

Translation branching strategy

The policy is that translations are now no exception most of the time, changes are done in trunk and can be backported if desired (although backports still don't need to be fixes).

During the [semi-automatic merging period](#) (ex.: from pre 7.x to 7.1), there is no guarantee that changes on 7.x will be merged to trunk. Changes should be merged if there are no conflicts, but otherwise may be ignored. People should commit to trunk and backport to 7.x. If not, people who commit to 7.x are responsible to make sure their changes are properly merged to trunk in the next semi-automatic merge.

If someone is willing to handle merges for some languages, please speak up and the policy may be changed to one favoring more contributions.

How to merge language files

To merge language files from different branches you should use [doc/devtools/mergelang.php](#) script.

Say you want to update 7.x with some translations you did on trunk to the Brazilian Portuguese (pt-br) file.

The first think you have to do is make sure that language.php file in 7.x is updated. Running `get_strings.php` from within your branch folder will get new string that have been added to Tiki:

```
php get_strings.php lang=pt-br
```

And than `mergelang.php`:

```
php doc/devtools/mergelang.php pathToTikiTrunk pathToTiki7x lang=pt-br
```

Where `pathToTikiTrunk` is the path to trunk root directory and `pathToTiki7x` is the path to 7.x root directory. `lang` can be a single language code or a list of language codes separated by commas. This param is optional and if you omit all language files will be updated.

For each English string that this script find in both files, it will replace the translation in the 7.x version with the translation from trunk version (even if the string is untranslated in 7.x).

i18n.tiki.org

i18n.tiki.org will always use trunk (the next version of Tiki in development). Periodically (every one or two weeks) [Rodrigo](#) exports the translations done there to the SVN repository (on trunk). Before exporting the translation in i18n.tiki.org from the database, he updates the language.php files, so no translation should be lost in this process.

Discussion

Table of contents

-
- [How to merge language files](#)
 - [i18n.tiki.org](#)
 - [Discussion](#)
 - [Problems](#)
 - [Options](#)
 - [Translate to trunk, no automatic backport \(The safe way\)](#)
 - [Translate to trunk, with automatic backport to stable branches?](#)
 - [Translate to stable branch, with merge up](#)
 - [Unversioned translation files](#)
 - [Roadmap](#)
 - [Suggestions](#)
 - [Related](#)
-

Problems

Here is the problem:

<http://old.nabble.com/language.php%3A-trunk-versus-3.0-tt23746165.html>

2 conflicting things are desirable in the way translations are managed :

- Making the work of translators as easy and immediately useful to them as possible to obtain the most contributions.
- Minimizing the work of coders performing merges caused by new translations so merges are done frequently and translations not lost.

The policy used to be that translators could commit translations to a stable branch and they would be merged to trunk. Unfortunately this didn't work because nobody performed merged and translations were lost.

The problem is that the strings in language.php do not always stay in the same order so merging can be a nightmare (string by string). As the person doing merges generally knows only a fraction of languages, handling conflicts is a challenge. However, one easy solution to this problem is doing an alphabetical sorting of all lines in language.php, so that they have the same order regardless of versions, with some new lines in between here and there in the trunk version of language.php, and some missing. As far as Xavi knows, simple editors (such as gedit in GNU/Linux-Gnome, and probably many others in other OS's) already handle lines starting with "// text" as if they were starting with "text", and thus, keeping a line in the same place (once alphabetically ordered) regardless of the fact that it's been already translated or not.

Options

Translate to trunk, no automatic backport (The safe way)

Translators decide to backport to branch or not as they wish.

- If no backport, people have to wait next major version before using.

Issues:

- Less translations may be contributed as translators have to backport if they want to use the translations immediately.

Translate to trunk, with automatic backport to stable branches?

- People would not have to wait for a new major version to get new translations that weren't backported

Issues:

- Relies on volunteers to perform backports

Translate to stable branch, with merge up

Issues:

- Relies on volunteers to perform merges

Unversioned translation files

Instead of having one translation per branch, a single out-of-tree translation file would be used. This avoids merge issues.

Roadmap

We should ideally move to unversioned translation files.

One drawback they have is they contain some translations unused in certain versions. However it would be possible to strip unused translations if we can tell which translations a version uses. Current translations have a [unused translations](#) section which contains translations that could be used but that `get_strings` doesn't detect as being used. When all translated strings are properly marked in the code, stripping will become possible safely.

However, we have to keep in mind that mods used the same single and central `language.php` file for their translations. So that when someone reviews strings to be (potentially) striped out, some solution has to be provided for translations (such as Catalan) which have many strings for some mods translated (nowadays, and potentially in the future also). Potential solutions:

- some way to add extra `php` files under the `lang/xx/` folder , provided by mods, and used if tiki finds them (like with `custom.php`). One extra file per mod.
- whoever reviews potentially obsolete strings, installs all mods available for that version (some of them work, even if those mods are tagged as if they work only in earlier or unspecified versions of Tiki)
- Mods patch `language.php`s
- Have a file defining strings that will not be stripped

Suggestions

(Torsten)

Translation in trunk and no backport, but providing a **downloadlink** for certain languages that are very actively translated.

Update these few files out of trunk on a regular base.

Keep strings in Trunk that are not used anymore, but needed in the former branch, as long as the former branch is supported.

Siteadmins could then backport their individual languages by themselves by just downloading and copying the `language.php` of one or a few languages into `/lang/XYZ` .

For certain custom translations this would anyway to be considered by an admin.

(for ex. for one of my sites "shoutbox" is "roadbook", what is very specific and not for the main codebase - I do have several translations like this)

Related

- <http://translate.sourceforge.net/wiki/guide/svntips>