Korean

Q: How do Korean letters work in Unicode?

A: There are four main types of encoded Korean letters:
(a) Jamo
(b) Hangul Syllables
(c) compatibility Jamo, and
(d) half-width Jamo.
(c) and (d) are present for compatibility with legacy code pages, and are not required for the representation of Korean.

Q: What are the Hangul Syllables?

A: They can be fundamentally thought of as like composite characters — a compacted representation of certain sequences of Jamo. Of course in practice, these are the main characters in actual use, but from a logical point of view they are simply precomposed sequences, and treated that way in normalization and other processing.

Q: How are the Jamo used?

A: Jamo are divided into three classes: L, V, T (lead, vowel, trail). A standard syllable consists of L V, or L V T. As long as text is represented in sequences of these (e.g. L V L V T L V T L V...) there is no issue. If isolated jamo, such as just an L, are to be represented, there are two ways to do it:

(a) Simply use L on its own (but this must not be followed by V).
(b) Use a sequence with a filler, Vf, to make a standard syllable: L Vf

Similarly, for an isolated V, you could use V (if not preceded by L) or the sequence Lf V, and for isolated T you could use T (if not preceded by V) or the sequence Lf Vf T.

"OBSOLETE" directory?

Q: Do you ever get mixtures of Hangul Syllables and Jamo?

A: Yes, you could. If the text is in NFD, then it will only contain Jamo. If it is in NFC (or unnormalized), most text will be Hangul Syllables. However, Jamo could occur in certain circumstances:

(a) isolated Jamo
(b) pre-1933 orthography Korean text
(c) modern incomplete syllables (e.g. syllables without a leading consonant as used in dictionaries and grammar books)
(d) syllables used for a more faithful phonetic representation of some dialects

In the latter case, there are two possibilities. If the L or V are ancient Jamo, then the entire syllable would be in Jamo. If both are modern Jamo but the T is ancient, then the syllable would be represented by a sequence of two characters: a single code point for LV, followed by the code point for the T: <LV, T>

This is similar to the case of Latin. The NFC form of A + grave + umlaut is <A-grave, umlaut> : part is precomposed and the remainder is not. [JS]

Q: Does this make any difference in how a syllable should be displayed?

A: No. Whether a syllable is represented in the form <L, V, T>, <LVT>, or <LV, T>, it should still be displayed in a single 'cell'.

Q: But how should non-standard syllables be displayed?

A: An L that is not followed by a V should be displayed as if it were the sequence <L, Vf>. A V that is not preceded by an L should display as if it were the sequence <Lf, V>. A T that is not preceded by <L,V> or LV, should display as if it were the sequence <Lf, Vf, T>.

Q: When mapping to KS X 1001 (formerly known as KS C 5601), how should I handle conjoining Jamo?

A: The easiest approach is to first convert the text using NFC . Then convert any remaining conjoining jamo to the compatibility jamo characters. For example, U+1100 (ᄀ) to U+3131 (ㄱ). The conjoining filler characters can simply be removed.

Q: When mapping to KS X 1001-based MBCS character encodings, how should I map the 8,822 Unicode Hangul syllables not covered by KS X 1001?

A: KS X 1001:1998 covers only 2,350 pre-composed Hangul syllables. The same is true of the KS X 1001-based EUC-KR and ISO-2002-KR encodings. The rest of the Hangul syllables in Unicode (8,822 of them) have to be mapped to 8-byte sequences, as specified in Section 3.3 of the annotations to KS X 1001:1998 (KS C 5601-1992). This works as follows:

The first two octets (<0x24 0x54> in GL and <0xA4 0xD4> in GR) signify the beginning of a sequence; they are directly followed by 6 bytes which represent the initial consonant, the medial vowel, and the final consonant of a Hangul syllable, each using two bytes. By this mechanism, full round-trip conversion is possible between Unicode and KS X 1001-based encodings.

Note that both Windows Code Page 949 (Unified Hangul Code) used in Korean MS-Windows and JOHAB — specified as a supplementary encoding in KS C 5601-1992 Annex 3 (= KS X 1001:1998 Annex 3) — equivalent to Windows Code Page 1361 cover the full repertoire of 11,172 Unicode pre-composed Hangul syllables, and thus don't have this mapping problem. [JS]

Q: Why are the KS X 1001 (and KS C 5601) mapping tables in the Public directory on the Unicode site in an "OBSOLETE" directory?

A: Those mapping tables are of historical interest, but may not exactly reflect current mapping implementation practice in all cases. See the Conversions / Mappings FAQ for discussion and alternatives for East Asian legacy character set mapping.

Last updated: - 1/9/2014, 4:56:56 PM - Contact Us