Subject: Re: 16 bits wchar
Posted by mirek on Fri, 12 Oct 2007 09:52:16 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 04 October 2007 13:49luzr wrote on Thu, 04 October 2007 17:33OK, patch applied. And you are right about 0xC2, I missed the fact that 0xC0 and 0xC1 is represented by single byte...

Mirek
Did you replace the other one or do you plan to support both versions of Unicode? (5 - mine and what you have - I think 3 or 4). Hope there is no code that depends on six byte Utf-8, but I doubt that this will be an issue for U++.

I will tell you a little about what I'm implementing next. Right now you have a system which allows the use of ill-formatted Utf-8. When transmitted to GUI, it is converted to a valid Utf-16, and if needed you can convert it back to the same Utf-8. This system works, but it kind of creates a bias toward Utf-16.

I do not think that THIS creates bias toward Utf-16 - for Ucs4 (means, 32 bit integers), there is IMO no need to change anything in error escaping method.

Quote:
You will use something like ToUtf8(in.GetLine()) to get a valid Utf from the input for example. Just need to un-error-escape on store. Again, these two different steps will not be necessary in normal apps.

Do you find any utility in this (and not from a GUI programmers stand-point, but a generic library's stand-point)?

Well, actually, I do not see a problem that this is supposed to solve. I guess then if you are interested in valid utf8 only, there is no need for escaping at all - I guess that then it could/should be solved by error message...

Mirek