Designing & Developing Pan-CJK Fonts for Today

Ken Lunde

Adobe Systems Incorporated

What Is A Pan-CJK Font?

- A Pan-CJK font includes glyphs suitable for multiple CJK locales
  - China, Taiwan, Hong Kong, Japan, and Korea are the five most important CJK locales
  - "Han Unification" necessitates multiple glyphs for many CJK Unified Ideograph code points
- A Pan-CJK font is Unicode-based
  - No other character set or encoding in common use today can claim adequate CJK support
  - Unicode has become the preferred method for representing text in digital form
- A Pan-CJK font is a lot of work
- How does a Pan-CJK font differ from a Pan-Chinese font?
  - There are three primary Chinese-speaking locales: China, Taiwan, and Hong Kong
  - Some simplified/traditional distinctions have been unified
  - Some distinctions have not been unified, and are thus handled via separate code points
  - A Pan-Chinese font can be treated as a step toward developing a Pan-CJK font

Pan-CJK Font Goals

- Single-locale CJK fonts can be fully-functional with one glyph per code point
  - This is easily demonstrated by today's single-locale CJK fonts
  - Some single-locale CJK fonts still require multiple glyphs for some code points
    - For the purpose of supporting single-locale variant forms
- Multiple-locale CJK fonts require more than one glyph for some code points
  - CJK Unified Ideograph code points are the obvious target and concern

Pan-CJK Font Advantages

- Design consistency across multiple locales
  - Weight
  - Style
  - Width
  - Relative size
  - Other design factors
- Smaller overall footprint
  - A large number of glyphs are shared by two or more locales
  - Subroutinization benefits
- Single font file
- Streamlined testing

CJK Unified Ideographs: URO Versus Extensions

- Premise: CJK Unified Ideograph code points require multiple glyphs
  - Some code points require only one glyph—many are single-source code points
  - Some require four or more glyphs—these are multiple-source code points
- Single- versus multiple-source code points
  - Single-source code points generally require only one glyph
  - Multiple-source code points have the *potential* to require more than one glyph
    - For example, U+4E00 has six sources, but clearly requires only one glyph
- URO (Unified Repertoire & Ordering)
  - 20,902 + 22 (Unicode 4.1) + 8 (Unicode 5.1) + 8 (Unicode 5.2) = 20,940 code points
- Extensions
  - Extensions A (6,582), B (42,711), C (4,149), and D (223) exist
  - The higher the Extension, the greater the percentage of single-source code points
  - The higher the Extension, the lower the percentage of multiple-source code points

CJK Unified Ideographs: URO Versus Extensions (cont'd)

(Number of Sources)	1	2	3	4	5	6	7
URO (20,940)	9%	7%	11%	18%	32%	22%	1%
Extension A (6,582)	9%	27%	41%	20%	3%	>0%	>0%
Extension B (42,711)	45%	41%	14%	1%	>0%		
Extension C (4,149)	91%	8%	>0%	>0%			

CJK Unified Ideographs: URO Versus Extensions (cont'd)

- Significantly more "work" is required for URO code points
  - The URO has a high percentage of multiple-source code points
  - Remember that multiple-source code points have the "potential" to require multiple glyphs
- The higher the Extension, the less "work" that is required
  - Higher Extensions have a higher percentage of single-source code points
- Code-point/glyph-count ratios
  - The URO and Extension A require roughly a 50% increase in glyphs over code points
  - Approximately 30K glyphs are necessary to cover the 20,940 URO code points
  - Approximately 10K glyphs are necessary to cover the 6,582 Extension A code points

Locale-specific Glyph Issues

- Different glyphs for the same locale
  - Code charts versus current sources
  - Some sources have changed over time
    - JIS X 0213:2004 (Japan) is a good example
- Handling CJK Unified Ideographs without sources for specific locales
  - For some, such as those specific to a single locale, it is appropriate to ignore
    - Simplified Chinese is a good example
  - For the remainder, it becomes a policy issue
    - Extrapolate or ignore

Locale-specific Glyph Issues—Specific Examples

- Source glyphs that changed over time: U+8FBB
  - Original Japan source: 辻 (JIS X 0208-1990 36-52)
  - Current Japan source: 辻 (JIS X 0213:2004 1-36-52)
- Multiple-source CJK Unified Ideographs that require only one glyph: U+4E00
  - All sources: 一
- Two glyphs serve more than two code points: U+5668, U+FA38 & U+20F96
  - U+5668 glyphs
    - 器 for Japan, and 器 for all other sources
  - U+FA38 glyph (*ignoring that the distinction which is meant to be preserve cannot be preserved*)
    - 器 for Japan
  - U+20F96 glyph
    - 器 for Taiwan

Pan-CJK Font Implementation Details

- TrueType Collection—via separate font instances
  - Pro: GSUB feature support is not necessary
  - Con: Multiple font instances in application font menus
    - *Can also be considered a Pro*
- OpenType—via 'locl' GSUB feature
  - Pro: Single font instance in application font menus
    - *Can also be considered a Con*
  - Con: 'locl' GSUB feature support is necessary; must choose default locale
- Dealing with the 64K glyph barrier
  - Depends on the extent to which CJK Unified Ideograph blocks are covered
  - This is a clear concern when supporting all of Extension B

Implementing Pan-CJK Fonts: OpenType

- Use the “Adobe-Identity-0” ROS
  - ROS means /Registry = “Adobe”; /Ordering = “Identity”; /Supplement = 0
  - A dynamic, locale-unspecific special-purpose glyph set
- Use the ‘locl’ (Localized Forms) GSUB feature
  - One locale must necessarily serve as the default
    - Simplified Chinese is a suitable default locale due to GB 18030’s broad coverage
  - The remaining locales are supported via substitutions defined in the ‘locl’ GSUB feature
    - Language and script tags must be specified
    - Simplified Chinese = ZHS/hani
    - Traditional Chinese = ZHT/hani (Taiwan) and ZHH/hani (Hong Kong)
    - Japanese = JAN/kana
    - Korean = KOR/hang
- Fully-functional prototype fonts have been built

Other Pan-CJK Font Implementations

- TrueType Collection (TTC)
  - Single font file with multiple font instances
  - Each supported locale has its own font instance
  - Separate font “instances” can share common glyphs
    - Application font menus advertise multiple font instances, one for each locale
  - The ‘locl’ GSUB feature is not necessary
  - Two iPhone fonts, STHeiti-Light.ttc and STHeiti-Medium.ttc, are Pan-CJK TTC fonts
- Composite Font
  - A Composite Font is a “recipe” that references one or more Component Fonts
  - A Composite Font that can specify language/script can theoretically be a Pan-CJK font
  - A Composite Font can be used to overcome or work around the 64K glyph barrier

Pan-CJK Font Support in OSes & Applications

- For TTC, Mac OS X and Windows can generally handle such fonts
  - Applications enumerate fonts differently, so extensive application testing is required
  - Most TTCs to date have been single-locale
  - Multiple-locale, specifically Pan-CJK, TTCs are relatively new
- For OpenType, InDesign CS3 and greater supports the ‘locl’ GSUB feature
  - Can be specified in character and paragraph tags

Unicode Coverage Issues

- Which CJK Unified Ideographs should be included?
- Minimal coverage
  - IICore—9,810 CJK Unified Ideographs
    - 9,706 URO, 42 Extension A, and 62 Extension B
- Intermediate coverage
  - Common standards—GB 18030, Hong Kong SCS-2008, JIS X 0213:2004 and KS X 1001:2004
  - Equivalent to URO, Extension A, partial Extension B, and one Extension C code point
    - GB 18030 requires all URO and Extension A code points, plus six in Extension B
    - Hong Kong SCS-2008 requires 1,712 Extension B code points, plus one in Extension C
    - JIS X 0213:2004 requires 303 Extension B code points
- Maximum coverage
  - All of them
  - This obviously breaks the 64K glyph barrier that is inherent in today's font formats

Locale-specific Considerations

- Hangul
  - Specific to Korean
  - 11,172 code points
- Kana
  - Specific to Japanese, but included in standards of China, Taiwan, Hong Kong, and Korea
  - Accounts for 70% of Japanese text, so the glyph design must be good
  - Requires vertical variants for the small versions and for the long vowel mark
- Vertical variants for punctuation and kana
  - Some vertical variants are locale-specific

Pan-CJK Font Prototype Details

- A “proof of concept” OpenType font
- Makes use of the ‘locl’ GSUB feature
- 44,000 glyphs
  - 29,925 code points
  - URO + Extension A + partial Extension B—close to intermediate coverage
  - Supplied by Changzhou SinoType
- The default locale is Simplified Chinese
  - 11,267 ‘locl’ GSUB feature substitutions for Traditional Chinese
  - 8,106 ‘locl’ GSUB feature substitutions for Japanese
  - 5,312 ‘locl’ GSUB feature substitutions for Korean
- Its glyphs have not been extensively checked
- An IICore subset version includes 15,770 glyphs—minimal coverage

Demo…

- InDesign + OpenType Pan-CJK font prototype
  - Use of paragraph tags
  - Use of character tags

Future Predictions

- Today, Pan-CJK fonts clearly require multiple glyphs for many code points
  - It is difficult to argue this point due to locale-specific conventions
- In the future, cross-cultural unification efforts are possible
  - Unicode may serve as the catalyst
  - The Web is making the world smaller, and cross-cultural interaction is ever-increasing
  - This is not likely during the current generation, but perhaps within 25 years

Further Reading & Resources

- *CJKV Information Processing*, Second Edition (O'Reilly Media, 2009)

  http://oreilly.com/catalog/9780596514471/

- OpenType Specification

  http://www.microsoft.com/typography/otspec/

- The Unicode Consortium

  http://www.unicode.org/

Adobe