Subject: Re: User lists of "bad" naming of classes, functions etc in U++...
Posted by andrei_natanael on Wed, 18 Nov 2009 11:05:58 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 18 November 2009 09:41
I would not bother 2 more key presses.

Then why we have << operator in TopWindow, THISBACK macro, etc.?
Regarding this, i like more to write win.Add(widget) than win << widget, and callback(this, &Window::Something) instead of THISBACK(Something) even if you type more because even if someone is a newcomer to U++ he figure out that win.Add(widget) will add the widget to win window.
Now that i'm talking about that i may summarize the things which i don't like at U++.
We say about U++ that it's a modern framework using advanced C++, though i don't see many design patterns used here, the platform abstraction is not so well implemented(IMO Chameleon it's not so good, dirty code inside) and there are a lot of macros (some have a good reason why them exists).

```
GUI_APP_MAIN // U++ initialization is hidden by this macro
{
  TopWindow w;
  w.Run();
}
```

It could be written:

```
int main(int argc, char *argv[]) // it is possible to write main() for GUI applications in Windows too
{
  Application app(argc, argv); U++ initialization is hidden in Application class
  TopWindow w;
  return app.Run(w);
}
```

Don't get me wrong, i like U++ that's why i'm using it but here are some bits that i don't agree with because they could be better.

Back to naming conventions if you're using STL with camelCase you may make the difference between you functions, algorithms, etc. and those provided by STL and say: Hey this Sort is provided by U++ not STL (because different naming conventions). However Mirek if you are supposed to re-create U++ which naming convention would you use?