Subject: Re: 16 bits wchar
Posted by cbpporter on Fri, 12 Oct 2007 11:54:36 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 12 October 2007 11:59P.S.: Really, more and more we are dealing with this, more and more it is apparent that the real solution is

typedef int32 wchar

Yes, these conversions are tricky, but can be done. If you use wchar as a 32-bit value, that would simplify things as in you only need two conversion functions to UCS4 and back, and all the fuss could be ignored. This would be a great idea for GUI. But if I can create some useful things for other standards too and you don't mind including them, I don't know why we shouldn't do it.

luzr wrote on Fri, 12 October 2007 11:59
Anyway, what might be a good idea for now is Utf8 <-> Utf16 conversion utilities, what do you think?

After I finish my round-trip conversion code, I'll get right to it.

luzr wrote on Fri, 12 October 2007 11:59
Also interesting question: While longer UTF-8 sequences are invalid, would not be actually a good idea to accept them as a form of error-escapement? I can imagine a couple of scenarious where this might be very useful... E.g. what are we supposed to to with invalid UCS-4 values after all?

Yes, that would also be a good alternative. I choose the EExx encoding out of two reasons:
1. You already use this approach.
2. Private code-units are more unlikely to be found in exterior sources than overlong sequences, but I guess this depends a lot on circumstances. And as for invalid UCS-4, there are only single surrogate pairs and a couple more values, I'm sure we can find a good place for them somewhere in the private planes (0x0EExxx for example).

And can I use exceptions in these conversion routines?

I really need to document myself on the differences between UCS4 and UTF32.