Unicode and UTF-8

- What is Unicode?
- What is UTF-8?
- Illegal Encodings
- Other standards
- Resources

What is Unicode?

Unicode is this character encoding standard

A few notes to the above link:

- Values 0 – 127 are good old USASCII
- 127 (DEL) is all 8 holes on paper tape to maintain even parity. One could delete a character with a hand punch that way.

What is UTF-8?

UTF-8 is a standard for encoding Unicode in 8-bit bytes.

A few notes to the above link:

- If the leading bit is zero, this byte is a USASCII character.
- If the leading bit is 1, this byte is part of a multi-byte character encoding sequence.
- If the leading 2 bits are 1, this is the first byte of a sequence. This allows searching forward or back for the start of a sequence. Continuation bytes start 10, and carry 6 data bits.
- The number of leading 1-bits is the number of characters in the sequence. Currently the maximum defined is 4, which can encode 2,097,151 code points of which 1,112,064 are legal.

Illegal Encodings

There are 3 kinds of illegal UTF-8 encodings

1) There are almost one million code points with values above the allocated maximum U+10FFFF.
2) Overlength encoding happens when a character is encoded in more bytes than necessary. For instance, ASCII space U+20 can be encoded as 0xc0 0xA0
3) “surrogates” enable UTF-16 to represent characters up to 20 bits long. Surrogate codepoints are illegal in UTF-8. (UTF-8 can represent up to 21 bits, but only the first 64K of 21-bit codepoints is assigned).

Other Standards

UTF-8 and UTF-16 are encoding standards in Unicode. There are others for internal use:

- WTF-8 is a superset of UTF-8 that encodes surrogate code points if they are not in a pair (so, ill-formed UTF-16). WTF-8 is a superset of UTF-8 and not part of Unicode.
- CESU-8 uses six-byte sequences to encode surrogates. It is “recognised” by the Unicode Consortium.
- The WTF-8 article also describes UCS-2 and WTF-16.

Resources

There is a dictionary of Unicode code points.
I wrote a couple of shell scripts to convert between byte sequences and Unicode code points:

1) b2u converts a series of numeric bytes to a series of Unicode codepoints. The fancier of the two scripts, it has a **-h** (help) option.
2) u2b converts a series of Unicode codepoints to numeric bytes. The leading *U+* is optional.