

Hello slashupp

Handling 60+ years of inconsistent mess in a single package is not easy. :d I really don't know how it can get more intuitive than this. (You can check vte et al. to see the difference)

Here is an -almost- barebone terminal example with comments.

```
#include <Terminal/Terminal.h>
#include <PtyProcess/PtyProcess.h>

using namespace Upp;

struct Terminal : TopWindow {
    TerminalCtrl term;
    PtyProcess pty;

    Terminal()
    {
        Sizeable().Zoomable().CenterScreen().Add(term.GetSizePos());
        SetRect(term.GetStdSize());    // Optional: Sets a 80x24 display
        term.WhenOutput = [=](String s) { pty.Write(s); }; // Writes user input to the pty..
        term.WhenResize = [=]()        { pty.SetSize(term.GetPageSize()); }; // Informs the pty/system
        about the terminal size change
        pty.Start(GetEnv("SHELL"), Environment(), GetHomeDirectory()); // Runs the user shell
        (e.g. bash) with the user environment and user home dir.
        SetTimeCallback(-1, [=] { // Polls for the pending data in pty (average performance, but
        simple).
            if(!pty.IsRunning())
                Break();
            term.WriteUtf8(pty.Get()); // Writes the data from the source (in this case, pty+shell to
        terminalctrl...
        });
    }
};

GUI_APP_MAIN
{
    Terminal().Run();
}
```

If you need more help, just ask, I'll try my best.

Best regards,
Oblivion
