Create a tool that extracts all text to be displayed

Status
● Open

Subject
Create a tool that extracts all text to be displayed

Version
18.x

Category
• Feature request

Feature
i18n (Multilingual, l10n, Babelfish)

Resolution status
New

Submitted by
hman

Lastmod by
hman

Rating
★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★
So everything will have to be re-checked., time and again.

Since Tiki Wiki does not have one file of text that resides in constants, you cannot simply generate a diff to see what has changed, and thus needs new (or altered) translations.

Therefore a tool would be necessary that extracts all text to be displayed. That tool could then also look up if there are pre-existing translations of those strings, and present them, too. As a side effect, such a tool could help developers to guard their code against strings NOT passed into tra() or tr (which makes translating them impossible).

In the first stage, such a tool could simply produce one file as output, in later development could be interactive. Such a tool would be a necessary prerequisite for version advancement of Tiki, because without it, all (!) translations are rendered into "status unknown" with any change of code.

Side remark: Such a tool does not necessarily have to be part of Tiki. It could make development easier it at first it would be stand-alone.

Importance
7

Easy to solve?
2

Priority
14

Demonstrate Bug (Tiki 19+)

Please demonstrate your bug on show2.tikiwiki.org

Version: trunk

Ticket ID
7511

Created
Tuesday 25 August, 2020 09:52:55 GMT-0000

LastModif
Tuesday 09 March, 2021 19:09:21 GMT-0000

Comments

Jonny Bradley 25 Aug 20 11:43 GMT-0000

Have you seen these commands maybe?

php console.php help translation:englishupdate
and
php console.php help translation:getstrings

They update the language files with everything that is "translated" and are run as part of the release process, and should be also run periodically as part of the CI in between
Thanks. Didn't know this. How can it help me?

What exactly does "update the language files with everything that is "translated" mean? An automated process can't distinguish between a minor change (like a typo) or a change of meaning of a string, for that, a knowledgeable human is needed.

Also, it seems to have some prerequisites I do not meet. It throws an error:


Can this only be run on a server hosting the site (or the DB)? I ran it on my Linux desktop, which has an up to date copy of the Tiki installation directory. I COULD run it on the server, because I have SSH access to my rented space, but I'm very hesitant to run scripts that I think do not understand fully...

I have now created my own tool, because I wanted to a) be independent from inner workings of Tiki and b) fully understand (and control) what's going on. After some weeks with little progress and a weekend where several use cases with rare exemptions from the ordinary either just wouldn't run as I'd like, or would bloat the code out of proportion, I rewrote the tool from scratch, utilizing PHP's built-in tokenizer to do the job, only the processing of template files were left to my original design, as the tr in curly braces will not be recognized by the tokenizer.

Now I have a tool that creeps into every tiny little corner of a Tiki installation, finds all .php, .tpl and .js files, extracts all tra(), tr(), tr in curly braces, tr_0 in curly braces and compiles neat files with the structure of language.php, plus additional info. All found English strings get "translated" into "dummy" translations that instead of a translation text contain reference to the file where the string came from, and its location in the file. And for every new file, a comment is inserted with the path.

The only thing missing in my tool is a routine to deduplicate the numerous (!) duplicates. I will re-use the algorithm from my tool merge_language, but instead of merely logging duplicates, I will eliminate them.
Hi hman

I'm not sure why you want to be "independent from inner workings of Tiki" as that's not how Tiki works. This is a community project and it works (in my view) because we share code and help each other, it is dev:Software Development the Wiki Way after all!

Apart collecting from the `tr()` strings currently missing from the built in tool, what does yours do that the Tiki one doesn't? Surely adding anything missing from here would be better than starting again completely?

Thanks though 😊

---

hman 10 Mar 21 13:53 GMT-0000

To be honest, I did not look at the "official" tool. Why? Because I noticed that the actual way Tiki translations were dealt with in real life prove to have nothing to do with official. Officially, translation files would be separated into strings in use, strings that were indentified as deprecated, and a future run of the tool would delete those etc.

The idea is fantastic. But in reality, language.php for German (de) never had such a structure (I can backtrace to many Tikis before 18.8...).

I wanted to be independent because I wanted to really catch everything that there is. A really comprehensive snapshot of current reality in a given Tiki installation.

That was driving factor #1.

Factor #2 was that I wanted ease of use, so I wanted tools that really handle language.php and custom.php, analyze whether there are unnecessary duplicates (which can be very frustrating, because of the way the associative array is read by PHP the LAST of such duplicates overrides all prior definitions, which can be frustrating to debug).

This is why my tool operates outside of Tiki (which your command line tools do also!), resting solely on
the source code.

But I wanted something else, and here comes factor #3:

Tikis internationalization is a mere shadow of the hype that it creates. In reality, Tiki is -besides English- only reasonably translated into German, Catalan and Korean. That's it. Just look at the size of language.php in all those subfolders to /lang. Even French can hardly be complete, merely 200 KBytes. Without looking at it, it cannot be complete. Most probably it is a complete translation of a very old Tiki version that only represents a very small number of the strings a modern Tiki contains.

I have seen that the backtracking method I used to generate a more or less complete German translation proved ineffective, English text kept popping up here and there, because you cannot backtrack a complex software like Tiki. You cannot estimate ALL branches the code might take, thus you are missing lots and lots of special use cases. Or error cases.

Although it is always preferrable to translate with a view on the actual use of a string (because only then you can comprehend the context), it's just not practical.

So another approach is "catch it all". From the sources.

---

**hman 10 Mar 21 14:01 GMT-0000**

So in order to enable Tiki users from other language background, I wanted that my tool first creates a "generic language". All strings found are "translated" into a "dummy language". That is, they are written to file as an associative array with English as key and value is two number signs, then the filename where the string was found (basename) and then the line number.

So the file that gets produced can actually be used as a language.php and Tiki will run. Of course, every string everywhere is just
But this file isn't meant to be implemented in a running Tiki, although possible. It is meant to be read by a translator line by line, and the dummy language replaced by real language. To help the translator, he or she could look up the occurrence of the string in the source code, therefore getting a glimpse of the context.

To further help this, for every new file that the tool analyzes, a comment is inserted with the full path to the file.

So the output looks like this:

```php
<?php
/* Extracted tra() calls by
tra_extract2.php V1.1 20210309 by
hman */

$lang_extract_raw = [
// blog_post_status.tpl
"Leave a comment" => "## blog_post_status.tpl:8",
"%0 comment" => "## blog_post_status.tpl:11",
"%0 comments" => "## blog_post_status.tpl:13",
// prefs.php
"Simple" => "## prefs.php:103",
"Advanced" => "## prefs.php:103",
```

But creating new Tiki translation from scratch is a daunting task, so every translators in spe be warned:

It is a massively time consuming job. We are talking of manweeks.

I could only do it because thanks to the corona virus (I never thought I would say something like this ever) I have enough spare time, due to
German short time regulations.

A prospective translator faces > 
20,000 lines of text...

**hman 11 Mar 21 14:03 GMT-0000**

I still got to add deduplication. When that is done, I will upload my tool.

I had some struggles with some Javascript files from the vendor_bundled/vendor and vendor_extra/fullcalendar-resourceviews directories and could not work around those without bloating the code.

At this point I'll have to add that I did a nasty and dirty trick: I do Javascript extraction with PHP's built-in tokenizer, too. After inserting a mock <?php at the beginning the tokenizer will take Javascript and break it down just nicely. With the only exception of the above mentioned directories. Since the official Tiki script excludes them also, I have decided to also exclude them. The dirty PHP trick works perfectly on all other Javascript.

So, with deduplication still WIP, I counted 316 lines of tr() in Javascript in Tiki 18.8. I'll upload my results file.

**hman 15 Mar 21 22:02 GMT-0000**

To show you the output of my tool, and why deduplication is needed, I'll upload the raw output: 39,000 lines... That's how many strings rest in Tiki's source code in calls to tra(), tr etc... Gathered from .php, .tpl and .js. PHP and JS get processed using PHP's built-in tokenizer, .tpl is dealt with a procedure I wrote. When deduplication has been added, I'll upload my tool here.

**hman 11:25 GMT-0000**

This is my deduplication tool, in the version for Javascript. It gracefully handles single line comments (not multi line!) as my tra_extract2 generates them to mark the paths of processed files, to create an inner structure that does not disturb the array, and the array definition. It surely isn't the most elegant or efficient code, but it does it's job
and produces nice stdout and logs.

### Attachments

<table>
<thead>
<tr>
<th>filename</th>
<th>created</th>
<th>hits</th>
<th>comment</th>
<th>version</th>
<th>filetype</th>
</tr>
</thead>
<tbody>
<tr>
<td>tra_dedup.js.txt</td>
<td>11:31 GMT-0000</td>
<td>0</td>
<td>Deduplicated Javascript output file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>duplicate_error_js.txt</td>
<td>11:33 GMT-0000</td>
<td>0</td>
<td>Log file from deduplicating the Javascript file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tra_extract_raw.js.txt</td>
<td>15 Mar 21 22:05</td>
<td>0</td>
<td>Raw output of tra_extract, without deduplication. Strings from *.js, in the format of language.js.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>deduptest_js.php</td>
<td>11:27 GMT-0000</td>
<td>0</td>
<td>Tra dedup js, a tool to deduplicate the output of tra_extract2, for the Javascript file.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The original document is available at [http://dev.tiki.org/item7511-Create-a-tool-that-extracts-all-text-to-be-displayed](http://dev.tiki.org/item7511-Create-a-tool-that-extracts-all-text-to-be-displayed)