Subject: U++ vs Qt
Posted by 281264 on Sun, 27 Jun 2010 13:54:03 GMT
View Forum Message <> Reply to Message

I have been exploring U++ and Qt (free version of it, of course). One advantage of Qt is how well documented the tool is. In this respect, in my humble opinion, U++ needs to improve. One advantage of U++ is its BSD license that allows you to create commercial applications without having to pay a fee (is this correct?)

I see U++ a bit inclined toward SQL applications. As my plan is to build an engineering application with extensive usage of OpenGL, do you think U++ is appropriate to do it? What limitations has U++ got?

By the way, I am finding problems with the debugger: it does not work properly.

Please, what do you think about the topic?

Subject: Re: U++ vs Qt
Posted by koldo on Sun, 27 Jun 2010 20:11:43 GMT
View Forum Message <> Reply to Message

Hello Javier

Quote:I have been exploring U++ and Qt (free version of it, of course). One advantage of Qt is how well documented the tool is. In this respect, in my humble opinion, U++ needs to improve. One advantage of U++ is its BSD license that allows you to create commercial applications without having to pay a fee (is this correct?)
You are right. If you develop commercial applications not open sourced, you do not need to pay any fee (including Windows).

Quote:I see U++ a bit inclined toward SQL applications. As my plan is to build an engineering application with extensive usage of OpenGL, do you think U++ is appropriate to do it? What limitations has U++ got?
I think U++ is very appropriate for engineering applications. I can credit it. And many other too. The limit is in the extent of the libraries, but we are widening it.

Quote:By the way, I am finding problems with the debugger: it does not work properly.
For Windows MSC compiler is better supported than MinGW, specially in debugging. I advise you to install it. It is free for commercial use (free, not open).

Subject: Re: U++ vs Qt
Posted by andrei_natanael on Sun, 27 Jun 2010 21:23:57 GMT
View Forum Message <> Reply to Message

Hi Javier,

281264 wrote on Sun, 27 June 2010 16:54One advantage of Qt is how well documented the tool is. In this respect, in my humble opinion, U++ needs to improve.
I don't miss the documentation to often. Once you know the basics in U++ you may learn new things from code. One fast way to search for something is to use CTRL+J in TheIde, feed it with function(ality) you're looking for (try with abbreviations and synonyms). Also TheIde help have search functionality.

Quote:
I see U++ a bit inclined toward SQL applications. As my plan is to build an engineering application with extensive usage of OpenGL, do you think U++ is appropriate to do it? What limitations has U++ got?
U++ really rocks in SQL applications development , still it have OpenGL support in GLCtrl (see OpenGL example from reference assembly). In U++ if you want to use OpenGL you have to know it's API. Qt provides a bit more advanced OpenGL support in it's QtOpenGL module.
Quote:
By the way, I am finding problems with the debugger: it does not work properly.

Please, what do you think about the topic?

The simple way of getting ride of problems created with debugger is to write code that doesn't need to be debugged .
I used it only few times and that mostly on non-U++ projects. If your problems are memory leaks, then U++ provide support to find they easily. See *.log file after you build (in DEBUG mode) and run your application (ALT+L in TheIde).
If you follow U++ guidelines you won't have to use the debugger.

Best regards,
Andrei

P.S.: See comparison between U++ and Qt, it may put a spell on you and use U++ forever

---

Subject: Re: U++ vs Qt
Posted by dolik.rce on Sun, 27 Jun 2010 22:48:50 GMT
View Forum Message <> Reply to Message

281264 wrote on Sun, 27 June 2010 15:54By the way, I am finding problems with the debugger: it does not work properly.
Bit off-topic, but: You can use LOG and DUMP* macros. They support some cool tricks of U++ automatic formating, so it is very easy and quick to use. I do most of my debugging that way.
Some examples: int i=4;
Point p(2,3);
String s="abcdefg";
Vector<int> v; v.Add(2); v.Add(14); v.Add(3);
LOG("i has value "<<i<<" s="<<s<<" and point p is "<<p);
DUMP(i);

DUMP(p);
DUMP(s);
DUMPC(v);
When you run the app in debug mode, the values are saved in log file (Alt+L). In release mode, they are ignored.

Honza
