Subject: Re: is there U++ coding style reference?
Posted by mdelfede on Sat, 31 Jul 2010 21:00:51 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sat, 31 July 2010 22:07
Hi Max!

Proper documentation does not have to be written as comments. Theide is actually very good at it. The topic++ system let's you describe each functions inputs, outputs and also how it works in "reference" section of documentation, which can be swiftly shown in code editor just by moving the mouse pointer to the squares on the left side gutter. Wider ideas and concepts should be in "implementation" section and end user manuals in "app" section.

I am well aware that writing documentation this way is not as simple and fast as just typing the comments into the code. But I prefer the little extra work if it keeps the code clean.

Also, well named functions and a good structured code can make wonders

That said, I agree that non-documented code that is not understandable to anyone is useless. I just wanted to point out that there is more than one way to do it

Best regards,
Honza

Hi Honza,

well, I didn't mean that comments should be like that one :

```
// increment i by one
i++;
```

But, when you read this :

```
#endif
  Vector<Scroll> scroll;
  VectorMap<Ctrl *, MoveCtrl> move;
  VectorMap<Ctrl *, MoveCtrl> scroll_move;
  Ptr<Ctrl>      owner;
 };

 Top        *top;
```

you surely agree that :

1) You can't have the minimal clue of understanding what the hell 'top' variable means, without reading 3/4 of CtrlCore code

2) You have no hint either of what are move and scroll_move members

and so on.
Nor you'll find them in TPP topics, because they're mostly private members non accessible from external code, so no use on document them in public TPP files.

BUT, if you have to add something to CtrlCore code, fix some bug or simply try to understand the code you're quickly in a trouble.
You can't either have a clue of what Top is looking at its definition :

```
 struct Top {
#ifdef PLATFORM_WIN32
  HWND          hwnd;
  UDropTarget   *dndtgt;
#endif
#ifdef PLATFORM_X11
  Window        window;
#endif
  Vector<Scroll> scroll;
  VectorMap<Ctrl *, MoveCtrl> move;
  VectorMap<Ctrl *, MoveCtrl> scroll_move;
  Ptr<Ctrl>      owner;
 };
```

So, when I say "better overcommenting than undercommenting" I mean exactly that.
A couple of years ago I had to put my hands in CtrlCore code to add X11DHctrl (and so X11 GlCtrl), ant this costed me a lot of work just to figure out what 'Top' was used for, among other stuffs. And, the bad stuff is that NOW (after a couple of years) I completely forgot all the implications of touching at it so, If should put my hands again on it I'd have to redo all the work from beginning.
So, I guess that (IIRC, of course....) putting something like that :

```
// top : pointer to underlying native window, if the control is native, NULL otherwise
Top top;
```

would have spared me some work at the expense of a short comment line. Worse than that, before the X11DHCtrl control (so, when I had to write it...) 'top' variable was unioned with another var in order to spare some memory; on non-native controls the top was NOT null but had another (IIRC completely unrelated) meaning.

If you look ad wine core graphics code, you'll find tons of these examples, and I defy you to manage to understand what's going on without loosing some months to go through all code and without tons of trials-and-errors patching it.....

Purpose of commenting code is *not* to document it from the user's point of view, but making it quickly readable from developer's point of view, even many years later.

Max
