I18nDev

Any relevant info about **improving Tiki features** should be moved to http://dev.tiki.org/i18n

Status/RoadMap

- Where are we?
  - {tr smarty function and tra(php function translates text by looking up a key in an associative array or database table.
- Where do we want to be?
  - more robust translation - allowing default texts to be changed without requiring an update to all the language files - mdavey
    - *? A lot of changes in English strings require a review of the translations (except for the typo that are usually quickly fixed by the translators and solved with a sed) sylvie*
- Who is working on what? (Priorities/goals/majors issues/roles)

TikiTeam

See i18n Team

Trackers

Known problems :

- 2 letters language name must be upgraded to 2 letters+2letters specific because it seems we need more possibilities (ex: one could have both en_UK and en_US for British and American English, and zh_TW for Taiwan.)
  - =>In 1.9, It can be what you want but better to follow naming convention
- substr doesn't work on a UTF-8 string :
  - the solution is to overload the function substr() with mb_substr() & http://www.goldvision.com/us/en/manuals/phpen/ref.mbstring.html, but this needs to be able to configure php
  - do we impose php to be compiled with multi-byte string support and change all the substr ... with mb_substr....?
  - do we write a specific substr ? *(that would be the more clever - mose)*
    - => In 1.7.2 , a developper has to use mb_substr, if the multibyte support is not installed, a local mb_substr has been supplied. No other string routines have been rewriten. (Sylvie)
- date format : the date format is set by the admin and not with the language - and a multilingual site the date is not correctly displayed
  - do we put the date format in language.php?
    - => in 1.8 the formats have been added to the language.php file, the admin settings overwrite the language site format (sylvie)

- some translations need a context

    - table headings need to be short : a ttranlator needs to know it is a heading to choose a shorter translation or an abbreviation
    - a substring applied after translation is not always good: (ex: the month in the calendar) the number of characters depends of the language and the string
    - do we add a context at the end of a string that is eliminated is not usefull in the smarty prefilter. ex: "from[[date" .

::

Member	Vote
sylvie	yes

::

    - do we plan a major update like using gettext

Member	Vote

    - I think most of those problems can be fixed by using the sprintf() function in PHP as I've described in this message. I believe that's the common practice in other programs. --gimpster ''

- {tr} improvment

Even if it possible to translate some strings with variable (example: {tr}Hello {$username}{/tr} with in language.php file "Hello {\$username}"=>"Bonjour {\$username}", all configurations are not possible. For instance "{tr}Hello {$title} {$username}{/tr}". ($title needs a translation, not $username).

    - One idea to solve this is to add parameters to the tr function.
        - Example "{tr t={tr}$title{/tr}}Hello %t {$username}{/tr}" and a language file "Hello Mr {\$username}"
    - other idea?

- some language needs UTF-16.
    - What languages should that be? As far as I know, then a language cannot *need* a specific encoding such as UTF-8 or UTF-16. All you have to decide is how you want to represent your (Unicode) characters and UTF-8 is capable of representing all characters found in Unicode. See CharacterEncodingTrouble for more about the same. --gimpster
    - do we set a global parameter for the encoding?

- Calendar
    - Internationalize Calendar module
    - The first day of the week in French is Monday

RFEs

- WikiWords generation for non-anglophone characters
- Cut language.php's size for lowering memory usage, see botttom of page for more.

Competition and standards

List of other products with similar/interesting/related features.

Here I would like to see some "editorial" content. How do our features compare to others?

CVS Doc section

This is where new features being developed and only in CVS are documented. When the CVS becomes RC/official release, the info in the CVS docs is transferred to update the official docs (FeatureXDoc).

Discussion/participation

language.php's memory usage

mose said :the language.php is so heavy that it's one reason we need huge amount of memory for tiki. (*sylvie: 228KB .... is it really bigcompared to the 5MB - but optimization is everywhere*) Maybe that file cane be regenerated by admin with only the selected options translated, to a language_light.php. That would imply to declare somewhere the list of tpl for each feature, but imho it's not so a huge work when that first step is done.

Chealer9 adds :
I was not aware of this issue. Here's an idea this gave me. I'm highly unsure about it and would like your comments. I am Not proposing myself for this, it's just a brain game.
Well, I had more ideas since, I'll write it in steps so it's easier to follow.

1. Replacing english strings in tpls with numbers used to specify which line in language.php corresponds with translation. This slightly cuts tpls's size but reduces language.phps' size by half (for memory). This must be nearly as long as writing a new core though, and it means that strings changes or additions would take long.
2. Adding both a line number and the english equivalent makes tpls nearly as easy to read as before. But it's still long to change translations, so instead we can proceed just as we do now and add the line number periodically running a script similar to get-strings.
3. Actually, the english corrections can be made directly in template. It can be done either by periodically running a script to watch if all the strings inside {tr}s still match their associated number and if not updating the english language.php. It would probably be more efficient to rely on something watching CVS commits to run the script on the file immediately.

Here are associated **advantages** :

- I guess we could make spotting the context (discussed higher in trackers) pretty easy keeping the string number unique even for identical strings.
- Actually this can push mose's idea of language_light to its limit since Smarty could first find all line numbers it has to translate and then only read those. I think it also saves the admin to compile files ans us to associate each file with a feature. This would mean a nearly inexistant memory usage of translation
- We could currently save about 1.5 MB from the uncompressed Tiki size removing the english strings. I guess this would make fixing the english in tpls less annoying for translators too.
- I don't know much about the email notifications' linguistic problem, but I guess the change would make it much easier to solve.

Assuming someone codes this it would leave three **problems** :

- Development is a bit complicated. That's yet another thing you have to understand.
- Also those two spam issues
  - line numbers in tpls
  - for each CVS commit changing tpl strings, a bot would commit another version with line numbers.

I realize this is not really a Tiki only thing, maybe others already have something like that...Smarty? At least, if not then coding it could probably be contributed to Smarty. Also, I totally forgot about PHP's tra() while writing this and I just hope it can still work with that. This can really contain errors, feel free to correct me.

Update : I attached a discussion I had at #smarty with other devs. One talked about gettext, I wonder if we can use it optionally. It was supposed to be useful for all those advantages except context.

Update 2 : Sylvie came with a link to Smarty gettext today on IRC without much feedback, but it sounds good

Update3 : sylvie: In 1.9 I took away a couple of include(language.php). Now (I think) the include is done only when it is really necessary. If the tpl files are precompiled, the include(language.php) is only done by the tra calls.
=> to optimize the perf:

- transform tra into tr: can be done in some cases
- include a language_tra.php in the tra function that includes only the tra strings (around 700 strings on a total of 4100) - that can be interesting for memory also
- develop a more efficient access to the strings (that is what gettext is doing)

The creation of the language_tra.php can be done when you compile all the tpls.

Remove bloat from language.php

Observation

As we see on i18n status, many translations are incomplete. I have in mind two patches submitted to SourceForge, one proposing Ukrainian with 30% completion, the other proposing Korean with 10% completion. I hesitate to include those, since Tiki's default lang folder already eats about 4 MB, and those languages are not even complete.

Proposed improvement

Why not add something to the release script that would remove untouched strings from (incomplete) language.php-s? The package's size would be reduced, but also the memory consumption of those languages. This sounds quite easy to code, and it would make include partial translations much more reasonable to consider.

Chealer9

What about the question that Isam Bayazidi posed about RTL languages, ones that are read right-to-left, and the benefit of right justification? He wanted a bi-lingual site, i.e. Arabic and English. — UserPagejcwinnie 2004-May-17 -

> a partial solution in i18n status right to left langauge paragraph - sylvie

tr tokens proposal

The following mechanism (or something very similar) is used in most popular operating systems - it could be added to Tiki with minimal impact on the Tiki environment.

Syntax

{tr token=$token}default text{/tr}

Either token or default text **must** be supplied. Supplying both is acceptable.

New message string format:

<pre> <token> ::= <tokchar> { <tokchar> }* <tokchar> ::= <char> | <wildcard> <char> ::= any character > code 31 (' ') except ',', ')', ':', '?' or '/' <wildcard> ::= '?' (matches any character) <value> ::= { <anychar> | '{$variable}' | '{{' }* </pre>
Spaces may only appear in <value>

Alternative tokens

Alternative tokens are seperated by '/'.

Wildcards

The '?' character in a token in the file matches any character in the same position in the token supplied to be matched.

Case significance

Case is significant.

Unmatchable tokens

The actions to find a match for a token are:

1. search the appropriate language file using token as the key
2. search the appropriate language file using default text (if supplied) as the key
3. use the default text (if supplied) as the translated text
4. generate a error

Note: there is no attempt to check the default language file. It is expected that the default text will always be supplied

Namespaces

Initially the extended features of tr would only be used for text labels and button text. The token always begins with:

token	description
button.	for button text
label.	for label text

Examples: button.edit label.user_flip_modules

Supplying a default string

Whenever you use {tr you can supply a default string to be used if Smarty cannot match the token. The syntax is:

{tr token="label.user_flip_modules"}Users can Shade Modules:{/tr}

Errors

- Message token {$token} not found for language {$lang}

get_strings.php

get_strings.php would need to be extended to understand the new syntax and do the right thing.

- is there a better way to represnet the messages file as an associative array, while keeping the flexibility of many:1 tokens:strings relationships
- how do we make this work with database-driven translations?

- I don't like at all the notion of token. Thinking that with token you can change English translation without reviewing the other translations is wrong. If the English changes, the others have to change. I don't think about typo fixes, and sed is easy to do, a php can be also very easy to do (if the language file are writteable) sylvie

    - I understand your point, but I disagree. Firstly, if a button is renamed from 'remove' to 'delete' there is no need to update the language translations. Secondly you won't catch the changed syntax if a developer makes the button labelled 'remove' no longer delete, but instead archive an artifact (but doesn't update the button label). Aside would it be spotted if a developer changed a button text and updated all the language files at the same time (to use the same translation with the new string?). — *mdavey*

    - perhaps tokens should only be used in limited circumstances, to add context to short strings such as button labels? *mdavey*

- The major I see now is:

- to be able to specify the context af a translation.

The other I had in the chart feature "perm" that is not perms but the abbreviation of "permancency"
- to optimize the access to the tra strings sylvie

- So I propose

{tr context=xxxx}string{/tr}
the context will be added by each translator each time he has some problem. For an abbreviation it will be the complete string. For a table header, it will be the indication "header".
get_strings will have to create the lines
//"context#string" => "translatedstring"
The translator can uncomment the line and adds his translation if needed sylvie

For the optimisation I propose to compile the remaining strings in a (md5, translatedstring) to speed the process. sylvie
^

See also

- i18n Team
- doc:Internationalization
- MultilingualStep2 and all other pages in this category

IrcHook