

EditorDev

The Wiki editors, even the "WYSIWYG" ones, have several issues.

RFEs

- See "Plug-ins and other magic", "HTML and Wiki Pages", "Previewing Wiki Pages", "Saving Wiki Pages", and "QuickTags" below

Who is working here

- [userpageterris](#)
- [userpagemarbox](#)

Lost Work

IE users commonly lose their work if, say, mysql has crashed before they press the save button. Clicking the back button (if mysql is back up) results in a warning that the page has expired. Clicking OK results in the page being reloaded from disk in editpage.php, resulting in the loss of the last edits. However, Mozilla doesn't exhibit this behavior.

More commonly, sessions can time out while users are typing for a prolonged period of time. This leads to lost work on IE which causes extreme frustration followed by compensating behavior of saving pages too frequently which leads to a lot of bogus revisions and change notifications. Note that the session time is configurable in the "general" admin menu.

This behavior is by far the worst behavior exhibited by Tiki Wiki and has a very negative user impact. This issue must be dealt with as soon as possible.

ML: I agree: Losing work by using the

[bahttp://tiki.org/tiki-index.php?page=UserBookmarkDoc&highlight=bookmarksck](http://tiki.org/tiki-index.php?page=UserBookmarkDoc&highlight=bookmarksck) button is a big pain. Also can happen if you take lots of time to write an article. If there is a time-out, Tiki should offer to save data as a note or something... (offer relogin)

[userPageterris](#): It's more than a big pain. My users are ready to give up on Tiki after having lost several documents. Why does IE think the page has expired? I guess IE doesn't want to cache the page.

[userPageisotopp](#): Because PHP session management goes to great pains to have IE not to cache the page. See <http://msdn.microsoft.com/workshop/author/perf/perftips.asp#Use%20Cache-Control%20Extensions> and http://php.net/session_cache_limiter.

[userPageterris](#): I discovered the "session" settings in php.ini. These helped me tremendously. For example, I was able to increase the session timeout length and specify "none" (blank) for cache expiration. Just search for the text "session" in php.ini. On Windows, php.ini is located in c:\winnt or c:\windows. There are other settings in php.ini that deal with the browser cache. I was eventually able to fix most of the problems by altering php.ini and setting the TW session timeout to a huge number.

Cons to Textarea

- Although this will be fixed soon (hopefully), Mozilla Firefox is unable to find/replace text
- You don't know what you're going to get until you click preview, and you lose your last caret position after you do so

- The Wiki markup is great shorthand, but it's wacky in many ways. XML would be a welcome change to using curly braces and ~'s. For example, why is __ for bold and === for underlining? _ should be for underline. Why are there two _'s and three ='s? This weirdness makes Tiki markup difficult for ordinary people. The only saving grace are the quicklink icons.
 - [userpageterris](#): Unless there is a viable WYSIWYG editor, Tiki is unlikely to have widespread appeal for large-ish documents

Cons to HTMLArea

- The WYSIWYG editor is ignorant of CSS styles and if used will pollute pages with repetitive font and other style information, which can't be changed via the themes or styles
- Existing text doesn't come up in the editor automatically; users get confused and overwrite their work
 - [larrykluger](#): Auto-loading of existing text does work with current version of HTMLArea.
- Creating tables doesn't work very well
 - [larrykluger](#): Most basic user's pages don't contain tables. IMHO, HTMLArea is a "good-enough" solution to solve the basic users' need for a WYSIWYG editor. Currently basic users have **nothing** in their pages because they are too intimidated by the wiki syntax--they become viewers (lurkers) rather than participants.

Previewing Wiki Pages

Preview causes the caret position and other settings to be lost. How about showing preview in a new window?

Saving Wiki Pages

1. The Save button in the wiki editor needs to also be on the top and bottom of the textarea/htmlarea box
2. Need ability to Save wiki page and stay in edit mode

HTML and Wiki Pages

- The Allow HTML checkbox in the wiki editor needs to be stored on a page per page basis. Currently, if you check the box and edit the page again, the box will be unchecked and if you save the page, the HTML tags will appear instead of being displayed as HTML.
- When Allow HTML is checked, the page can't display XML tags. The only way to display XML tags with Allow HTML is to put a space after the leading bracket (<). A module is needed that can turn < into < at runtime.

Plugins and Other Magic

- The "wiki quick help" has a link to "Show Plugins Help". Is this a dynamic list? e.g., if a new plugin is added, will it appear in this box?
 - Why isn't this list sorted alphabetically by plugin name?
- Ironically, it takes a very long time for authors to discover other useful magic incantations like { maketoc} and which demonstrate the true power of non-wysiwyg editing. The "wiki quick help" should also provide this information.

QuickTags

1. [Ability to set background color via ~~ quicktag](#)
2. Need a QuickTag for "in-line" proportional font (similar to the way bold and underline work)

1. QuickTags could be one of Tiki's main drawing points for collaborative authoring if the QuickTags feature was redesigned and supplemented by corresponding changes in the DynamicContentSystem feature. I'll box these suggestions so the suggested improvements can be identified as an integrated whole.
2. Permissions
3. I'd like to be able to set permissions on which QuickTags are available depending on permissions. E.g., I would use QuickTags a lot more if I didn't have to make them available to registered users. Also, being able to control what QuickTags are available by category permissions would be useful. E.g., display only the quicktags that are relevant to the category.
4. Listbox or phpLayers Menus instead of or in addition to icons
 1. I advocate doing away with QuickTags icons for reasons discussed below. They have been the biggest barrier I have faced in creating QuickTags.
 2. I've been using Tiki for several months and I still don't remember what the different icons stand for, have to do the mouseover thing every time. (Although my eyesight isn't bad, I'm old enough that tiny icons don't convey a lot of information.) (I just noticed that icons have already been eliminated in 1.9.2, if this site is any example. We're still using Polaris for a few more days. I'll leave my criticism of them for posterity.)
 3. I think QuickTags would be far more useful if a user could select them from a listbox or a menu structure with text identifiers. Using icons in the listbox or menu entries too might be an option but should not be mandatory. You shouldn't have to track down your system administrator to find out which icons are available for use and where they are located before you can create a QuickTag. This is a major usability flaw, deserving of bug status.
4. Icon selection
 1. Assuming icons are retained for QuickTags, a few changes are definitely needed.
 2. The QuickTags editor provides no method for the user to select from a display of available icons or a list of their file names. It would be nice if available icons might be stored in one directory on the server, displayed in a listbox for selection, then moved to an icons-in-use directory upon selection. Deleting a QuickTag might restore its icon to the available icons directory.
 3. At bare minimum, the QuickTags editing dialog should provide a default icon so people don't have to chase down their administrator every time they want to create a QuickTag. That way they can get on with their work while the admin is figuring out what icon they should use and where it should be located.
5. MyQuickTags
 1. QuickTags would be far more useful if Tiki had a well-designed MyQuickTags feature for MyTiki.
 2. Draw inspiration from the [UserBookmark](#) feature.
 3. Envision a feature like the existing Bookmarks module (see last link) displayed in the sidebar with two editable fields. An end user can, on the fly, clip from a page or enter text to one field and assign it a MyQuickTags name with the other. No need to switch to the MyQuickTags personal admin screen to create one. (This militates against requiring icons for QuickTags.)
 1. Would need to error check to prevent use of name already in use.
 4. In the MyTiki section, the user could manage the MyQuickTags, assign them icons, create and edit them manually, etc.
 5. Make it easy for users to share MyQuickTags with other users.
 6. Enable a similar feature for use by admins and editors so that they can create QuickTags (not MyQuickTags) on the fly without having to navigate to the QuickTags editor..

7. Would need a corresponding Tiki admin tool so that MyQuickTags could be renamed and/or graduated to QuickTags available to everyone with appropriate permissions. It would also be helpful if an admin could review and perform actions on all QuickTags on the system from a single interface, even if they are MyQuickTags.
6. MyDynamicContentSystem
 1. Many of the QuickTags I have created on my system are tags that insert a DynamicContentSystem object (one of the reasons I don't want <http://tiki.org/tiki-index.php?page=UserBookmarkDoc&highlight=bookmarksmy> end users to be able to create or have access to full-fledged QuickTags). The interface to create such objects is cumbersome.
 2. Implement a MyDynamicContentSystem feature that parallels the MyQuickTags feature sketched above.
 3. Consider security implications. E.g., should a user be able to edit a MyQuickTags object once its content appears in a page that has been edited by someone else?

--marbux

Brainstorming

- CSS support — wouldn't it be nice if you could select a style and start typing in a WYSIWYG environment?
 - See [WysiwygDev](#)
- Could the Mozilla editor be used?
 - See [WysiwygDev](#)
- Could XML be used?
 - See [WikiXML](#)
 - What if Tikiwiki supported embedded XSLT references and used them on the fly to convert xml to html? Then I could use docbook or whatever I wanted. However, there are plenty of opportunities to launch a denial of service attack on Tiki Wiki with XSLT. Perhaps Tiki should only support XSLT files that are located on the Tiki server. Perhaps Tiki would even "modify" the XSLT to suit the theme.
 - The slideshow functionality probably wouldn't work with XML. What is needed is an XML description of a slide show. This overlaps with Open Office's documents.
- Is it possible to return to the last caret position after clicking preview?
- Plug-ins are a challenge — open-ended rendering means that it will be impossible to write an editor that can render everything
- [UserPageterris](#): I would be happy if Tiki could open my favorite editor containing the text, so I can edit it there. Upon closing my browser, the new text would appear in the textarea so I can decide whether to click Save or not. This is how Microsoft FrontPage deals with text files.
 - If there was a hyperlink to a .txt file, browsers would do this automatically; however, there's no automatic way to "post" the editor's contents back to the textarea or back to [TikiWiki](http://tiki.org/tiki-index.php?page=UserBookmarkDoc&highlight=bookmarks)<http://tiki.org/tiki-index.php?page=UserBookmarkDoc&highlight=bookmarks>
 - Java could be used as a basic text editor. Web services or regular http could be used to get and post content.
 - A Java applet could act as a go-between via your text editor of choice and the textarea control
- Tarlbot: When I'm editing large tiki pages I often copy the whole Textarea and paste it into BBEdit, do my edits there and move the text back into the textarea and preview and commit. External editors could be cool.