Subject: Wayland & U++
Posted by dolik.rce on Tue, 16 Aug 2011 20:32:38 GMT
View Forum Message <> Reply to Message

Hi everyone!

dolik.rce wrote on Sun, 24 July 2011 22:11mirek wrote on Sun, 24 July 2011 20:56Next steps now should be

- macosx
- android
- maybe "gtk backend" - one which is more tightly bound to gtk
- ditto for QT

Volunteers needed You forgot to add the Wayland display server to your list The drawing part will be probably quite similar to framebuffer, so it shouldn't be very difficult.

I already tried to compile wayland and get it running on my Arch Linux with mixed results - the server runs, the clients don't But as soon as I get it to work, I promise to volunteer
As I promised, I'm here to volunteer . I've finally compiled everything necessary to run wayland as well as the example compositor and client applications. I also tried to incorporate U++ into the examples - and here comes the first problem

I started with the simplest available example (just about 200 lines of code) and tried to put in some U++ drawing code. But as soon as I add the Draw package into the project, I started getting these runtime errors:Quote:unknown object (138760168), message create_buffer(nhiiuo)
unknown object (2049), message attach(oii)
wl_display@1: error 1: invalid arguments for wl_shm@6.create_buffer
disconnect from client 0x9637030
AbortedThis happens as soon as I link Draw to the executable, even when using no code from the package (not even include Draw/Draw.h). Based on this behavior I figured it is related to INITBLOCKs and tracked the problem down to call of sInitFonts() in Draw/Font.cpp:49. I have no idea how can it influence the wayland code (any hints?), but as soon as I commented it out, the problem disappeared. It also works fine when explicitly called from CONSOLE_APP_MAIN after the wayland display is initialized.

So far I have just very basic stuff, drawing an Image on the screen. In the next days, I'll begin to incorporate this into the rainbow framework. As I already said, it should be pretty straight forward. I will use this thread to keep you informed about the progress. Also I can post the package if anyone is interested (and has libwayland-* libraries ready)...

Best regards,
Honza

Subject: Re: Wayland & U++
Posted by Tom1 on Thu, 18 Aug 2011 07:08:27 GMT

Hi,

I'm running Kubuntu Natty and wayland package is currently broken/uninstallable there due to some cairo dependency that has been deliberately disabled after causing other problems somewhere else. Nevermind, I'm sure it will get fixed in a year or two...

Anyway, after reading several articles on the web about what's going on with the GUI on Wayland and Windows 8 for that matter, I finally understood -- something the rest of you guys obviously figured out a long time ago -- that Rainbow backed with Painter are the most critical components required to keep U++ as a modern platform running on top of all important targets. The key issue here is the inevitable separation of (compositing) display server and graphics rendering -- now being moved to a responsibility of the application itself. This already happened on Windows a bit behind the scenes as GDI continued to work just fine, although the original architecture was heavily changed. Now Linux is on the line with Wayland coming up.

In future I can imagine different HW accelerated rendering solutions surfacing that will give more speed. I hope it will then be possible to wrap them inside the Painter API for easy utilization.

So, I wish to thank all of you people involved for your work on this important subject.

Best regards,

Tom