

---

Subject: Re: Graphics Context and Draw Object  
Posted by [arixion](#) on Mon, 23 Oct 2006 15:05:59 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

luzr wrote on Mon, 23 October 2006 22:53arixion wrote on Sun, 22 October 2006 23:03\*Sigh\*

My main problem is that I'm trying to use the Scintilla Library Classes alongside UPP classes.

Here's how a port (any port) of Scintilla is supposed to work:-

- 1) The port code sets up the Window on the display device.
- 2) The code links a customized Scintilla class to the window wrapper.
- 3) This customized class inherits from ScintillaBase in the Scintilla Core Library, which provides lexing and folding support. ScintillaBase in turn derives itself from an Editor Class, which provides find/ replace, basic text-editing and autocomplete. The Editor Class calls on the helper classes Window, Surface and Menu to help it do its work:-

- Menu generates the auto-complete context menu from another helper class ListBox.
  - Window provides access to windowing functions like resizing, moving, minimizing/maximizing and closing. It is meant, I think, as a link between the port code and the Scintilla Library.
  - Surface provides functions for painting onto specified areas in the screen.
- 4) Editor calls on an AutoSurface Class whenever it needs to do painting. The AutoSurface class basically generates a surface upon which painting methods are called on.

My main problem here is how to create an implementation class for Surface that can draw onto the component area with persistence, because it is useless to paint things like context menus or cursors or whatever else only for a fraction of a second before it is erased again by TimerAndPaint in the Ctrl class. So, does anyone have any suggestions on how to achieve this?

the confused programmer

Well, even scintilla must support repaint on host platform demand (WM\_PAINT, expose event), which is exactly what "TimerAndPaint" does...

In other words, I am quite convinced that as addition to (4) there must some interface which host platform can call to redraw portion of view area. Use that in Paint and you are OK.

Mirek

I shall look at the Scintilla Library code again. But if Scintilla has such a method, which it does seem to have, how would u synchronize it with TimerAndPaint? Or are you suggesting to use that method to send a \*direct\* msg to the display system? Scintilla does have a sendWndMsg

function...

---