Subject: Re: 16 bits wchar
Posted by mirek on Sun, 21 Oct 2007 18:19:14 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Tue, 16 October 2007 05:13OK, fixed all bugs I could find and judging by the the number of runs test I done both automatically and manually I'm reasonably sure that the algorithms are correct. Any input string can be EE-ed to a valid Utf and back, even if the original input is too short.

There is only one issue left. If the original input contains one of our codes for EE-ing (range EE00-EEFF), it will gladly accept it as a valid sequence, thus preserving it's representation. But when you undo the EE-ing, it will think that the input sequence was generated, so it will destroy that given character and replace it with an incorrect 1 byte character. We knew from the start that this issue will arise when the input contains these codes (which normally it shouldn't), but it would be nice if the algorithm would detect these codes and either EE them or just give an error.

Which method would you prefer?

Well, I now might sound stupid, but I got a little bit lost in regard what problem we are really trying to solve.

In fact, I have already asked in some of previous posts...

My suggestion back then was that perhaps, if we are about rigid Unicode processing, we should not error-escape at all.

Well, what might help me: Do you have any real world scenario that can be solved using your routines? Maybe considering it will tell us something about what we are trying to do.

Mirek