

This is not documentation about **using** Tiki but documentation for tiki site admins about customizing the interface. Perhaps the part about **using custom translations** should be on doc and the part about **contributing translations** should be on a i18n workspace on tiki.org.

## Translating the Tiki interface

This page is a kind of [tutorial](#), that needs [needs review](#). Moreover, some more [screenshots](#) would be nice... See related page: [i18n Admin](#).

See also \* [Interactive Translation of Tiki Interface](#)

### 1.1. Status of the translations

See: <http://i18n.tiki.org/Status>

### 1.2. Translate through i18n.tiki.org

One of the easiest ways to contribute to the translations of the Tiki interface is registering at i18n.tiki.org and use the interactive translation feature to get most of the strings available translated into your known languages.

That ususally means committing changes to the next development version of Tiki (since these changes will be merged into the trunk branch of development).

See: <http://i18n.tiki.org>

### 1.3. Translate editing the language.\* files on disk and use svn

For example, to improve the translation into Catalan, it is necessary to modify the content of the file:

```
./lang/ca/language.php
```

And since [tiki6.1](#), you can edit the strings at:

```
./lang/ca/language.js
```

These .php (and .js) file contains a bunch of lines with pieces of sentences or words ("strings of text") like:

#### Code

```
(...) "Internationalization" => "Internacionalització (I18n)", "I18n setup" => "Configuració de I18n ",  
"Best Language" => "Millor idioma", "Restrict available languages" => "Restringeix idiomes  
disponibles", "Available languages:" => "Idiomes disponibles:", // "Intertiki exchange feature" =>  
"Intertiki exchange feature", // "Intertiki client" => "Intertiki client", (...)
```

Comments start with "// ", but some lines that have not been translated yet also starts with those characters:

#### Code

```
// "Intertiki exchange feature" => "Intertiki exchange feature",
```

When a line have been correctly translated, you can delete the two initial slashes and the blank which follows to let Tiki use the translation:

#### Code

```
"Intertiki exchange feature" => "Funcionalitat d'intercanvi Intertiki",
```

Once you have improved or fixed the translations in the **language.php** file, you can overwrite the original file on the server (/lang/ca/language.php). It is highly recommended to make a backup copy of the original file; you can rename it to language.php.old. Done the above, to load the translations to the server, login as administrator and then execute the following request on the browser's address bar:

#### Code

```
http://yourdomain/get_strings.php?lang=ca
```

Finally, to apply the new translation in the current instance of Tiki, you must clear the system cache. You can then continue translating and improving translations by repeating the process as frequently as desired.

If you want to share the translation improvements for a language, you can share the language.php file in the Tiki Git repository (the central repository of code; *keep reading, further down* ).

This general process has some variants, depending on how to edit the file that contains the text strings. Shortly, you can modify:

1. Each time directly against the central repository svn. [How to commit](#)
2. Previously through Wiki pages
3. Previously through Tiki itself (database)

Be careful that the language.php is a php parseable file. So you must not change some lines at the top

#### Code

```
<?php // -*- coding:utf-8 -*- $lang=Array(
```

And some lines at the bottom

#### Code

```
"###end###"=>"###end###"); ?>
```

Be also careful to respect the syntax on each line

#### Code

```
"..." => "...",
```

Each string is surrounded with double quotes ". If you need to add a double quote in your string, precede it with a \ like this : \". Do not forget the comma at the end of each line. Separate the 2 strings by =>.

### 1.3.1. Character sets

The file must be saved in utf8 encoding.

If you're using Emacs (external link) then it's easy to change the encoding of a file. Simply open the file, and then type '+C-x RET f+-' which runs the commandset-buffer-file-coding-system. Now choose utf-8 from the list.

### 1.3.2. How to contribute your translation improvements to Tiki

In order to have your translation improvements contributed to Tiki code (so that they are included in new releases, etc), you have to commit then to the branch equivalent to your tiki installation. Or to trunk branch.

For more information about branches and how to commit, check:

<http://dev.tiki.org/Commit>

For more information on how to merge strings translated from trunk into previous branches, see:

<http://dev.tiki.org/Merge+language+improvements>

### 1.3.3. Reference sites

- OpenTrad Apertium: On-line free software translator, including Catalan-French, French-Catalan, Catalan-English, English-Catalan, among others.  
<http://xixona.dlsi.ua.es/prototype/>  
<http://www.opentrad.org>
- Dictionary English-Catalan (open source code):  
<http://www.ibiblio.org/dacco/cat/>

### 1.3.4. PHP/Smarty translations

#### 1.3.4.1. Others: Convert to .PO file

Also there is the option to convert the language.php to a **.po** file, in order to take advantage of other tools which support the translation of **PO** files using the GNU *Gettext*.

This system is not explained here. You can find more information at:

- <http://tiki.org/PO+Convertor+for+TikiWiki>

#### 1.3.4.2. Translation through Tiki interface (database)

You need to enable the option: "Admin > [i18n](#) (Internationalization) > **Use database for translation**", and afterwards, clean caches ("Admin > System Administration" > ./templates\_c/ **Empty**).

Then, the link is enabled in the main application menu: *Admin* > **Edit languages**, that will bring to an url like:

---

[http://yourdomain/tiki-edit\\_languages.php](http://yourdomain/tiki-edit_languages.php)

Then it is necessary to create the translation into the database, for the Catalan language in this example, and tell Tiki to import the text strings already translated at the file **language.php**. This is performed in some short steps:

1. Tell Tiki to create a record for the new language (Catalan) into the database

1. Create Language  
Shortname: **ca** (like en)  
Longname: **Catalan** (like English)
2. Click in the *Create* button
2. Select the language you want to edit (Catalan) in the drop down menu.
  1. Select the language to edit: **ca**
  2. Click on the button *Set*
3. Then go to the link above called: *Im- Export Languages*, that will lead you to to an url like:
 

◦ http://yourdomain/tiki-imexport\_languages.php?
4. Import the text strings of the language.php in Catalan
  1. Go to "**Import > Select the language to**", and choose **Catalan (Catalan,ca)** in the drop down menu. ## Click on the button *Import*.
5. Clean cache ("Admin > System Admin" > empty the three sections).  
You will see already the interface in your new language (Catalan, in this example).

To edit the text strings, then you need to:

Go to "Admin > Edit languages". In "Edit or export/import Languages", select the language you want to edit (do as explained above), and afterwards go to "**edit Translations**".

Then you will be able to edit the text strings you want, search any of them, etc. The process is quite intuitive.

**EDITA I CREAM IDIOMES**

**Crear Idioma**

Nom curt:  (com en)

Nom llarg:  (com English)

Seleccionar el llenguatge que es vol editar: ca ▼

☐ Afegeix una traducció

☒ edita traduccions

☐ Traducció gravada

**edita traduccions:**

Original: <input type="text" value="for rows"/>	Traducció: <input type="text" value="for rows"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="Note 1: if you allow you"/>	Traducció: <input type="text" value="Nota 1: si permeteu als usuaris confi"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="- Php GD library required"/>	Traducció: <input type="text" value="- Llibreria Php GD requerida"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="at"/>	Traducció: <input type="text" value="un"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="is interactive but has no role assig"/>	Traducció: <input type="text" value="és interactiu però no té funció assignada"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="is non-interactive and non-autoreg"/>	Traducció: <input type="text" value="no és interactiu ni tampoc autodirigit ni té fun"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="is not mapped"/>	Traducció: <input type="text" value="no està mapejat"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="is standalone but has transitions"/>	Traducció: <input type="text" value="està en solitari però té transicions"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="item found"/>	Traducció: <input type="text" value="registre trobat"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>
Original: <input type="text" value="items found"/>	Traducció: <input type="text" value="registres trobats"/>	<input type="button" value="tradueix"/>	<input type="button" value="elimina"/>

Once finished, export the translation to the file `language.php` (through the link "**Import/Export Languages**"), and select the language you want to export. It is necessary that the file and unix folder are writable for the apache user, if in a GNU/Linux or similar environment.

At the end, it is necessary to take this generated file `language.php` (in `./lang/ca/language.php`) and upload it (commit changes) the central repository of code of Tiki at [sourceforge.net](http://sourceforge.net), through SVN, in case you want that other people can benefit from your improvements (desirable! 😊).

### [An example of adding a language](#)

#### 1.3.5. Custom translations

It is possible to customize your strings translation without changing the `lang/your_language/language.php` file. It is very convenient when you upgrade or sync with the latest. You can keep your translations. Your strings must be put in a file `lang/your_language/custom.php`. This file must be readable by the server. The file must have UTF-8 encoding. The format is very close to the format of `language.php`

---

```
<?php $lang_custom = array( "Hello"=>"My Hello", "This is the place to customize your translation"=>":-)", ); $lang = array_merge($lang, $lang_custom); ####trebly: this changed will not be probably necessary for 7.x after 01/18/2011
```

In Tiki 9.x there is this on the last line:

---

```
$lang = $lang_custom + $lang;
```

A file example is located at `lang/fr/custom.php_example`.

You can customize any translation with this file by putting exactly the same string on the left as in `language.php`. The strings that are not in `custom.php` are still translated as specified in `language.php`. You can even change the English strings by using this file. For instance, if you do not like 'Contact Us' but prefer 'Contact Form' you can have a `lang/en/custom.php` like this

---

```
<?php $lang_custom = array( "Contact Us"=>"Contact Form", ); $lang = $lang_custom + $lang;
```

Note : Be aware that any file like `"language_*.php"` will be also included in the translation.

When you are on a [multitiki](#), you can have a `lang/your_language/custom.php` and a `lang/your_language/your_tikidomain/custom.php`

##### 1.3.5.1. Custom JavaScript Translations

Some text can only be translated with JavaScript in the client browser. To do this add a file `lang/your_language/custom.js` using code in this format (as found in `lang/fr/custom.js_example`)

---

```
lang = $.extend(lang, { "Favorite" : "My favourite!", "Example in English" : "Translated text here" // remember IE does not support ending comma on last item });
```

👉 Be sure the file is saved with UTF-8 encoding. If the file is not saved with UTF-8 encoding, a fatal error may result and no Tiki pages will display. Terms/phrases in the custom file will be used in place of the same terms/phrases in the regular language file. (Not just anything can be added to these files, of course.

They must be strings specified for translation in the templates, etc.)

#### 1.3.5.2. Custom JavaScript Translations Without a File

If you cannot access the file system you can add similar code into the *Custom JavaScript* preference on the Look & Feel control panel.

---

```
if (jQueryTiki.language === "fr") { lang = $.extend(lang, { "Are you sure you want to unassign this  
module?" : "This is a good test js string to try!", "Example in English" : "Translated text here" //  
remember IE does not support ending comma on last item }); }
```

#### 1.4. More Information

- [Internationalization](#)
- [Internationalization Admin](#)
- [Interactive Translation](#) of Tiki Interface
- [Multilingual Wiki](#)
- <http://tiki.org/i18n>

Alias names for this page

---

[Translating Tiki interface](#) | [Translating interface](#) | [Interface Translations](#)