
Subject: Re: AggCtrl

Posted by [dolik.rce](#) on Wed, 24 Sep 2008 09:54:56 GMT

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

luzr wrote on Wed, 24 September 2008 10:32: Some initial comments:

- great job

Thanks

luzr wrote on Wed, 24 September 2008 10:32: 370KB to upload here is quite lot - well, it is not really a problem if uploaded once, but I have deleted previous versions. Please, do so too. We have about 300MB for the forum now... Of course, this deserves to be moved into Bazaar anyway OK, I promise to keep my uploads low next time. By moving to Bazaar you mean to the svn folder?

luzr wrote on Wed, 24 September 2008 10:32: I think that you should try to "embed" agg24 into AggCtrl; maybe use "plugin/png" method (one file that includes all .cpp) to avoid package list to be too old. Or just put it into subdir...

There are all the files included through agg24/agg24.h, but I guess it's not exactly what you mean... The problem is, that agg is highly modular - I wanted to give the user a choice to load either all files (with agg24.h) or just those headers that he actually needs. Now as I think of it, it's probably not that important, compiler should not include code which is not used, right? The second (and more important) reason to have all headers in this package is to let Assist++ to read them, and provide user all the features like code completion etc. I'm not too skilled in aggyet, so I use those features a lot to find the functions I need... The only change to agg24 package I would do is to put there separators and organize the code into sections like renderers, rasterizers and so on (same as Aris had it in his plugin/agg24 - I could maybe even use his package with some minor changes)

luzr wrote on Wed, 24 September 2008 10:32: Now a little bit more design related issue:

I believe that correct "final solution" of this should be lower-level. I mean, we should definitely aim for ImageBuffer rather than full Ctrl.

One of fundamental advantages is that in future, drawing operations should be independent of GUI. Think about webservers that have to run without X11 but still have to produce .png files...

(Sure, current Draw does not allow this, but that should change).

Also, hypothetical AggDraw should reimplement all normal Draw methods (like DrawRect). The goal here is to allow existing code using Draw to be used with new renderer.

Mirek

I'm aware of the plans you have with Draw and Agg. I wrote this mostly as quick "patch" to get the functionality I need for my project (fast, platform independent rendering with antialiasing and easy alphablending). This is far from the "final solution", although I think it should not be too complicated to rewrite it to use ImageBuffer - AFAIK Aris already did it I don't now much about the drawing internals of U++ (yet), but if I could convert the agg rendering_buffer to Ximage and win32 pixmap, I think it should be possible as well for any other image storage format - that means ImageBuffer or directly to png, bmp, etc...

The only problem would be (as usually) with printing. I read all the discussions here about

printing and agg, but I can't think about anything else than you - just draw a huge bitmap and break it into rectangles... Btw: In future I may have to write postscript output for the pictures rendered by agg (sounds crazy, I know...), would that be of any help with the printing?
Oblivion wrote on Wed, 24 September 2008 00:40Reason was, "if with no expression" at aggctrl.h lines 25 and 58:

```
#elif PLATFORM_WIN32
```

Thanks for feedback. This was the only change I made after shutting down Virtualbox with my WinXP It worked in GCC this way and I supposed that all the preprocessors work the same... So I learned something new today

Bye
Honza