

Bugs & Wish list

Custom CSS values which overrides chosen CSS (and are kept in database to be future-proof) | Tiki Wiki CMS Groupware :: Development
Custom CSS values which overrides chosen CSS (and are kept in database to be future-proof)

Status

✖ Closed

Subject

Custom CSS values which overrides chosen CSS (and are kept in database to be future-proof)

Version

4.x

Category

- Feature request

Feature

Theme: Look & feel, Styles, CSS, Theme Control Center
Profile Manager

Resolution status

Fixed or Solved

Submitted by

Marc Laporte

Volunteered to solve

Jonny Bradley

Lastmod by

Marc Laporte

Rating

★ ★ ★ ★ ★ (0) ?

Description

I pick one of the themes included in Tiki (Codex.css for example). It is almost perfect for my needs. I would like to make a few modifications, which are not appropriate to share with the Tiki project.

A- If I edit the styles/codex.css, I will have to redo my change again at each upgrade.

B- If I duplicate styles/codex.css to styles/mycustomtheme.css, I will not be taking advantage of any future enhancements to the theme.

It is not possible to put CSS information as Custom code in Site Identity.

The other benefit is to distribute css adjustments via profiles, like:

http://profiles.tikiwiki.org/site_width

1- Smarty doesn't like the curly brackets used in CSS. {literal} bla bla {/literal} could be used to get around this

2- It is not in <head> </head>

I jsut tested by putting the code below as Custom code in Site Identity and it works 😊



```
{literal} <style type="text/css"> body { background-color: blue; } p { color: yellow; } </style>
```

{/literal}

Entering CSS "as-is" with curly brackets in Site Identity and having it render as internal CSS could be a solution.

<http://www.tizag.com/cssT/internal.php>

or maybe a custom.css should be maintained, and cached from this data.

Related:

http://dev.tikiwiki.org/tiki-view_tracker_item.php?itemId=1014

http://dev.tikiwiki.org/tiki-view_tracker_item.php?itemId=1178

Solution

For 4.0, we have custom head code now.

Custom CSS and Custom JS fields in Tiki5

Workaround

In Tiki 3.0:

<http://tikiwiki.svn.sourceforge.net/viewvc/tikiwiki?view=rev&revision=17331>

Importance

9 high

Priority

45

Demonstrate Bug (Tiki 19+)

Please demonstrate your bug on show2.tiki.org

Version: trunk ▼

Demonstrate Bug (older Tiki versions)

This bug has been demonstrated on show.tikiwiki.org

Ticket ID

1320

Created

Sunday 19 August, 2007 13:51:37 GMT-0000

by Unknown

LastModif

Sunday 02 May, 2010 09:21:53 GMT-0000

Comments



jeremyclarke 21 Aug 08 18:11 GMT-0000

I was actually wishing for the same thing. My ideal solution would be similar to how Wordpress handles admin css modifications, just give the user a location and filename they can use to have custom css and

always check for that file, e.g. /user/custom.css. That way the special styles exist across installations. Alternately, you could allow users to create a custom css file for a specific theme, like /user/codex.css. This would stop the new css rules from taking over when you are using a different theme.

For bonus points, you can show a simple form in the settings for look and feel that displays and can edit (if it's 777) the contents of the custom css file.



Gary Cunningham-Lee 25 May 09 11:15 GMT-0000

This can be done in effect in Tiki 3.0 via the theme option feature. True, it does require making a new css file and putting it in the 'parent' theme's styles directory (in an options directory). But it will be future-proof, since it won't be overwritten.

I think it won't be hard to have a textarea in Look and Feel admin to input some CSS for the html head (Tiki 4.0 if not sooner), which will apply globally (all themes) unless it has some logic to specify which theme(s) to be used with.

Attachments

filename	created	hits	comment	version	filetype
----------	---------	------	---------	---------	----------

No attachments for this item

The original document is available at <https://dev.tiki.org/item1320-Custom-CSS-values-which-overrides-chosen-CSS-and-are-kept-in-database-to-be-future-proof>