Subject: Re: Title() Bug
Posted by mrjt on Mon, 17 May 2010 11:35:36 GMT
View Forum Message <> Reply to Message

I've fixed the title issues in SVN revision 2403. Floating windows now also update their titles, I'm not sure why I didn't do this properly last time

WhenState is executed every time a DockableCtrl changes state (eg from docked to floating, or is close, or is opened). There was a case where it wasn't working (when dragging into tabs), but I've fixed that now. It does correctly call all child tabs when a DockCont changes state.

The idea of the Docking package was to implement docking in a completely Upp style way. So everything would be owned by something (no passing around naked heap pointers), the interface would be based around property methods and it would be relatively simple for the programmer to use. The most difficult bit was that it would sit as an external package to CtrlLib, so there are no alterations or special cases in CtrlLib to accomodate it.

The programmer uses the light-weight DockableCtrl class to wrap the essential requirements for docking around whatever GUI componenets he wants. This can be done by inheritance or by using is as a ParentCtrl with a layout and the class is really just a data container (for size & state data etc) and has almost no control over the docking process. A reference is then passed to the DockWindow and then everything more-or-less takes care of itself.

DockCont is the interesting class. This is a container for one or more DockableCtrl classes and is strictly internal to the Docking system. Unlike DockableCtrl it is non-persistent and they are created/destroyed as required by user actions. For instance: When a DockCont is dragged into another DockCont the tabs from the first are added to the second and then it is destroyed. When a DockableCtrl is dragged out of a DockCont a new one is created to hold it.

What makes this more complicated is that a DockCont can also be 'nested' as a tab in another DockCont. To facilitate this all controls that are owned by a DockCont are actually cast to Values and stored directly in the TabBar derived DockTabBar (all the DockCast and ContCast calls in DockCont.cpp are casting back from these Values). The DockTabBar class knows about these two classes and can draw them into tabs with the correct title and icon.

DockCont really is the 'nastiness' that I have tried to hide from the programmer-fronting interface. There are a lot of complications here such as the fake title bar when docked, managing all the possible user interactions and avoiding infinite recursions (you would not believe how many of these I got during development!).

DockWindow handles all of the wide scope docking control. It's responsible for creating and destroying the DockConts, storing all the global settings and coordinating drag-drops when signalled by the related DockCont. There is a lot of stuff in there to cater for as many user requirements as possible.

There are some other helpful classes: DockMenu is my attempt to make all of the Docking context menus as customizable as possible without complex inheritance, DockConfig is the config window (mainly intended as an example) and DockPane is a modified SplitterFrame that (tries) to do

intelligent repositioning based the size hints from DockableCtrls.

TabBar is an external class that can be used completely independently of Docking. This grew out of TheIde QuickTabs package by unodgs as my requirements grew more complex. The TabBar ctrl is pretty awesome IMO

I hope that's explained some of it (though probably not very well). I'll happily answer any more specifi questions you have. It's a very complex package and I'm quite poroud of it for that reason.