Subject: Re: 16 bits wchar
Posted by cbpporter on Mon, 01 Oct 2007 11:24:46 GMT
View Forum Message <> Reply to Message

I finally finished my Unicode research (I took longer than planed because of computer games...). I read a good chunk of the Unicode Standard 5.0, looked over their official sample implementation and studied a little U++'s String, WString and Stream classes.

I think that the first thing that must be done is extended PutUtf8 and GetUtf8 so that it reads correctly th values outside of BMP. This is not too difficult and I will try to implement and test this.

The only issue is how to handle ill-formated values. I came to the conclusion that read and write operation must recognize compliant encodings, but it also must process ill-formated characters and insert them into the stream. If the stream is set to strict, it will throw an exception. If not, it will still encode. I propose the Least Complex Encoding TM possibility. Non-atomic Unicode aware string manipulation functions will should not fail when encountering such characters, so after a read, process and write, these ill-formated values (which could be essential to other applications) will be preserved. In this scenario, only functions that display the string must be aware that some data is ill-formated.

Next, there should be a method to Validate the string, and a way to convert strings containing ill-formated string to error-escaped strings and back, so we can use atomic string processing if needed. This conversion should be done explicitly, so no general performance overhead is introduced.