
Subject: A terminal emulator widget for U++
Posted by [Oblivion](#) on Sat, 23 Mar 2019 21:31:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

tl;dr: I am going to publish a terminal emulator widget for U++, in the following month (April).

From the user POV, it is a very simple Ctrl that can be embedded in any U++ applications via traditional U++ way: (E.g. `Add(terminal.SizePos())`)

And it has a very "thin" public api.

Also its page and parser components are completely decoupled, and documented, and can be used seperately in any project (Read: You can write your own terminal emulator that suits your need with ease).

Before I publish it, I'd like to know what features you would like to see in it.

Currently it is more capable than Win10's new ANSI console.

At the moment it supports:

- VT52/ANSI/1xxx/2xx and -partially- 4xx/5xx emulation and Xterm extension.
- Both 7-Bit and 8-Bit modes.
- UTF8
- OCS and DCS
- ANSI colors
- Alternate screen buffer.
- Resize

Planned:

- Copy-paste support
- Scrollback buffer
- Mouse support
- And various optimizations.

I will also publish two reference examples with it: a simple terminal emulator constructed with Terminal widget, and a remote terminal to demonstrate SSH integration.

A screenshot is much more meaningful than words.

(On a humble test setup (Linux 5.0/Gnome/relatively old test AMD machine. Top row: Nano editor, AnsiArt, Vttest. Bottom row: Emacs, Top, RadeonTop).

Best regards.
Oblivion

File Attachments

1) [Terminal Emulator.png](#), downloaded 455 times

Subject: Re: A terminal emulator widget for U++
Posted by [koldo](#) on Sun, 24 Mar 2019 15:15:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Maybe a sample of it could be a kind of new Putty.
Putty is so basic that with any feature you would add, yours would be better. And in U++ :)

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Tue, 26 Mar 2019 09:02:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Quote:Maybe a sample of it could be a kind of new Putty.
Putty is so basic that with any feature you would add, yours would be better. And in U++

Sure. Currently a barebone SSH terminal example (with the upcoming Terminal ctrl), and TERM variable set to "xterm") in U++ is 52 lines of code. (Single file, password authentication).

What I have also in my mind is VirtualGui integration. :)

Think about accessing to your server or machine via a terminal running on web browser with SSL, on-demand.

This would be a very cool and (somewhat unique?) feature. And it's high on my TODO list. :)

Best regars,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Tue, 26 Mar 2019 12:11:12 GMT

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

And here it is. It took five minutes. We now have terminal access over web browsers, using TURTLE :)
It works as expected.

This is the problem with U++: It is so good yet so underrated. :/

Screenshot:

Best regards,
Oblivion

File Attachments

1) [2019-03-26 15-06-40 ekran görüntüsü.png](#) , downloaded 460 times

Subject: Re: A terminal emulator widget for U++
Posted by [koldo](#) on Tue, 26 Mar 2019 12:21:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

:lol:

Subject: Re: A terminal emulator widget for U++
Posted by [Novo](#) on Wed, 27 Mar 2019 22:14:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oblivion wrote on Tue, 26 March 2019 08:11 And here it is. It took five minutes. We now have terminal access over web browsers, using TURTLE :)
It works as expected.

Are you able to run a Release version of your terminal emulator using TURTLE?
I'm asking this because I'm unable to run a Release version of WebWord (a demo app).

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Thu, 28 Mar 2019 07:01:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Novo,

Quote:

Are you able to run a Release version of your terminal emulator using TURTLE?
I'm asking this because I'm unable to run a Release version of WebWord (a demo app).

Yep, it works on Linux on localhost:8088 with latest versions of U++, gcc. I have yet to test it (Turtle build) on CLANG or Windows.

Setting the Ctrl::host to localhost was sufficient.

The only problem I encountered was the loss of caret due to losing focus.

Best regards,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Novo](#) on Thu, 28 Mar 2019 15:57:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oblivion wrote on Thu, 28 March 2019 03:01
Setting the Ctrl::host to localhost was sufficient.

Thanks a lot!

I didn't realize that Ctrl::host is used to connect back from JavaScript to a server.
I still have several problems like I cannot create more than one connection, although Release configuration is forking. And WebWord is using 100% of CPU because it is spinning in a connection loop which doesn't have any timeouts.

Are you experiencing the same problems or is it something related to my web-server setup?

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Thu, 28 Mar 2019 18:21:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:
Are you experiencing the same problems or is it something related to my web-server setup?

Yes, after you've pointed out, I've checked the behaviour and it is indeed eating a lot of CPU cycles (On Linux 5.0, GCC, release mode, AMD ryzen) I get 16% cpu usage in idle state).

Good news is I found a possible candidate and a workaround (or fix maybe?) but Mirek should see it first.

is waiting for connections) seems to be the problem:

A simple workaround or a possible fix is to call Sleep(10) in Ln: 75.

E.g.

```
for(;;) {  
    if(quit)  
        return false;  
    Sleep(10); // Let's eat less CPU cycle while waiting for connections.  
    //...
```

OTOH, there are also bad news:

- 1) It seems impossible to open multiple clients at once (at least on Firefox (latest)).
Symptom: When trying to open a second connection to turtle server, tab immediately closes (yet a second webworker process runs in the background!).
- 2) I've noticed a potential security flaw: Closing the client using the client's "close window" button does not clear the browser tab (or canvas). It should. Or else sensitive data may be visible.

Best regards.
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Mon, 08 Apr 2019 09:45:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

As the day that I'll publish the code on my github repo draws closer, final bits for the first version of TerminalCtrl are coming together well.

- Scrollback buffer is implemented.
- Chameleon (theme) support is implemented.
- VT52/1xx/2xx keyboard (with function keys) support is implemented.
- Conformance levels support is added (i.e. it is now possible to restrict the operation level to e.g. VT52, VT102, or VT220 etc.)

Here is a screenshot.

On the left is our SSH package/SshShell (running in console mode) running on TerminalCtrl, with scrollback buffer enabled.

In the middle is TerminalCtrl running with the widely used "Solarized theme"

On the right is the mighty emacs, with dropdown menus and function keys on default theme.

Suggestions are welcome.

Best regards,
Oblivion

File Attachments

1) [Chameleonized-TerminalCtrl.png](#), downloaded 372 times

Subject: Re: A terminal emulator widget for U++
Posted by [Xemuth](#) on Mon, 13 May 2019 11:47:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Oblivion,

Very good job, it seems to be awesome.
I will try it on my Raspberry with TURTLE !

When do you think you'll publish it ? :blush:

Thanks in advance,
Best Regard

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Tue, 14 May 2019 07:57:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Xemuth,
Xemuth wrote on Mon, 13 May 2019 14:47: Hello Oblivion,

Very good job, it seems to be awesome.
I will try it on my Raspberry with TURTLE !

When do you think you'll publish it ? :blush:

Thanks in advance,
Best Regard

Terminal packaga is already delayed (initially I'd planned to publish it in April).
Good news is that the first version of the package is basiactly complete. So it will be published at

the end of this month.

Currently I am making some profiling and optimizations (which led to 37% performance gain and 30% less memory consumption on average.

Also, now the parser is a complete vt parser, i.e. it can handle every single VT instruction (not every command is implemented though) correctly from VT52 to Vt520)

I am testing it with vttest, xterm's and ncurses's test tools. And fixing some "paper-cuts" that I have noticed. :)

Best regards,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Sat, 08 Jun 2019 14:48:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

A small but "big" update.

Terminal ctrl for U++ is on its way, and here is a short video preview:

<https://vimeo.com/341085501>

It is now mostly compatible with xterm.

- Mouse tracking support added. Terminal can now handle mouse events if they are supported by the applications (See above video for example)
- Bracketed paste mode added.
- "G-sets" support added. It now supports legacy applications that rely on font shifting (g0, g1, g2, g3, gl, gr).
- DEC Technical charset is added.

Now I feel it is eligible for first release. (Probably next friday...)

P.s.: Terminal may seem sluggish in the video. (It is not.)

That's because I encoded it on my good old test machine, and I limited the framerate to 15.

Best regards,
Oblivion.

Hello,

It's been a while since I posted any news about the Terminal package. I had to focus on my other works for a while.

The good news is last week I implemented the final missing pieces for the 0.1 release:

- 256 colors support added.
- ANSI colors support added.
- XTerm dynamic colors support added.
- VT 4xx rectangular area operations are added (copy, invert, move, fill, in both selective and normal modes)
- UDK (DEC's user-defined function keys support) added.
- Lazy resize option is added (to reduce flickers on network terminals such as SSH-based ones)
- Size hint added.
- A .usc file is added to the Terminal package. (The most common options (font, ink, paper, cursor, sizehint etc.) can be set using TheIDE's layout editor. Also it shows a size hint (in calculated cell size) to simplify positioning the widget in the layout editor.
- It is also tested on Windows, and it works well. :) (currently as SSH terminal, in the near future as a frontend for Windows power shell too)

Two notes on the upcoming initial release:

- 1) Terminal package currently does not contain any external code/library. It uses U++, and it's plugins. :)
- 2) Although a virtual terminal requires a pty device, and Terminal package contains one, they are completely decoupled.

Terminal can be used and compiled without PtyProcess. This gives it a huge flexibility

In this regard I will provide 4 basic examples with the package:

- TerminalExample | Uses PtyProcess (currently PtyProcess requires POSIX-compliant operating systems (or possibly Cygwin on Windows.)
- TerminalExampleWithLayout | The same as above.
-
- SshTerminalExample | Does not use PtyProcess. It uses Core/SSH package instead
- SshTerminalExampleWithLayout | The same as above.

Here is the actual code of TerminalExample (36 LOCs total):

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



```

using namespace Upp;

const char *nixshell = "/bin/bash";

struct TerminalExample : TopWindow {
    Terminal term;
    PtyProcess pty;

    TerminalExample()
    {
        term.WhenBell = [=]() { BeepExclamation(); };
        term.WhenTitle = [=](String s) { Title(s); };
        term.WhenResize = [=]() { pty.SetSize(term.GetPageSize()); };
        term.WhenOutput = [=](String s) { PutGet(s); };
        SetRect(term.GetStdSize()); // 80 x 24 cells (scaled).
        Sizeable().Zoomable().CenterScreen().Add(term.SizePos());
        SetTimeCallback(-1, [=]() { PutGet(); });
        pty.Start(nixshell, Environment(), GetHomeDirectory());
    }

    void PutGet(String out = Null)
    {
        term.Write(pty.Get());
        pty.Write(out);
        if(!pty.IsRunning())
            Break();
    }
};

GUI_APP_MAIN
{
    TerminalExample().Run();
}

```

Below was a sort of "final boss" for the first release. It shows the mapscii, an OpenStreetMap implementation for terminal devices, running on the above code and on Gnome-terminal.

On the left is TerminalExample, running mapscii.

On the right is gnome terminal running mapscii

Both are running on 256 colors mode + mouse tracking support. :)

As a final note: Terminal package will be available within this week.

Best regards,
Oblivion

File Attachments

1) [Terminal.png](#), downloaded 207 times

Subject: Re: A terminal emulator widget for U++
Posted by [Xemuth](#) on Tue, 16 Jul 2019 07:34:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Oblivion,

Impressive work !

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Sat, 20 Jul 2019 13:33:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

It is time to unleash the beast :)

The initial version of Terminal package is finally released.

I suggest you read the ReadMe.md on GitHub or the "Overview" doc (qtf) included in the package. It'll give you a better idea of what it is.

Github address of Terminal package:
<https://github.com/ismail-yilmaz/upp-components/tree/master/CtrlLib/Terminal>

Examples are included in the Examples section of my git repo.

Yes, but what is it?

Excerpt form the ReadMe.md on github:

Terminal package is a flexible, easy-to-use yet powerful cross-platform virtual terminal emulation library written in C/C++ for Ultimate++.

It is designed from the ground up with modularity and maintainability in mind. In this respect the package consists of several re-usable classes, only one being the Terminal

widget.

Requirements

- Ultimate++ (ver. >= 2019.1)
- POSIX, Windows (and probably MacOS, though not tested.)
- A decent enough C/C++ compiler that supports at least C++11. (GCC/CLANG/MinGW/Msc)
- Snacks & beer.

Features

*Note that below list is only a summary of the currently supported features. Technical specifications and documentation will be available soon.

- Supports whatever platform Ultimate++ supports. (Linux, Windows, MacOS).
- Supports VT52/VT1xx/VT2xx, partial VT4XX/5XX, and xterm emulation modes.
- Supports user configurable device conformance levels (1, 2, 3, 4, and 0 as VT52 emulation).
- Supports both 7-bits and 8-bits I/O.
- Supports Unicode/UTF8.
- Supports user configurable, legacy "g-set" (G0/G1/G2/G3), and related shifting functions (LS0/LS1/LS1R/LS2/LS2R/LS3/LS3R).
- Supports ANSI conformance levels.
- Supports various terminal state, device, and mode reports.
- Supports DEC VT52 graphics charset, VT1xx line-drawing charset, VT2xx multinational charset, and VT3xx technical charset.
- Supports VT52/VT1xx/VT2xx keyboard emulation with function keys.
- Supports UDK (DEC's user-defined function keys feature).
- Supports user configurable blinking text and blink interval.
- Supports ANSI colors (16 colors palette).
- Supports ISO colors (256 colors palette).
- Supports xterm dynamic colors (dynamic ink/paper/selection colors).
- Supports bright colors.
- Supports background color erase (BCE).
- Supports transparency (i.e. allows background images).
- Supports VT4xx rectangular area operations: copy, invert, fill. erase.
- Supports VT4xx rectangular area checksum calculation and reporting.
- Supports both DEC and ANSI style selective erases.
- Supports alternate screen buffer.
- Supports history/scrollback buffer.
- Has a user switchable scrollbar.
- Supports xterm style alternate scroll.
- Supports resize (and optional lazy resize to reduce flicker on network terminals such as SSH-based ones).
- Supports both immediate display refresh and delayed (buffered) display refresh.
- Supports xterm style mouse tracking: button, wheel, motion, focus in/out events.
- Supports user configurable cursor styles (block, beam, underscore, blinking/steady).
- Supports cursor locking.
- Supports clipboard operations (copy/paste/select all - including history buffer) and basic

drag-and-drop ops.

- Supports bracketed paste mode.
- Has a predefined yet programmable context menu (left mouse button menu).
- Supports window titles.
- Supports bell notifications.
- Supports VT1xx LEDs.
- Supports size hint.
- Supports Ultimate++ style data serialization.
- Supports per-widget customization (i.e no global variables or properties are used).
- Includes a Terminal.usc file for TheIDE's layout editor.

Reviews, bug reports, patches, suggestions are welcome.

Best regards,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Sun, 21 Jul 2019 19:15:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

A Turtle example is added to the package.
You can run the Terminal example in your favorite browser and see how it works. ;)

Terminal: <https://github.com/ismail-yilmaz/upp-components/tree/master/CtrlLib/Terminal>
Example: <https://github.com/ismail-yilmaz/upp-components/tree/master/Examples/TerminalInWebBrowser>

Also a name clash with Turtle is fixed.

Screenshot: Lynx running on Terminal in Firefox (a.k.a "Browserception")

Best regards,
Oblivion

File Attachments

1) [TerminalExample-Turtle.png](#), downloaded 161 times

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Mon, 22 Jul 2019 13:56:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

More updates:

A terminal multiplexing example is added to the package.

This example demonstrates how a terminal multiplexing can be achieved simply by using a splitter widget. Splitter is a container ctrl that can be used to split any parent ctrl into resizable horizontal and/or vertical panes. It also demonstrates the usage of NTL containers with Terminal widget:

Also a mouse capture issue is hopefully fixed.

Best regards,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Wed, 24 Jul 2019 11:24:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

A short video demonstration of Terminal ctrl (simple multiplexing example running mapscii, htop, emacs, lynx nano..):

<https://vimeo.com/349761874>

Best regards,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [koldo](#) on Fri, 26 Jul 2019 06:18:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Terminal based world map viewer is cool :)

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Sat, 27 Jul 2019 09:30:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Yeah it's very impressive tool. I enjoyed it.
It also helps a lot for testing.

Best regards,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Mon, 29 Jul 2019 21:23:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

A small but important update.

True/direct (24-bit) color support has landed.

It is added as a compile-time option this time. (It can be easily switched using TheIDE)
The reason for this decision is that true color support increases the size of each cell by 4 bytes.
(Not worryingly high but I'm being cautious. :))

Latest code of terminal package can be found at:
<https://github.com/ismail-yilmaz/upp-components/tree/master/CtrlLib/Terminal>

Best regards,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Fri, 02 Aug 2019 11:54:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Latest round of updates:

Terminal: Clipboard and DND operations improved.
Terminal: Drag animation for drag-copy operation implemented.
Terminal: WhenClip event added. (Allows client code to inspect/accept/reject pasted/dropped clips.
Terminal: Api doc updated accordingly.

Best regards,
Oblivion

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Wed, 07 Aug 2019 19:24:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Terminal package is updated.

Initial support for a rare but much requested feature in the world of terminal emulators has landed:

- Sixel graphics support is added to the Terminal package. It is now possible to view sixel images, using the Terminal package. However, the initial support is only for external viewing. Embedded images are a TODO.
- A basic terminal example with sixel viewer is also added to the package.
- SixelRenderer class and a convenience function: RenderSixelImage is also added to the package. This class and the function can also be used as a stand-alone sixel renderer.
- Alt-key handling is improved.

A screenshot:

The git repo address of upp-components: <https://github.com/ismail-yilmaz/upp-components>

Reviews, criticism, patches, bug reports, etc., are always welcome.

Best regards,
Oblivion

File Attachments

1) [Terminal-SixelViewer.jpg](#), downloaded 77 times

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Mon, 12 Aug 2019 09:39:52 GMT

Hello,

A small update on the progress of the Terminal ctrl's embedded image (sixel, etc.) support. 8) (Note that I have yet to push changes. I'll probably push the changes next week, as there are some minor issues to solve.)

This will not break the existing behavior. Sixel support will be available in both embedded and external mode.

Best regards,
Oblivion

File Attachments

1) [Terminal-Embedded-Sixel.jpg](#), downloaded 59 times

Subject: Re: A terminal emulator widget for U++
Posted by [Oblivion](#) on Fri, 16 Aug 2019 12:29:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

A day for big updates :)

Terminal ctrl has gained embedded images and Upp::Display support.

Terminal: Embedded images support is added (currently, sixel).

Terminal: Renderer is improved and further optimized.

Terminal: Upp::Display support for image objects are added.

Terminal: LeftAlignedImageDisplay() and RightAlingedImageDisplay() functions added.

Screenshot:

You can always find the new version here:

<https://github.com/ismail-yilmaz/upp-components/tree/master/ CtrlLib/Terminal>

Best regards,
Oblivion

File Attachments

1) [Terminal-Screenshot.jpg](#), downloaded 52 times
