
Subject: Re: Will UPP support full UNICODE (21bits long codepoint)?

Posted by [mirek](#) on Fri, 14 Aug 2020 23:39:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

Oblivion wrote on Sat, 15 August 2020 01:02I would love to see in U++ too.

One problem is that going 32-bit codepoints is just the part of the problem, real solution should deal with all composing issues. But I guess it is a good start...

As a first step, I can send in the updated tables for Upp::UnicodeCombine() (full set of canonical compositions, including codepoints > 16 bit). Currently this function is missing a lot of compositions anyway...

They can replace the existing tables (and maybe we can cast down the dword table results to word for the time being? Later we can switch to dword)

But beware, it is a long list of tables, that would deserve another file (unicode.i maybe?)

comb300
comb301
comb302
comb303
comb304
comb306
comb307
comb308
comb309
comb30a
comb30b
comb30c
comb30f
comb311
comb313
comb314
comb31b
comb323
comb324
comb325
comb326
comb327
comb328
comb32d
comb32e
comb330
comb331

comb338
comb342
comb345
comb5b4
comb5b7
comb5b8
comb5b9
comb5bc
comb5bf
comb5c1
comb5c2
comb653
comb654
comb655
comb93c
comb9bc
comb9be
comb9d7
comba3c
combb3c
combb3e
combb56
combb57
combbbbe
combbd7
combc56
combcc2
combcd5
combcd6
combd3e
combd57
combdca
combdcf
combddf
comb72
comb74
comb780
combfb5
combfb7
comb102e
comb1b35
comb3099
comb309a
comb110ba
comb11127
comb1133e
comb11357
comb114b0

comb114ba
comb114bd
comb115af
comb11930
comb1d165
comb1d16e
comb1d16f
comb1d170
comb1d171
comb1d172

What do you think?

Best regards,
Oblivion

Combine is fine, but..

After a lot of thinking, I believe that one step forward is to have

```
int GraphemeLength(const char *s);  
int GraphemeLength(const wchar *s);
```

functions...

Mirek
