
Subject: Very first impressions and.... [FEATURE REQUESTS]

Posted by [mdelfede](#) on Tue, 11 Sep 2007 22:50:12 GMT

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

I'm testing U++ and TheIDE since less than one week, so I will put some (newbie) first impressions and what I'd like to see in next releases

-The concept of packages, nests and assemblies is very good. It forces some order in code, besides of other advantages. I'd prefer a three view on left pane, indeed. I like also how output files are organized.

-The editor is nice, and quite fast, but compared to scintilla found in some other IDEs (codeblocks, for example) it miss many comfortable commands, like rectangular blocks, comment blocks of code, and so on. Why not scintilla inside TheIDE ?

-Too many toolbars are missing; I know that in some IDEs they're too much, but I really think that a SAVE button is a must!. Ok, CTRL-S does the job, but when you have a hand on mouse, it's more comfortable to hit the toolbar.

-I like the layout editor besides 2 points : the first is the way to add a widget... I'd prefer a bin 'a-la-borland', for example. The second is the location of 'anchors' (springs, ecc...); I think they're part of widget, so they should be put in widget property panel. But I really like the layout editor, I tested some wxwidgets editors, and they were all quite uncomfortable.

-Help system should be customizable, so users can add his/her own help files and topics. I think this would help also to have more tutorials from users.

Now, some 'small' things that are missing :

- Draggable and dockable toolbars and panels. If you write a customizable app with many many commands (a cad, for example) they are a must. Those or command line... menus are too slow, and fixed toolbars are difficult to customize by end user. In such apps user normally select a (small) subset of command and put in toolbars, and the less used commands are done with menu or keyboard.

- MDI interface and DOC/VIEW support. I know MDI is old, but many people likes it.

- The parser. I've read in another thread that his problem are macros. So, why not to feed the source in CPP preprocessor before parsing ? I know it's time consuming, but result could be cached on diskfile and updated only on changes... maybe in background.

a last word.... I find U++ code style one of the best I've ever seen. That's a good usage of C++, not a bunch of #defines !