

---

Subject: Re: Docking package (plus examples)  
Posted by [Oblivion](#) on Sat, 24 May 2008 19:22:09 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quote:

And if you have any more of those good ideas please let me know, or try them out yourself.

Actually I have an idea.

As you know, I had planned to add a dockable bar control to the DockCtrl framework. Now that the development of DockCtrl is stopped, I would like to share what I have in my mind. I have studied your framework and I've come to the conclusion that there is no reason that it can't be done (since, it's similar to mine). Here is the idea: A "dockable" DockBar . What I mean is, a dockbar which will have both user defined standard buttons (as usual) and dockwindows which can be "docked" to the dockbar as buttons. This is both theoretically and practically possible. Since the dockwindows have both a label and an icon, we can use them as buttons (or, we can supply a default icon if they don't have one) and by using a simple overlay icon (something like on Windows desktop shortcut) on the dockwindow-buttons, user can easily separate between standard button and a dockwindow-button. As for the behaviour of this dockwindow-buttons, when the user presses the button the dockwindow can pop-up (just like the autohidebar behaviour) and when the user drags the buttoned dockwindow, it can float in its standard dockwindow form. I know that we already have AutoHideBar for similar behaviour, but I think this will save a lot of space in professional projects. Think about having, say, a dockable "virtual functions" or other windows in TheIDE which can be buttonized and docked. Now, I didn't see something like this in any other docking framework (probably this is going to be a unique feature). Imho, it would be both useful and eye-candy. It shouldn't be so hard to implement this (actually, I had started to implement this, but then abandoned due to the reasons I've mentioned in my last post). Since, I think we both have learned a hell lot about docking mechanisms and U++ behaviour throughout our "competition" we can add this feature easily. If it seems reasonable to you, we can collaborate on it or you can do it all by yourself. What do you think about it?

Quote:

Otherwise, if you're looking for a project I believe some people are desiring a port of WebKit, and I'd happily collaborate on it if you wanted help.

Yes, I myself was considering that since Mirek has suggested a plugin/port for the webkit. I've examined webkit and now I'm examining the WxWidgets port of Webkit. Their solution (and interface) seems good. I can plunge into this "adventure" by myself, but probably it would be better to do it as a "party". Well, after June 3, I'm free. And, probably "I'm in!".

---