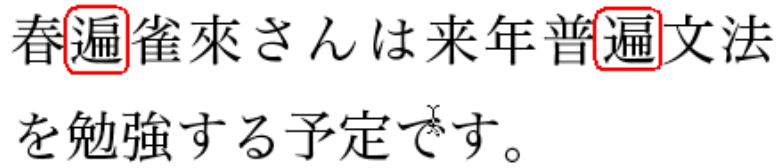
About the registry of Ideographic Variation Sequences

Eric Muller, Adobe Systems Inc.
August 6, 2004

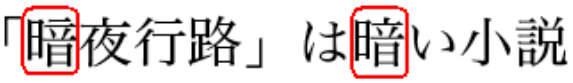
In L2/04-219, we described a process for the registration of Ideographic Variation Sequences. This paper provides some examples of the need for such sequences, shows the registration process at work, and addresses common questions.

1. Why one may want to preserve distinctions

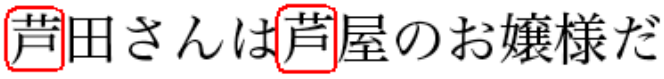
Here are some examples were it can be desirable to restrict the range of glyphs that are to be used to represent an ideographic character. In all four examples, we have two occurrences of the same Unicode character (circled), with different preferred glyph shapes for the two occurrences.



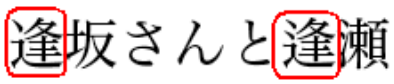
This translates into “Mr. Jack Halpern is planning to study universal grammar next year.” Both circled glyphs render the character U+904D. The first one is the more traditional form, which Jack Halpern chose to write his name (e.g. see The Kodansha Kanji Leaner’s Dictionary, which he edited, where his name is written with this form).



This translates into “‘A Dark Night’s Passing’ is a depressing novel.” Both circled glyphs render the character U+6697. The novel in question was completed in 1937, and the then-current glyph was the form shown above in the title (first circled glyph). The second circled glyph is the modern, Touyou Kanji form of the character. Modern editions of the book tend to use second form.



This translates into “Ms. Ashida is a young lady from Ashiya.” Both circled glyphs render the character U+82A6. Ashiya is a famous town in Japan, and is written here with the more tradional form of the character (second circled glyph). In fact, this traditional form can be seen on the web site of Ashiya (http://www.city.ashiya.hyogo.jp) in the top banner.



This translates into “A date with Ousaka-san”. Both circled glyphs render the character U+9022.

As can be seen from these examples, older forms tend to survive in modern use for people and place names. Using the modern form uniformly would certainly be possible, and would not impair the readability. However, it could be considered rude to use a modern form in writing one person’s name, hence the need to distinguish the forms in rendering. Since this involves people’s names, this

distinction practically needs to be carried in plain text. For example, most databases of CRM systems use plain text only for customer records.

2. Example of registration

Let's assume that some organization (e.g. Adobe) wants to support the distinctions in the first two examples above. Their registration submission will consist of two parts. The first part is the registration of a collection, which may look like:

> Registrant: Adobe Systems Inc
> Contact: Eric Muller, emuller@adobe.com
> Collection URL: http://partners.adobe.com/asn/developer/pdfs/tn/5078.Adobe-Japan1-6.pdf
> Suggested collection identifer: AdobeJapan1

This will result in the following entry being added to IVD_Collections.txt (assuming the collection name AdobeJapan1 is accepted by the registrar):

> AdobeJapan1;http://partners.adobe.com/asn/developer/pdfs/tn/5078.Adobe-Japan1-6.pdf

The second part of the registration populates the collection, and looks like:

> Collection: AdobeJapan1
> Sequence: base: U+904D; registrant id: AJ1-3623
> Sequence: base: U+904D; registrant id: AJ1-14017
> Sequence: base: U+6697; registrant id: AJ1-1161
> Sequence: base: U+6697; registrant id: AJ1-13638

Here the registrant has used its own internal numbering, which in this case happens to be the CID number in the Adobe Japan1 CID collection. The registrar will select four available variation sequences using the appropriate base characters. This may result in the following entries in IVD_Sequences.txt:

> 904D; E0124; AdobeJapan1; AJ1-3623
> 904D; E0125; AdobeJapan1; AJ1-14017
> 6697; E0100; AdobeJapan1; AJ1-1161
> 6697; E0101; AdobeJapan1; AJ1-13638

The choice of the variation selectors to use was made by the registrar, and is based uniquely and selecting *sequences* which are unique. In other words, no existing entry in IVD_Sequences.txt started with "904D; E0124", or any of the other three sequences assigned here. The entry "904D; E0100", if it exists, may or may not be registred in the AdobeJapan1 collection: there is no a priori correlation between two sequences which involve the same variation selector.

It is also important to stress that the registrar did not have to decide if the *glyph shapes* corresponding to the variation sequences submitted for registration were already captured by some existing variation sequence. For example, it could very well be that the glyph shape for AJ1-14017 happens to be the same as that of the sequence U+904D U+E0100, which was registered by another group.

With those sequences registered, the examples can now be represented in plain text:

> ... U+904D U+E0125 ... U+904D U+E0124 ...
> ... U+6687 U+E0101 ... U+6687 U+E0100 ...

If some other group wishes to handle examples 3 and 4, it can register its own collection and sequences:

> Collection: OtherCollection
> Sequence: base: U+82A6; registrant id: Ref-1142
> Sequence: base: U+82A6; registrant id: Ref-7961
> Sequence: base: U+9022; registrant id: Ref-8266
> Sequence: base: U+9022; registrant id: Ref-13408

This will result in additional entries in IVD_Sequences.txt:

> 82A6; E0102; OtherCollection; Ref-1142
> 82A6; E0103; OtherCollection; Ref-7961

9022; E0101; OtherCollection; Ref-8266
9022; E0102; OtherCollection; Ref-13408

Because of the registration process, there is no need to announce which collection is in use in a document. When the sequence U+82A6 U+E0102 is encountered in data, its meaning can be found by lookup in IVD_Sequences.txt: it is Ref-1142 in OtherCollection.

Furthermore, text involving both collections can coexist peacefully:

... U+6687 U+E0101 .. U+9022 U+E0101 ...

Thus, processes such as cut and paste do not need to worry about any state.

3. FAQ

Q: Isn't this AFII all over again?

A: One important differences between AFII and IVSes is that the complete set of registered sequences (across all collections) is not meant to form itself a unique, well organized collection. The process of registration is a fairly straightforward one, that does not involve any technical judgment. The same glyph shape may be registered multiple times, it is up to the registrants to determine if that is the best course of action or not.

The other important difference is that the IVS are designed to augment Unicode rather than being an alternative to Unicode. In particular, a process consuming data that uses the registered variation sequences can simply ignore the variation selectors, and pay attention to the base characters only. Also, the anticipated usage model is that the bulk of the data will use bare base characters, and that variation sequences will be relatively rare.

Q: What if two collections include the same glyph shape for the same base character?

A: If the registrants each ask for a sequence, then two different sequences get registered. There is no attempt on the part of the registrar to combine collections. The only purpose of the registration mechanism is to allow the independant development of collections, and to avoid the need for any identification of a collection in use, either in-band (e.g. using the Plane 14 tag characters) or out-of-band (e.g. using markup). Furthermore, while two collection may have the same representative shape for two sequences, the ranges of variation that are still implicit with each sequence may not match precisely.

Q: Why not use markup?

A: As can be seen in the examples, one common class of uses is for person and place names. Such pieces of data are likely to transit through databases, where it is not practical to keep markup. The use of a plain-text only mechanism make existing systems just work.

Q: Doesn't that imply a lot of work for rendering engines?

A: We need to distinguish two scenarios. In the first, the variation selectors are simply ignored, and the variation sequences are rendered by considering the base character alone. Existing rendering engines should already behave that way without further work, since any release of Unicode can add variation sequences to . The second scenario involves rendering engines which interpret the variation sequences. With modern font technologies, the rendering engine can include processing that does not depend on the exact set of registered variation sequences, nor even on the set of collection which are registered and leave the bulk of the work to the fonts; this is the best we can hope for, as full rendering cannot possibly happen without the apppropriate fonts.

4. Thanks

Thanks to Ken Lunde and Chie Oshima for helping prepare this document.

Document History

Author: Eric Muller

Revision	Date	Comments
1	August 6, 2004	Initial version