
Subject: Using TerminalCtrl - does not compile
Posted by [slashupp](#) on Tue, 28 Dec 2021 06:57:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

(nightly 16138)

Trying to use Terminal in a `test` package (U++ CtrlLib application with main window), no edits, and as a first-time dry-run I included the Terminal package in `test`, again no edits, & try to compile, got:

```
/home/mais/upphub/TerminalCtrl/Terminal/Page.cpp (82): error: invalid operands to binary expression ('Upp::WString' and 'Vector<Upp::char16>' (aka 'Vector<unsigned short>'))
```

There does not seem to be a ToUtf16()-variant that takes dword as argument and outputs WString

Also, in Terminal.h for `class TerminalCtrl : public Ctrl {` the reference/implementation topic (bluebox in editor's left column) flashes while there is mousemovement but never shows. This may be a Upp-bug since a random check shows same happen in line 16 of Draw.h for `class Draw;`

Subject: Re: Using TerminalCtrl - does not compile
Posted by [Oblivion](#) on Tue, 28 Dec 2021 07:29:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello slashupp,

Should be fixed now. Please check.

Best regards,
Oblivion

Subject: Re: Using TerminalCtrl - does not compile
Posted by [slashupp](#) on Tue, 28 Dec 2021 09:40:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

what did you change so I can make the same change local without having to download the whole thing needlessly

Subject: Re: Using TerminalCtrl - does not compile
Posted by [Oblivion](#) on Tue, 28 Dec 2021 09:47:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

The problem was wchar32 support. ToUtf16() variant is apparently removed (part of wchar32 changes...).

In Page.cpp, ln: 82, change the existing code to:

```
txt.Cat((int) cell.chr, 1); // wchar32 text-copy
```

It should work.

For a quick and heavy test, I suggest you trying out the vtm demo on TerminalExample:

```
ssh vtm@netxs.online
```

Best regards,
Oblivion

Subject: Re: Using TerminalCtrl - does not compile
Posted by [slashupp](#) on Tue, 28 Dec 2021 11:33:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

I want to do my own tests of this ctrl - always best way

observations:

0. initially blank box with blinking box-caret and right-click menu & nothing much to do
1. should by default open either at user's home-dir or in root(/) if user is root
2. no clear and obvious way to specify in code where/what the ctrl should display
...browsing the source & can find no way to specify initial display...

Subject: Re: Using TerminalCtrl - does not compile
Posted by [Oblivion](#) on Tue, 28 Dec 2021 12:12:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello slashupp,

Thank you for your observations. They will certainly help me improve and polish the ctrl.

As for the specific issues you've pointed out:

Quote:I want to do my own tests of this ctrl - always best way

I respectfully disagree. Playing with the reference examples first would be the recommended way, if not the best. Always. Because the provided reference examples demonstrate the different aspects and use-cases of the package. And they are meant to run out-of-the box. Not to mention they are well-tested over different platforms.

Quote:

0. initially blank box with blinking box-caret and right-click menu & nothing much to do
1. should by default open either at user's home-dir or in root(/) if user is root

Again, the reference examples already demonstrate this. IT is the job of pty process to provide the environment, not of TerminalCtrl.

Have you connected the TerminalCtrl to a "source"? (e.g. a pty, via the provided ptyprocess class. See the TerminalExample)

Quote:2. no clear and obvious way to specify in code where/what the ctrl should display
...browsing the source & can find no way to specify initial display...

Documentation certainly needs improvements. While the package already comes with an API doc, which you can display with TheIDE, ReFERENCE examples already show how to do this and more (e.g. even tabbed or splitted displays).

If you need more help let me know.

Best regards,
Oblivion

Subject: Re: Using TerminalCtrl - does not compile
Posted by [slashupp](#) on Tue, 28 Dec 2021 13:07:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK I'll RTFM (sigh)
(I like things to be intuitive, easy, like the rest of UPP/theide)

...

Just for lols...

Quote:I respectfully disagree. Playing with the reference examples first would be the recommended way, if not the best. Always.

Two ways to acquire knowledge/experience:

1. be taught - you hear/see someone else's experiences and need figure out how to emulate that

2. learn - the hard way through own experience

The first way is dicey, can take long time and is repetitive, the second way is surer, shorter, more retentive by nature first time round.

...

Anyway, thx for your patience, I'll sure give the ctrl a hammering, & keep you updated with anything worthwhile.

Subject: Re: Using TerminalCtrl - does not compile
Posted by [Oblivion](#) on Tue, 28 Dec 2021 13:42:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello slashupp

Handling 60+ years of inconsistent mess in a single package is not easy. 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

Subject: Re: Using TerminalCtrl - does not compile
Posted by [Oblivion](#) on Sat, 12 Feb 2022 18:09:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oops, wrong topic. [PLEASE DELETE ME]
