Subject: Re: Choosing the best way to go full UNICODE
Posted by cbpporter on Thu, 08 Jun 2017 09:22:53 GMT
View Forum Message <> Reply to Message

Here are the primary stats:
- there are only 5721 characters that have decomposition.
- 2060 out of them have normal decomposition. All of these are two characters. So representing them is easy. Unfortunately the highest CP is 2FA1D. But it is CJK compatibility. I think it should be ignored. Without those the highest codepoint is 1D1C0. But if you ignore a bunch of stuff, like hebrew, hiragana, musical notations I think one could stop even as low as the aptly named character: http://www.fileformat.info/info/unicode/char/2adc/index.htm

Going lower than 0x2adc will soon cut of stuff like Greek. You need 2000 to not cut off Greek.

- the rest of decomposition are 2 to 4, but there are some weird exceptions, like a 18 character one.
- if you stop ar 0x2000/Greek the max CP a decomposition is 8190. If yous top at 0x2adc, it is 12297.

So a dump scheme for the first 0x2adc or a bit higher would be 43.888 bytes. In my lib I already have (256 * 256 * PLANES / BS) * 2 + count2 * 8 = 131456 bytes of data used by uppercase/lower case, so even 43k more is pushing it. I'll investigate how to represent sparsely both the first 0x2adc characters with exact 2 character long decomposition the entire Unicode range.