

---

Subject: Re: DockCtrl (A dockable window widget for U++)

Posted by [Oblivion](#) on Sun, 17 Feb 2008 04:00:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi, finally I have updated the DockCtrl to ver 0.50. Now DockCtrl has all the basic functionality for Win32 platform.

Also, from now on, I synchronize the releases with monthly U++ releases, so it is actually DEV802b.1. Namely, it is in Beta stage So, I am officially supporting the ctrl now and in the future.

This is a major update. You can find the details of the update and the example win32 exe in the first post of this topic. Also, I finally updated the SVN and added the self-explanatory source code of the example exe.

From this version on, DockCtrl has full AutoHide and Tabbing support. The "nested tabbing" is made optional and can be switched on, on-the-fly. Also, tabs has "automatic alignment" feature. by default tabs are bottom-aligned. But this can be changed on-the-fly by the user. In automatic alignment mode, Every tabwindow has it's tabs on the opposite side of it's alignment (e.g, LEFT/right, TOP/bottom and vice versa)

Chameleon (skins) method was broken, now fixed. Also, DockCtrl now has a unified chameleon structure for easy customizing (not complete yet).

I am planning to add internationalization support and (actually, it does have support; all I have to do is to add a \*.t file -- I'm a bit lazy nowadays ) write documentation, add X11 support.

=====

The Easter EGG:

One of the most interesting "feature" of DockCtrl is that now it is Ctrl dependent, therefore it can be "nested". This means, theoretically, you could have infinite number of independent dockctrl framework (yes, not only dockable windows, but also the framework itself in a single application window! Namely, even DockCtrl could be "nested" too! For example, you could even have child DockCtrl frameworks in Dock widgets(!) or other standard U++ or CtrlFrame derived classed! I will write an example code to demonstrate this "feature" (All you have to do is to pass the target Ctrl reference to SetLayout() method. Thats all.

=====

As always is, there should be some bugs (I didn't encounter any yet). I need feedback, so please feel free to report bugs you encounter while playing with the exe. I'm also going rename some classes and methods (I know they are not well named).

Also, code needs refactoring.

Regards.

