White Paper: Ideographic Variation Sequences

Version: March 26, 2008

Dr. Ken Lunde
Senior Computer Scientist, CJKV Type Development, Adobe Systems Incorporated
lunde@adobe.com

An Ideographic Variation Sequence (IVS) is simply a sequence of two Unicode characters, specifically a Base Character (BC) followed by a Variation Selector (VS). An IVS is considered plain text, is standardized because it is in full compliance with *The Unicode Standard*, ultimately resolves to the glyph of an ideograph, is registered, and is unique. This White Paper details the benefits of IVSes, along with how they work in the context of fonts, applications, and OSes.

The Benefits of IVSes

IVSes represent a revolutionary step in the ability to reliably and accurately represent otherwise unencoded ideographs in environments that support only plain text.

The Power of Plain Text

An IVS is a sequence of two Unicode characters that resolves to a glyph, meaning that it is considered *plain text*. A plain text representation is simple, and this simplicity allows IVSes to wield extraordinary power due to their ability to persist in environments where stylized text cannot.

The Encoding of Previously Unencodable Glyphs

IVSes effectively encode ideographs that are unencoded or unencodable, because they are unified with another ideograph whose glyph is considered the encoded or parent form.

The Anatomy of an IVS

An IVS is a standardized and registered sequence of two Unicode characters, named and defined as follows:

- *Base Character*—Defined as any CJK Unified Ideograph, meaning the URO (U+4E00 through U+9FC2), Extension A (U+3400 through U+4DB5), and Extension B (U+20000 through U+2A6D6), and explicitly excludes CJK Compatibility Ideographs, KangXi Radicals, CJK Radicals Supplement, and CJK Strokes. Future CJK Unified Ideograph blocks can thus serve as the BC component of an IVS.
- *Variation Selector*—Defined as U+E0100 (VS17) through U+E01EF (VS256), specifically 240 VSes, and explicitly excludes U+FE00 (VS1) through U+FE0F (VS16).

All Unicode encodings—UTF-8, UTF-16, and UTF-32—equally support IVSes. An IVS resolves to a glyph, which means a GID (*Glyph ID*) of an OpenType® or TrueType font.

The important text-related requirements for IVSes are enumerated as follows:

- The 240 VSes that are used as the second component of an IVS are encoded in Plane 14, which is well outside the BMP (*Basic Multilingual Plane*). Proper handling of non-BMP code points is thus a basic requirement for IVS support.
- An IVS must be recognized and treated as a single unit according to The Unicode Standard. If an IVS is not supported, either by the application or the selected font, its VS shall be ignored and not displayed, but not removed, and its BC shall be displayed as-is.

IVSes are *default ignorable*, meaning that at a minimum an IVS shall be displayed as its BC. Under these circumstances, the VS component shall not display, but shall be preserved. Such behavior is considered IVS-*aware*. IVS-*savvy* behavior means that the correct or otherwise expected glyph is displayed.

The Adobe-Japan1 IVD

Of the 23,058 glyphs in the Adobe-Japan1-6 character collection, 14,665 are classified as ideographs. The Adobe-Japan1 Ideographic Variation Database (IVD) currently registers 14,647 IVSes that support 14,645 of the 14,665 ideographs in Adobe-Japan1-6.* The following table provides an example of two IVSes that share the same BC:

Ideographic Variation Sequence	Base Character	Variation Selector	Glyph	Adobe-Japan1 CID
8FBB E0100; Adobe-Japan1; CID+3056	U+8FBB	U+E0100	辻	3056
8FBB E0101; Adobe-Japan1; CID+8267	U+8FBB	U+E0101	辻	8267

Registering IVSes for a glyph collection involves at least one ninety-day Public Review Issue (PRI) period. The Adobe-Japan1 IVD went through two, specifically PRI 98 and PRI 108. The Adobe-Japan1 IVD was declared final, and thus officially registered, on 12/14/2007.

IVSes & Fonts

The use of IVSes in applications and OSes is driven by the use of fonts that include them. This makes perfect sense, because an IVS resolves to a glyph, and fonts are what contain or otherwise provide glyphs to applications. IVS-enabled OpenType fonts are expected to include a Format 14 'cmap' subtable that enumerates the following two types of IVSes:

- *Default*—Defined as directly encoded through a single Unicode code point, in addition to its IVS.
- *Non-default*—Defined as accessible only through the use of an IVS, meaning that it is otherwise unencoded.

IVS-enabled Adobe-Japan1-6 fonts currently enumerate 13,276 default IVSes and 1,371 non-default ones. IVS-enabled fonts can be built today using AFDKO tools, specifically *makeotf*.

Whether an IVS is considered default or non-default depends on whether its glyph, mapped from its BC, is directly encoded in a Format 4 or 12 (the latter is preferred, if present) 'cmap' subtable. The Format 14 'cmap' subtable is thus designed to depend on a Format 4 or 12 'cmap' subtable, meaning that IVS-savvy clients must use the Format 14 'cmap' subtable, along with a Format 4 or 12 one.

The Kozuka Gothic® and Kozuka Mincho® "Pr6N" font families, consisting of six faces each, have been IVS-enabled from Version 6.003, based on the IVSes set forth in PRI 108. Version 6.004 of these fonts include the 14,647 IVSes that match those that were registered on 12/14/2007.

IVSes & Applications

Some applications, such as web browsers and PDF Forms, work in what can be considered harsh *plain text* environments, and clearly benefit from IVSes. But, any application benefits from IVSes in that they enable the user to enter into their document otherwise unencoded ideographs without the need to support OpenType GSUB (Glyph SUBstitution) features, such as 'jp78', 'trad', 'aalt', and others that allow otherwise unencoded ideographs to be used.

IVSes & OSes

IVSes are expected to be supported at the OS level in the near future. This means that OS-level APIs may soon exist that will allow more applications to take advantage of the benefits of IVSes. IVS-savvy Input Methods are also expected to become available.

* http://www.unicode.org/ivd/ & http://www.unicode.org/reports/tr37/