CUNY Academic Commons - Feature #1982

i18n for CAC Featured Content

2012-07-09 03:37 PM - Boone Gorges

Status:	Resolved	**Start date:**	2012-07-09
Priority name:	Low	**Due date:**	
Assignee:	Dominic Giglio	**% Done:**	100%
Category name:	WordPress (misc)	**Estimated time:**	0.00 hour
Target version:	Future release		

Description

Be sure that text strings in CAC Featured Content are localizable, that the plugin specifies and loads a textdomain, and that a .pot file is shipped with the plugin. Here's a quick guide: http://www.farinspace.com/wordpress-plugin-i18n/

Related issues:

Related to CUNY Academic Commons - Feature #1871: CAC Featured Content Rewrite	**Resolved**	**2012-04-30**

History

#1 - 2012-08-25 11:27 PM - Dominic Giglio

Boone,

I'm trying to get these last few cfcw issues outa the way so I can get a featured content widget release out the door.

I'm a little confused by this issue. Here's what I'm having problems with:

1. When I added the esc() functions in an earlier issue, the Codex said to use them "as close to output as possible." Is it the same with i18n? I started putting the __() function in cac-featured-controller.php, but now I'm thinking they should go in the view files to be "closer to the page." And do I also need to add them to the main cac-featured-content.php file, in the form() method, to translate the admin widget text? Or does WordPress translate the admin on its own?
2. I found poedit in the Arch repos so I know I can install it to generate .po files (it was mentioned in the post you link to in the description above). I don't have to translate the plugin though, right? I don't really understand what kind of .po file I need to create, an English .po file? That just doesn't make sense, it's already in English! :-)

#2 - 2012-08-26 10:05 AM - Boone Gorges

1. The gettext functions must be wrapped around literal strings. So:

```
$my_text = __( 'Foo', 'cac-featured-content' );
echo $my_text;
```

or

```
<h2 class="bar"><?php _e( 'Foo', 'cac-featured-content' ); ?></h2>
```

Generally, you don't have to run _() and esc*() on the same strings. If a string is generated from user-provided content, it needs to be sanitized, but is not translatable: in these cases, use esc_*(). If a string is literal in the plugin, it should be translatable, but doesn't need sanitization: use _(). More concretely, when putting out helper text, like the labels on the widget form, use _(). When outputting stuff like widget title, etc, on the front-end (stuff that is either generated dynamically, or provided by the user) use esc_html().

2) Yes, create a .po (convention actually says that the ur-po file should be called .pot) with poedit, and yes, it should be in English. No, you do not have to create translations. The .po file is basically a formatted x-reference file, matching line numbers from the plugin up with strings and their translations. In the case of the "null" translation - your initial .pot file - the translations will be empty. Translators load the .pot file into poedit, do their translations, and save a .po and .mo for their language, which can then be distributed with the plugin if the translator volunteers it to you (which they often do, without solicitation). See, eg, https://github.com/boonebgorges/buddypress-group-email-subscription/tree/master/languages (the .pot is at the bottom of the list, with specific languages higher up).

#3 - 2012-08-26 04:17 PM - Dominic Giglio

OK, this explanation makes a lot more sense.

Unfortunately, it leaves me with a new question. I'm gonna leave #2 alone for right now and just address #1:

There are currently only two areas of the plugin that deal with literal strings. On line 94 of cac-featured-content.php the protected array() $widget is defined, which serves as the structure that will be output by the form() method in the admin. Apparently, PHP's associative array "hash rocket" (=>) assignment does not allow the gettext functions. I get an error about an unexpected "(" using either of the gettext functions: __() or _e(). I don't think I've ever tried to use the return value of a function in an associative array before - except maybe in a loop - but you're almost always using temp variables in a loop situation. I'm gonna search for an answer to this on my own of course, but I figured I'd ask in case you already have a solution in mind.

The second place that literal strings are used is in the error() method I defined for the helper class ([line 163](#) of cac-featured-helper.php). That static method is called in strategic locations throughout cac-featured-controller.php when the plugin can't find essential content for the chosen resource. It should be simple to add the gettext functions in this case (I need to edit that function to match the slightly altered view HTML structure anyways), but I'm forced to ask: if my question above cannot be easily addressed, is it even worth adding the gettext functions to *just* the error messages?

#4 - 2012-08-26 04:55 PM - Dominic Giglio

Doesn't look good:

http://stackoverflow.com/questions/3273494/php-use-function-return-as-array-data-in-static-class-definition

#5 - 2012-08-27 07:57 AM - Boone Gorges

> is it even worth adding the gettext functions to just the error messages?

I'll be the hard ass here :) Any time we ship plugins in the wordpress.org repo, they must be fully localizable, no exceptions. It's generally very easy to do (you should get in the habit of doing it as you develop anyway, so you don't have to go back and fix later). And releasing non-localizable plugins alienates a large percentage of possible user bases (though, as you note, it would only be for a small handful of strings in this case).

As for your technical issue, I suggest just doing what the stackoverflow thread suggests. The problem is not that the hash rocket doesn't allow function values, it's that you can't use function values when defining a method's default properties. So do something like this instead:

```
protected $widget = array();

function __construct() {
    $this->widget = array(
        'name' => __( 'CAC Featured Content', 'cac-featured-content' ),
        'description' => __( 'A widget that allows you to feature content from across the Commons, including B
logs, Groups, Wiki Articles, and People.', 'cac-featured-content' ),
        // etc
    );

    // do whatever else your constructor does
}
```

#6 - 2012-09-09 02:06 AM - Dominic Giglio

The specifics of this issue have been addressed in my update to [#1998](#).

If everything looks good in that update, we can close this one.

#7 - 2012-09-11 10:23 AM - Dominic Giglio

- Status changed from Assigned to Resolved

- % Done changed from 0 to 100

Renamed default language file to cac-featured-content.pot

Commit: https://github.com/castiron/cac/commit/60d2280ab416751dbceedc462fdf56377ba70559

#8 - 2012-09-11 05:57 PM - Boone Gorges

Dy-no-mite!

#9 - 2012-09-11 06:01 PM - Dominic Giglio

I'm in the middle of Calc homework now, and I've got Assembly language tonight, but I'm hoping to get the rest of the "repo work" done on this when I get home, so I can push a release [and knock a few more 1.4.4 issues out of course :-)].