

The Design of an Extended Japanese Character Set

Ken Lunde

Adobe Systems Incorporated



<ftp://ftp.ora.com/pub/examples/nutshell/ujip/adobe/uc9lunde.pdf>

Today's Japanese Fonts

- Not enough characters
- Too few or no kanji variants
- No alternate metrics—*metrically-challenged*
- No true italics
- No extended ASCII characters
- No macroned vowels—for romanization
- Not cross-platform

Some Improvements...

- **Alternate metrics—breaks the “full-width barrier”**
 - Half-width and proportional symbols
 - Proportional kana
 - Proportional kanji
- **Glyph substitution**
 - Access to unencoded character forms
 - Ligatures
- **Available using technologies such as:**
 - Adobe’s sfnt-wrapped CIDFonts
 - Apple’s QuickDraw GX
 - Microsoft’s TrueType Open

Character Set Issues

- Character- versus glyph-based
- Information interchange versus professional publishing
- How are these concepts related?
 - Character-based character sets = information interchange
 - Glyph-based character sets = professional publishing
- Most Japanese fonts for DTP purposes—that is, TrueType and PostScript fonts—are *not* glyph-based, and are *not* designed for professional publishing
- Time to raise the proverbial “bar”...

An Historical Perspective...

- **ASCII and ISO 8859-1:1987 versus their extensions (Adobe, Apple, and Microsoft)**
 - ASCII enumerates 94 printable characters
 - ISO 8859-1:1987 enumerates 95 more
 - Neither standard enumerates typographically-oriented characters, such as smart quotes, the various dashes, and so on
- **Dingbat characters handled as a separate font resource—Zapf Dingbats**
- **Japanese fonts must evolve...**



First, Choose an Encoding

- **Shift-JIS for first implementation**—for use on today's machines
- **Migration to Unicode**—when the time is right—eased through the use of CID-keyed font technology

Improve Roman Character Handling



- **Support true italics**
 - Current implementations provide algorithmically-obliqued Romans—inadequate for professional publishing
- **Support Extended ASCII**
 - Windows extensions
 - MacOS extensions
- **Support macroned vowels**
 - For transliterating Japanese text
 - Circumflexed vowels is not an ideal solution

The Cross-platform Character Set

- **Design a cross-platform character set...**
 - Proportional Roman
 - Half-width katakana
 - JIS X 0208-1990 (to become JIS X 0218-1997)
 - NEC Row 13
 - IBM select kanji and non-kanji
- **With some extensions thrown in...**
 - *Enhanced Japanese Publishing* (EJP) extensions

“Enhanced Japanese Publishing” Set



- Each Shift-JIS row provides 188 code points
- Allocation:
 - Row 0xF5: Numerals
 - Row 0xF6: Ligatures (Kana, Kanji, and Roman)
 - Row 0xF7: Typeface-dependent Symbols
 - Rows 0xF8–0xF9: Kanji (no variants)
- Rows 0xF0 through 0xF4—940 code points—still available as a valid user-defined range

Unencoded Characters

- The *Adobe-Japan1-3* character collection will enumerate all glyphs, encoded and unencoded
- **Related characters—“true” variants**
 - Simplified kanji become traditional kanji
 - JIS90 kanji become JIS78 kanji
 - Kanji become variant forms
- **Annotated forms**
 - Encircled kana, kanji, numerals, and Roman characters
 - Parenthesized kana, kanji, numerals, and Roman characters
 - Roman numerals

Other Font Resources

- **Ruby fonts—three types**
 - Generic
 - Typeface-specific
 - Typeface-family-specific
- **Build ruby fonts as independent subset Japanese fonts—allows the use of standard input schemes**
- **Dingbat fonts**
 - Typeface-independent characters

Summary

- The inadequacies of today's Japanese fonts need to be addressed before they are suitable for professional publishing
- Adobe's EJP under development
- Most of the issues that need to be addressed exist regardless of the underlying encoding: Shift-JIS or Unicode
- Migration to Unicode is trivial



Adobe