Subject: Re: 16 bits wchar
Posted by sergei on Wed, 26 Sep 2007 12:55:57 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 26 September 2007 07:43sergei 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?

Well, 4MB of memory would yield 1 million characters. Do you typically need more, even for a rather complex GUI app? With memory of 512MB/1GB on many computers and 200GB hard drives, I don't think space is a serious issue now. I was more worried about performance - memory allocation and access is somewhat slower (but not always, for 256-8k sizes it's quite good).

The issue isn't UTF-8EE, it's more of a side effect. The main gain is char equals cell. That is, LString (or whatever the name) can always be treated as UTF-32. Unlike WString, which might be 20 wchars or unknown-length UTF-16 string. Even worse with UTF-8, where String length would almost always be different from amount of characters stored. Replace char is a trivial operation in UTF-32, but might require shifting in UTF-8/16 (if the chars require different amounts of space). Search char from end (backwards) - would require to test every find if it's the second/third/fourth char of some sequence. Actually, even simplier - how do you supply a multibyte char to some search/replace function in UTF-16/32? Integer? That would require conversion for every operation.

Unlike currently, when String is either a sequence of chars OR a UTF-8 string, LString would always be a sequence of ints/unsigned ints AND UTF-32 string. String could be left for single-char storing (like data from file or ASCII-only strings), WString for OS interop, and LString could supply conversions to/from both.