Subject: Re: Choosing the best way to go full UNICODE
Posted by cbpporter on Mon, 29 May 2017 12:36:09 GMT
View Forum Message <> Reply to Message

Hi Mirek!

Welcome to my problems, circa 1-2 years in of using U++ :lol:.

I didn't know if you remember, but I was pressuring you left and right to fix these issues, with occasional fixes and what not.

Some fixes managed to get in, but there was no real interest in it, so my solution was to eventually deprecate whole String support in U++, using it as a pure storage. String + custom conversions methods + some custom things, like ToUpper, fixes the problems.

U++ is at least a decade behind on Unicode, with absolutely no good reason. It is not like the issue can't all be fixed in like 3 weeks. I'm sure it would fail 100% of the more difficult compatibility tests.

The good news is that these issues are so rare that you are extraordinarily unlikely to encounter them as an American or European.

As anecdotal proof, this is the first time you encountered issues.

I'm using a very dense data representation method, but even like this, full support is almost 200 KiB of data in every single executable. This is without more advanced stuff, like scripts. I believe 180 KiB is the minimum data that can be stored for the first 3 Unicode planes, this vs. your 2048 entry table in U++, but this gives you lower, upper, title case and Unicode category. I have unit-tests covering these, making sure they are always correct.

String and WString are used for storage. Mostly String.

Contrary to popular belief, Unicode code points are not indexable, so DString does not help. Unicode is glyph based. This + DString being the least beneficent method of storage makes DString not really needed.

In most cases, you can treat the String as opaque.

When not, the ideal solution is an amortized string walker and light algorithm rewrites. You can create an indexed string walker, that seeks, but if you ask it for indices in order, the seek overhead is basically zero. This is only needed for glyph based algorithms, like case conversion and rendering. Or a traditional string walker without indices, a bit faster, but a lit uglier to use, with Get and ++/-- methods.