Subject: Re: 16 bits wchar
Posted by cbpporter on Wed, 26 Sep 2007 05:43:49 GMT
View Forum Message <> Reply to Message

sergei wrote on Wed, 26 September 2007 01:56
I didn't mention that I tested basic read/write performance. UTF handling would add overhead to 8 and 16 formats, but not to 32 format. I also remembered the UTF8-EE issue. UTF-32 could solve it easily. IIRC only 21 bits are needed for full unicode, so there's plenty of space to escape to (without overtaking private space).

The only problem with UTF-32 is the storage space. It is 2/4 times the size of UTF-8 and almost always double of UTF-16. And I don't think that UTF-8EE is such a big issue, you just have to make sure to use a more permissive validation scheme. And what is RTL anyway?

luzr wrote on Tue, 25 September 2007 23:18
I am afraid you expect too much. GetLength returns exactly the same number as GetCount, two names in this case are there because of each fits better for different scenario (same thing as 0 and '\0').

Rather than thinking in terms of UTF-8 / UTF-16... String is just an array of bytes, WString array of 16bit words. Not much more logic there, except that conversions between two can be performed - in conversions there is one and only encoding logic.

Then I don't understand how can you insert values 280-230 into a 8 bit fixed character length format. Are they translated to some code page? And if the values are 8 bit and there are 20 of them, why do I have a length 40 string in the output. And why is the length of the same string 40 and not 20 when I switch over to wide string?