Subject: Re: A terminal emulator widget for U++
Posted by Oblivion on Fri, 29 May 2020 10:40:48 GMT
View Forum Message <> Reply to Message

An example demonstrating the absolute minimum setup for a standalone terminal app, and two different loops:

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

using namespace Upp;

struct MinimalTerminalExample : TopWindow {
   Terminal term;          // A terminal ctrl instance.
   PtyProcess pty;         // A pseudoconsole process instance.
   MinimalTerminalExample()
   {
      SetRect(0, 0, 800, 600);
      Sizeable().Zoomable().CenterScreen().Add(term.SizePos());
      term.WhenOutput = [=](String out)   { pty.Write(out);  };                    // Terminal output
      term.WhenResize = [=]               { pty.SetSize(term.GetPageSize()); };    // Notifies the pty
about page size changes.
      pty.Start("/bin/bash", Environment(), GetHomeDirectory());                   // Start a shell
process on pty.
      // For a reasonable throughput, a callback-based loop is sufficient.
      SetTimeCallback(-1, [=] { term.WriteUtf8(pty.Get()); if(!pty.IsRunning()) Break(); });
   }

//  For very high throughput, below loop (or something better) can be used.
//
//  void Run()
//  {
//     pty.Start("/bin/bash", Environment(), GetHomeDirectory());
//     OpenMain();
//     while(IsOpen() && pty.IsRunning()) {
//        ProcessEvents();
//        String s = pty.Get();   // PtyProcess reads are non-blocking.
//        if(!IsNull(s))
//           term.WriteUtf8(s);
//        int l = s.GetLength();
//        Sleep(l >= 1024 ? 1024 * 10 / l : 10); // Scale...
//     }
//  }

};
```

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

Best regards,
Oblivion