

---

Subject: U++ GTK Wayland Port

Posted by [Klugier](#) on Sun, 15 Sep 2024 21:33:28 GMT

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

---

Hello,

I would like to announce that I am currently working on porting U++ to Wayland using our GTK backend. I did some progress in this area:

- SSD (Server Side Decoration) variant works just fine with minor issues. Please, keep in mind that SSD is not supported on many distros, currently KDE support it properly
- CSD (Client Side Decoration) variant works, but there are more problems there in comparison with SSD variant
- In general the responsiveness of the app is much higher in comparison to X11

However, there are some problems as well:

- When backing to previous window after closing dialog window, focus is not being properly restored with CSD
- Menubar positioning is not optimal for example when you spawn menu near the right end of the screen it will be rendered outside of the screen instead of inside
- Second menubar on Gnome doesn't work for some reasons - CSD variant
- Drag and drop needs to be properly implemented on both CSD and SSD
- OpenGL doesn't work (This is out of scope for initial release)
- Window positioning is not optimal. However, it depends on the compositor since it is very powerful in this area for example in comparison with X11
- Probably more...

I attached screenshot from Ubuntu 24.04, where U++ is running in Wayland session and windowing is done using Wayland, not X11:

If you want play with it you can try my branch [klugier/gtk-wayland-initial-version](#) in main u++ repo. Also, once running, please make sure that `CtrlCore/GtkApp.cpp` file has following backend initialization order:

```
#if GTK_CHECK_VERSION(3, 10, 0)
  gdk_set_allowed_backends("wayland,x11"); // this fixes some wayland issues
#endif
```

Alternatively, you can add `WAYLAND` compilation flag and it should force prioritizing Wayland backend over X11.

Klugier

---

## File Attachments

1) [WaylandCSDDemo.png](#), downloaded 456 times

---

### About TheIDE



*The***IDE**

Version: 17412  
(64 bit) (CLANG) (C++17) (SIMD) (Gtk::Wayland)  
Compiled: 09/15/2024 21:23:04  
/home/klugier/upp/theide  
Using: 269968 KB  
CodeBase: 2844 classes, 74368 items

**Copyright** © 1998, 2024 U++ t  
**Authors:** Mirek Fidler, Iñak  
Massimo Del Fed  
**Contributors:** Sandor Hojtsy, Zsolt  
Augusto Roman, Arij  
Marcus Mauhart, Jan  
Victor Berchet, Ivica  
Michal Mierzejewski  
Bas Wegh, Fabio Bus  
Damien Porquet, Jan  
Raf. Ralf Schneider

- main.cpp
- SVG
  - ../altum\_an...svg
  - ../brown\_fis...svg
  - svg\matrix.svg
  - svg\anne.svg
  - svg\tux.svg
  - ../rg1024\_Pr...svg
  - svg\symbol.svg
  - svg\idtest.svg
  - svg\yinyang.svg

- dirup → FrameTop<Butto
- Load(const char \*filenam
- Enter() → void
- DoDir() → void
- DirUp() → void
- CLASSNAME → typedef Sv
- Serialize(Stream& s) → voi
- LoadDir(const char \*d) →
- LoadDir() → void
- SvgView()
- Paint(Draw& w) → void
- Load(const char \*filenam
- LoadDir(const char \*d) →
- DirUp() → void
- Enter() → void
- DoDir() → void
- Key(dword key, int) → boc
- Serialize(Stream& s) → voi
- SvgView()
- SVGView/main.cpp
  - GuiMainFn\_() → void
  - main(int argc, char \*\*argv
  - GuiMainFn\_() → void

- altum\_angelfish\_01
- anne.svg
- brown\_fish\_01.svg
- Emojione\_1F495.svg
- idtest.svg
- matrix.svg
- rg1024\_Presentati..
- symbol.svg
- symbol2.svg
- tux.svg
- yinyang.svg

pnglib.c  
----- SVGView ( GUI MAIN