Subject: Re: Choosing the best way to go full UNICODE
Posted by cbpporter on Wed, 31 May 2017 08:30:09 GMT
View Forum Message <> Reply to Message

It looks like there are many possible ways to go forward. We can try several things and probably a lot of things will work.

As long as we understand that there is no universal way to make Unicode indexable, but on a case by case basis, you can. The only thing you can universally do is to iterate linearly over Unicode.

But I still think I gave you a partial solution so many years ago.

To reiterate:
1. Utf8 to Utf16 and vice-versa must be fixed under all scenarios. We also need to add Utf8 to Utf32, but that is trivial compared to Utf16. So proper error recovery must be implemented and 4 byte long sequences must be converted to surrogate pairs.
2. The Unicode table must be expanded to more than 2048 characters. Maybe not full range, but Unicode is based on blocks. We can move a bit closer to the CJK block, because for CJK, nobody expects more than the basics. Probably 8k at least.
3. Take the opportunity to clean stuff up. Core has had several clean-ups and now C++11 shook thing up. So this might be the perfect opportunity to make sure the designs and interfaces are future proof.
4. Implement DString? I don't know yet. On the other hand, implementing DSting is easy and it can be dropped in its own header and available for use.