cbpporter wrote on Thu, 12 June 2008 16:46luzr wrote on Thu, 12 June 2008 18:36Well, this sounds good - for class browser.

Well this was more of an idea for the main documentation presentation, a replacement for the current structure of the Topic++ articles.

This is something of a solution to what I perceive as the tree main disadvantages of the current system:
1. It is far from complete (as in coverage of the API) and quite selective. Some parts are explained in great detail, while others are barely mentioned. It is also not uniform, neither as formating or style.
2. It is not that cross referenced. For example, when I'm browsing the SetStyle function for a class, I would like to have the definition of that style and a srcdoc style article related to what is Chameleon and how to use it, at my fingertips, both only a click way. When I'm browsing the documentation for String, I want to have a list will all the stand alone functions that work on string, preferably ordered by category. For example, I had often had a self written Join function, before I found out that there was one already implemented.
3. The current documentation for classes are hard to read because the text follows every method. This is daunting for someone new to the framework, who is trying to find something and has at least to skim through all that text. And is annoying for someone familiar with the framework, because these persons really don't need to have text displayed for the methods they already know.

But my idea and your approach are anyway not auto-exclusive. Both can use the same information, only the presentation is different. And the analogy with a browser is logical, since I was thinking of extending a browser window with an editor to enter a description for each item and a category for it. This meta information would them be used for the auto generated pages and could also allow atomic commits to that repository you talked about. When someone is using the normal browser, an icon could show that it has no documentation attached, and somebody might be willing to fill it out, submit it and not have to worry about formating or ruining something in a larger document.

Anyway, my ideas are heavily influenced by three really good help systems (IMHO): Delphi help files, PHP documentation and JavaDoc.

Well, I have to say that I have had "internal fight" between the two concepts for years - I mean, more database like concept of documentation as you suggest, where each method has its own entry, and document like approach like we have now.

In fact, a couple of years before, the first Topic++ iteration worked like database.

In the end, I prefer current model because of several tiny advantages (if they are advantages):

- you have little bit more flexible ways to organize docs. E.g. quite often it is nice to document a

group of methods with single description.

- you have all formating capabilities of RichEdit, so you can put in pictures etc (well, this might be possible with some DB schemes as well I guess)

- and I think you are not really loosing the possiblity to present documentation in class browser JUST LIKE YOU SUGGEST - the topics++ are marked with code labels, so in fact existing topics can act as database too.

Mirek

P.S.: 2. - that is only a matter of putting these links into the text