Why does misdetected Unicode text tend to show up as Chinese characters?

devblogs.microsoft.com/oldnewthing/20140930-00

September 30, 2014

Raymond Chen

If you take an ASCII string and cast it to Unicode,[1] the results are usually nonsense Chinese. Why does ASCII→Unicode mojibake result in Chinese? Why not Hebrew or French?

The Latin alphabet in ASCII lives in the range 0x41 through 0x7A. If this gets misinterpreted as UTF-16LE, the resulting characters are of the form U+XXYY where XX and YY are in the range 0x41 through 0x7A. Generously speaking, this means that the results are in the range U+4141 through U+7A7A. This overlaps the following Unicode character ranges:

- CJK Unified Ideographs Extension A (U+3400 through U+4DBF)
- Yijing Hexagram Symbols (U+4DC0 through U+4DFF)
- CJK Unified Ideographs (U+4E00 through U+9FFF)

But you never see the Yijing hexagram symbols because that would require YY to be in the range 0xC0 through 0xFF, which is not valid ASCII. That leaves only CJK Unified Ideographs of one sort of another.

That's why ASCII misinterpreted as Unicode tends to result in nonsense Chinese.

The CJK Unified Ideographs are by far the largest single block of Unicode characters in the BMP, so just by purely probabilistic arguments, a random character in BMP is most likely to be Chinese. If you look at a graphic representation of what languages occupy what parts of the BMP, you'll see that it's a sea of pink (CJK) and red (East Asian), occasionally punctuated by other scripts.

It just so happens that the placement of the CJK ideographs in the BMP effectively *guarantees* it.

Now, ASCII text is not all just Latin letters. There are space and punctuation marks, too, so you may see an occasional character from another Unicode range. But most of the time, it's a Latin letter, which means that most of the time, your mojibake results in Chinese.

[1] Remember, in the context of Windows, “Unicode” is generally taken to be shorthand for UTF-16LE.

Raymond Chen

Follow