

PhpDocumentor

I would like to propose that Tiki adopt phpDocumentor as the official documentation tool for developers to use when writing code for the project. It makes generation and maintenance of technical (and user) documentation as easy as it can get. I am volunteering to add as much documentation as possible in order to get the effort started. inline source code documentation is complementary to whatever 'external' documentation is maintained in a wiki or with other tools and document formats. They are not mutually exclusive.

The commentary below is paraphrased from an email by Greg Beaver, a lead developer of phpDocumentor. Greg was answering the question of what tool to use for generating technical documentation for tikiwiki.

Since Tiki is written in PHP, the choice (for a documentation generator) is obvious. doxygen was written for C/C++ and they acknowledge openly on the site that the php support is limited. As for other PHP documentors, there is nothing that comes close to the functionality and extensibility of phpDocumentor, not to mention its stability. Version 1.2.1 is rock-solid. However, I am biased and would recommend that you or whoever is in charge grab a copy of each of our competitors and test them out to make a good decision.

Some advantages of phpDocumentor that may not be obvious are all listed in the tutorial found at

http://phpdoc.org/docs/HTMLSmartyConverter/PHP/phpDocumentor/tutorial_phpDocumentor.howto.pkg.html

Perhaps the most important difference from doxygen does not have is the ability to generate highlighted php source with cross-referencing links to documented elements, even going so far as to link to the php manual for pre-defined functions, and to link from class members as complex as `$this->data->field` or `$this->data->method()` (assuming the var `$data` has a `@var` tag defining the object type - it's all in the tutorial). In addition, phpDocumentor supports inline tags and cross-referencing between user-level manuals (written in DocBook XML) and the generated api docs. This means a function that handles a user element can link to the tutorial for that user element (the `parserExampleTag` class in phpDocumentor links to the manual entry for `@example`).

I haven't investigated doxygen too closely, but it is a very nice tool. We stole some of their ideas for tags such as `@internal`. I wouldn't be too concerned if tikiwiki decides to initially go with doxygen, phpDocumentor 2.0 will have a docblock parser that supports doxygen's native format as well as the traditional phpDocumentor tags, and switching will not be too tough. However, the templating of phpDocumentor is pretty straightforward, if you want to make a new look. We self-generate our own manual, so you can read the tutorial as:

http://phpdoc.org/docs/HTMLSmartyConverter/HandS/phpDocumentor/tutorial_phpDocumentor.howto.pkg.html

or in any of the frames converter templates to see how the look can be modified easily.

We are currently in the process of completing the new parser for phpDocumentor 2.0. It's a generated parser that is based off of PHP 5's `zend_language_parser.y`, so it will have the capability to detect syntax errors and handle all PHP 5 elements right out of the box, when php 5 is available. This won't affect pre-existing projects as much, like tikiwiki, but will in the future when they make the move. The docs will move with them.