

[Pm Wiki](#) supports internationalization (internationalisation) of web pages, allowing accented characters to appear in page names and almost complete customization of Pm Wiki's prompts.

Most customization is provided via the [XL Page?](#)() function in Pm Wiki, which loads a set of translation variables from a wiki page (typically named [XL Page?](#), but it can be named anything you wish).

The rest of this page is devoted to the installation, configuration and usage of other language(s) support. If you are looking for tools and help to *localize* Pm Wiki in your language, or how you can improve the existing translations, start on [pmwiki.org](#) with the page [Localization](#) - The Translation Portal.

Loading Translation Pages

Pages for many [other languages](#) have already been created and maintained at the [pmichaud.com](#) site. You can download an archive of these translations from <http://www.pmwiki.org/pub/pmwiki/i18n/>. Simply download the appropriate language archive(s), and unpack the archive(s) into the directory containing your *pmwiki.php* installation. Each archive contains a number of page files that are placed in your *wikilib.d/* directory, and some special scripts for translations that use a character set other than iso-8859-1 (Pm Wiki's default). You can also use [UTF-8](#) charset.

Once the translation pages are installed, you enable a language by adding a call to `XLPage()` in your *config.php* file. For example, to select French language prompts, one would specify

```
include_once("scripts/xlpage-utf-8.php"); # optional
XLPage('fr', 'PmWikiFr.XLPage');
```

which says to load the translations for French ('fr') from the page [PmWikiFr.XLPage](#). The `include_once` line is recommended if you start a new wiki, and it should be placed *before* the [XL Page?](#) line (for languages with alphabets other than the Latin, the `include_once` line is required). These lines should be placed near the beginning of *config.php*.

It's possible to load multiple pages; so if you want to create your own local translations without changing the ones you got from an i18n archive, just create another page (see below) and load it first. Be sure that you load first the page with your local changes:

```
XLPage('fr', 'PmWikiFr.XLPageLocal'); # my local translations
XLPage('fr', 'PmWikiFr.XLPage'); # from i18n.tgz
```

If your intention is to offer multiple languages on your site, and use [Wiki Groups](#) as language selectors, you may want to place this code in local customizations files (see [Group Customizations](#)). For example, if your site is published in French and English, and the French pages are in a group called Fr, you could create a file named *Fr.php* in the *local/* directory which contains:

```
<?php if (!defined('PmWiki')) exit(); ##change to French language
XLPage('fr', 'PmWikiFr.XLPage');
```

You may wish to create a page called [Pmwiki Fr?.php](#) with the same content to access the French documentation in the [Pmwiki Fr?](#) group. *En.php* is not necessary in this case since English is the default language.

An alternative to the above would be to add to *config.php* the following, which tests if there is an [XL Page?](#) in a group, and if it finds one it gets loaded:

```
$xlpag = FmtPageName('$Group.XLPag', $pagename); if (PageExists($xlpag))  
XLPag($xlpag, $xlpag);
```

With this method you would need to copy any relevant [XL Page?](#) into any group which needs the different language support.

See also [Cookbook:MultiLanguage](#)

Creating New Translations

If language pages don't exist for your desired language, it's easy to create one! An [XL Page?](#) translation file simply contains lines of the form

```
'phrase' => 'translated phrase',
```

where "phrase" is an internationalized phrase (denoted by `$[phrase]`) in Pm Wiki's `$...Fmt` variables, and "translated phrase" is what should be printed in your particular language. For example, the line (in `PmWikiFr.XLPag`)

```
'Search' => 'Rechercher',
```

converts `"$[Search]"` to "Rechercher" on output. The file [Localization:XLPagTemplate](#) is a good starting point for creating a new [XL Page?](#) and has most of Pm Wiki's key phrases already listed in it.

If you create new versions of Pm Wiki pages in other languages, please consider adding them to the [main PmWiki site](#) so that they can be made available to others in the i18n archives! (Be sure to check out [The Localization Portal?](#) for further information on effectively internationalizing Pm Wiki.)

The term "i18n" is commonly used as an abbreviation for the English word "internationalization". The abbreviation is derived from the fact that there are 18 letters between the "i" and the final "n" and few people want to type them all out.

Enabling "Special" Characters in [Wiki Links?](#)

To enable "special" characters like for example German umlauts in [Wiki Links?](#), it is necessary to configure the server locale to ensure that Pm Wiki uses the proper character set definition.

If this is not possible due to limited access to the server configuration, Pm Wiki can be configured to use a specific locale by using the [XL Page?](#) options (see [XLPagTemplate](#)).

For German umlauts, you'd need for example:

- 'Locale' => 'deu', <- for Windows servers, see [MSDN List of locale identifiers](#)
- 'Locale' => 'de_DE', <- for Linux servers; for the UTF-8 encoding, on some installations you may need to set 'de_DE.utf8' or 'de_DE.UTF-8'.

Note that the locale identifier depends on the operation system and perhaps on the specific installation.

Notes

If my wiki is internationalized by *config.php*, how do I revert a specific group to English?

Use `$XLLangs = array('en');` in the group's [group customization](#) file.

If my wiki is in English and I want just one page, or group, in Spanish do I say `XLPage('es', 'PmWikiEs.XLPage');` in the group or page configuration file?

Yes, that is usually the best method. If you were doing this with many scattered pages, or with several languages, you might find it easier to maintain if you load the translations all in *config.php* like this:

```
XL\_Page?\('es', 'Pm Wiki Es.XL\_Page?'\); XL\_Page?\('fr', 'Pm Wiki Fr.XL\_Page'\); XL\_Page?\('ru', 'Pm Wiki Ru.XL\_Page?'\); $XLLangs = array('en');
```

Then in each group or page configuration file, you'd just use `$XLLangs = array('es');` to set the language to use (in this case, Spanish). Note that though this method is easier to maintain, its somewhat slower because it loads all the dictionaries for each page view, even if they won't be used.

What does the first parameter of this function stand for? How can it be used?

The [XL_Page?](#) mechanism allows multiple sets of translations to be loaded, and the first parameter is used to distinguish them.

For example, suppose I want to have translations for both normal French and "Canadian" French. Rather than maintain two entirely separate sets of pages, I could do:

```
XL\_Page?\('fr-ca', 'Pm Wiki Fr Ca.XL\_Page?'\); XL\_Page?\('fr', 'Pm Wiki Fr.XL\_Page'\);
```

[Pm Wiki Fr.XL_Page](#) would contain all of the standard French translations, while [Pm Wiki Fr CA.XL_Page?](#) would only need to contain "Canada-specific" translations -- i.e., those that are different from the ones in the French page.

The first parameter distinguishes the two sets of translations. In addition, a *config.php* script can use the `$XLLangs` variable to adjust the order of translation, so if there was a group or page where I only wanted the standard French translation, I can set

```
$XLLangs = array('fr', 'en');
```

and Pm Wiki will use only the 'fr' and 'en' translations (in that order), no matter how many translations have been loaded with [XL Page?\(\)](#).

How can I add a translation for an individual string in a PHP file?

Use the XLSDV() function to provide a translation for a specific (English) string. For instance, with this in config.php

```
XLSDV('nl', array('my English expression'=>'mijn Nederlandse uitdrukking'));
```

any instance of the variable expression `$[my English expression]` in wiki mark-up will be displayed as *my English expression* in default (English) context, but as *mijn Nederlandse uitdrukking* in Dutch (nl) context, i.e. when `XLPage('nl', ...)` has been called for that page in config.php or a cookbook recipe.

If you need to get a translation in a PHP file, use the `XL()` function:

```
$local_string = XL("my English expression");
```

But beware: [XL Page?\(\)](#) uses XLSDV() internally for its translation pairs, too, and only the first definition is accepted! Thus, if the Dutch [XL Page?](#) already contains a translation and you want to override that, you need to use your XLSDV('nl',...) *before* calling the correspondent [XL Page?\(\)](#)('nl',...). Otherwise, by using XLSDV() *after* [XL Page?\(\)](#) - e.g. within a recipe that is included later in config.php - your translation will only work as long nobody defines 'my English expression' in that [XL Page?](#).