[quote title=mirek wrote on Thu, 18 September 2014 08:47]
Can you show me the code please? I know it sounds trivial, but there are more interpretations to this...

[quote title=Quote:]
That's not that easy since technically I am not written C++ code :).

Anyway, here is a bare bones WinAPI sequence that prints thing as expected:

SetConsoleOutputCP(65001);

uint8 s[] = { 0xC3, 0x80 };
uint8* h = GetStdHandle(4294967285u);
uint32 dummy = 0u;
WriteFile(h, s, 2, &(dummy), 0);

The console is set to UTF8. The À (0xC0) codepoint is encoded with two Utf8 code units 0xC3, 0x80. Now, if the console is set to use a non-bitmap font, it will work very well with a wide range of characters. If the character is not supported, one or more empty little rectangles are rendered. The good news is that you can copy&paste them and the correct information is preserved.

So technically if the String is Utf8, it should work. It works in my code and even with Utf8 U++ Strings prior to your changes. But TheIDE/LocalProcess messes up these values.

I did not manage to encode À as 0x00C0 (or even 0xC000) in Utf16 and send it to the console. Google seems to suggest that the console is inherently 8bit.