
Subject: opinions - wxWidgets

Posted by [fudadmin](#) on Thu, 08 Dec 2005 01:03:24 GMT

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

from [http://www.osnews.com/comment.php?news_id=11492&limit=no &threshold=-1](http://www.osnews.com/comment.php?news_id=11492&limit=no&threshold=-1)

By rover on 2005-08-08 23:36:26 UTC in reply to "RE[4: ...]"wxglade a layout an throw it into VC, do the callbacks and your're "done"

This works only if your app uses nothing more than the common controls. Any non trivial application requires subclassing widgets, implementing new funtionalty, etc. That's done by hand, no gui builder can help you. Then you face all of wxWidgets API ugliness.

Doing that is a nightmare in wxWidgets. Why do you think it has never really caught? wxWidgets have been around for ages (first version was for Windows 3.1). Why do you think many others have created other toolkits? Its API is ugly and it is very difficult to extend (due to the way it is designed, as a layer on top of other toolkits). To add a new widget to wxWidgets I need to be a WIN32/MFC programmer, a gtk+ programmer and a Cocoa programmer (which, incidentally, means I should know C++, C, and Objective-C) and then maintain 3 codebases. Not nice.

With any modern toolkit those hacks you suggest to reach threadsafety are not required. This is an example of what I said before: wxWidgets is a low productivity environment. You need to implement every time things that should be a part of the library.

That's also the reason there are not third party widgets to wxWidgets. Compare with what is available for Java/Swing, MFC or even Qt.

WxWidgets is slower than MFC or gtkmm (which are comparable libraries). U++ or Qt are not, because they are not wrappers, they are implemented using a low level API. Granted, this is not such a big issue these days of multi-Ghz machimes and GBs of RAM, but it may still be noticeable in a complex interface.

Incidentally, this also means I can create a new widget with U++, Qt or Swing with ease. I don't need to know any other API besides what each one of them provides. Compare this with wxWidgets or SWT.

The ugly comment was about the design of the library, not its look and feel. Although its look and feel isn't entirely native, as it needs to play with the geometry of widgets to achieve its cross-plattform compatibility goal.

The license does not allow those of us who can contribute to such a library to reuse the code. If you are only interested in using the library that is not an issue, but if you are going to examine its internals, to study the code, and to contribute to the library it would be nice that you could copy and paste any snippet of code you found there for your own projects. The LGPL (with or without static linking extensions) thwarths reusability. If I'm going to put the effort that it takes to familiarize myself with the internals of a library I'd rather work with BSD-licensed code that I can reuse without problems.
