "The required Group could not be found", then a semi-colon at the end. So each entry in the language files is really a piece of php code.

**HTML Code:**

In the files you will see there are html elements embedded in amongst the language and there are php code elements there as well.

I have tried to keep these to a minimum. They are worse when we get to the full pages like the ‘About’ page.

For example:

$LNG->HELP\_NETWORKMAP\_BODY .= '<li style="padding-bottom:5px;"><img style="vertical-align:middle;padding-right:5px;width:16px;height:16px;" border="0" src="'.$HUB\_FLM->getImagePath('magminus.png').'">Click to zoom the map out. You can also scroll the mouse wheel backwards.</li>';

You need to just translate the actual text (I have marked in red above) and leave the HTML around it alone.

**Language core references:**

The major system element names, like the names of the item types, ‘Issue’, Supporting Argument’ etc.. have been entered separately in the languagecore.php file and then referenced with php code in the other language files. This is to enable very fast changing of core terms. For example, if a client wishes to call ‘Ideas’, ‘Solutions’ instead on their site, it would only mean changing the key term in the languagecore file (and not every sentence where the word ‘Ideas’ was written) and it will ripple through all other language files. The down side is that is does make the language files a little harder to work with.

For example this entry:

$LNG->LIST\_NAV\_NO\_EVIDENCE = 'There are no '.$LNG->EVIDENCE\_NAME.' items to display';

Here ‘$LNG-> EVIDENCE \_NAME’ refers to an entry in the languagecore.php file for the word ‘Evidence’. So here you would just translate the words around it (I have shown in red).

If the sentence needs reordering to make it work in your language then the referenced part needs to move around too for example:

$LNG->LIST\_NAV\_NO\_EVIDENCE = 'There are no items to display '.$LNG->EVIDENCE\_NAME;

or

$LNG->LIST\_NAV\_NO\_EVIDENCE = $LNG->EVIDENCE\_NAME .' there are no items to display';

Notice that the variable name, here ‘$LNG->EVIDENCE\_NAME’ is joined to the text part with a full stop ‘.’ Don’t forget to leave spaces in the text parts before/after the variable name so it display as a sentence properly.

The text being translated must always be in double or single speech marks. I try and stick to single but sometimes forget. Whatever speech marks are used if that speech mark also appears inside the text being translated it needs escaping with a forward slash before it, e.g.:

$LNG->HELP\_BUILDER\_USING\_OPERA = 'If you are using <b>Opera</b>, right-click on the link above, select\'Bookmark Link...\'. You can then choose to \'Show on bookmarks bar\'.';