L2/09-080

§1 Difficulties with compatibility ideographs

Document SC2/WG2/N3525 makes the case that CJK compatibility ideographs are not an effective and reliable mechanism to capture glyphic distinctions in plain text. For example, if one whishes to use the shapes 侮 and 侮, simply using the code point sequences <U+4FAE> and <U+FA30> respectively does not *guarantee* the expected rendering: because of the canonical equivalence between the two code points, a process could replace either one by the other.

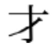
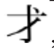
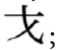
Furthermore, even in the absence of normalization, the expected rendering of <U+4FAE> depends on the locale and/or on font selection. This problem should not be underestimated even when restricted to a single locale. For example, in Japan, the expected rendering of some characters depends on whether one wants to follow the JIS 208 shapes or the NLC shapes, and relying solely on font selection has proven problematic.

§2 Variations sequences

The same document also explains how variation sequences can be used instead of compatibility characters, and are immune to the problem of canonical equivalence. The sequence <U+4FAE, VARIATION-SELECTOR 17> is an explicit request for the shape 侮 and <U+4FAE, VARIATION-SELECTOR 18> is an explicit request for the shape 侮. Furthermore, the expected renderings do not depend on the locale nor on font selection (beyond the support of variation sequences, of course).

The two sequences <U+4FAE, VARIATION-SELECTOR 17> and < U+4FAE, VARIATION-SELECTOR 18> are usable in this fashion because there have been registered in the Ideographic Variation Database (IVD), which is available at http://www.unicode.org/ivd.

§3 Recasting N3530

Document SC2/WG2/N3530 proposes the encoding of a set of CJK compatibility ideographs, for the explicit purpose of providing control over glyph shapes. Following the argument of N3525, we believe that variation sequences should be used instead. In that case, the proposal would be transformed in the registration of *n+1* variation sequences whenever *n* compatibility characters are proposed: one for the representative glyph of the unified ideograph and one for the representative glyphs of each compatibility character. For example, two compatibility ideographs for U+624D are proposed, to distinguish the three glyphs shapes , , and ; using variation sequences, three sequences should be registered, one for each glyph.

§4 The process of registration

The process of registration is rather straightforward. If we ignore for the moment the sequences already present in the IVD, the data supporting the registration request would be two files.

The first file, IVD_Sequences.txt would contain lines of the form:

```
624D; N3530; J0-3A4D
624D; N3530; JH-JTAD0B
624D; N3530; JH-JTB1DA
```

The first field `624D` is the code point of the unified CJK ideograph on which the variation sequence is based. The second field `N3530` is some identifier selected by the proposer that designates the collection to which the sequence belong (here we used `N3530`, but this is entirely arbitrary). The third field is a source identifier selected by the proposer; here we used the source_id of N3530 and the source (in the 10646 sense) of the ideograph.

The second file, IVD_Collections.txt, would contain a single line describing the `jN3530` collection and would be of the form:

```
N3530; J.+ ; http://example.com/N3530
```

where the first field `N3530` is the identifier of the collection. The second field `J.+` is a regular expression matching the source identifiers in IVD_Sequences.txt. The third field is a URL which points to a document describing the intent of the collection, maintained by the registrant.

Additionally, the registrant is encouraged to provide a font to the registrar, solely for the purpose of creating a code chart (see http://www.unicode.org/ivd/data/2007-12-14/IVD_Charts.pdf for the current code chart).

As part of the registration process, the registrar would assign particular variation sequences to the requested sequences. At of this writing, the sequence <U+624D, U+E0100> is already registered, so the three requested sequences would likely be assigned to <U+624D, U+E0101>, <U+624D, U+E0102>, and <U+624D, U+E0103>,.

§5 Leveraging the AJ1 collection?

As of this writing, the current version of the IVD (2007-12-14) contains a single collection, registered by Adobe Systems, to accomodate the Adobe-Japan1 (AJ1) glyph complement. The current sequences in that collection cover the AJ1 glyph complement up to supplement 7. The primary target of this collection is desktop publishing applications.

A comparaison of the glyphs proposed in N3530 and of the glyphs in the AJ1 collection show a certain degree of overlap. For example, the two glyphs for U+4E08 are already present, with the sequences <U+4E08, U+E0100> and <U+4E08, U+E0101>; two of the three glyphs for U+4E0E are already present. About 1/3 of the first 200 glyphs proposed in N3530 are present in AJ1.

From the point of view of the IVD, it is perfectly acceptable to construct the N3530 collection so that it complements the AJ1 collection. In that case, no sequence would be registered for U+4E08, and only one sequence for U+4E0E would be registered. The main advantage of this approach is that it would make the IVD simpler for the end users, as a given glyph shape would occur only once.

There is no danger of AJ1 sequence changing or disappearing from the IVD, as once a sequence is registered, it remains permanently in the IVD (just like a character is never removed nor moved in 10646). There is currently no formal mechanism to declare that a collection extends another collection.

Given the non-trivial overlap between the glyphs targeted by N3530 and AJ1, and given the overlap between the domains of application of N3530 (government and eventually citizens directly) and AJ1 (desktop publishing), constructing the N3530 collection as a complement to AJ1 seems perfectly reasonable. By constrast, it would probably not be very useful to build a collection targeting historical forms as a complement to either N3530 or AJ1.

In any case, the choice between creating a completely independent collection and creating a collection that builds on AJ1 is entirely in the hands of the registrant.