

Revising the Adobe Type Library: FACELIFT

Thomas Phinney
Product Manager
Fonts & Global Typography

28 April 2007

Agenda

- Why do a big update?
- Things we're NOT fixing
- Fixes we recommend
- Fixes possible for other foundries
- Adobe-specific fixes/changes

Why do a mass update?

- Fix bugs
- Update all fonts to current best practices
- Better consistency across families

Things we're NOT doing

- Try to avoid reflow or big changes to existing docs
- Only change kerning to fix major bugs
- Don't change advance widths
- Not removing PUA

Things we recommend

- sinf/subs duplication
- contextual substitutions using new format
- 'size' in all fonts (to spec design size)
- update version string

'sinf'/'subs' duplication

- MS changed their fonts so 'subs' does what everyone thought it should.
- Our change: for fonts that have 'sinf' duplicate the same stuff in 'subs'

Contextual Substitution Rebuild



'size' feature in all fonts

- Specifies font's design size (ideal size)
- Theory: useful for apps to know when to track font.

Update version string



Things you might want/need

- Update 'frac' code
- Hinting improvements
- Kerning efficiency
- Update notices in the font
- Duplicate small caps
- Eliminate glyph double mapping
- Add stylistic sets
- Add languagesystem coverage
- Language tags fixes
- Style linking/Panose

Update 'frac' code

- More fonts will do arbitrary fractions

Hinting improvements

- All glyphs rehinted with latest autohinter from FDK
 - Requires good font-level hints
- Bugs fixed, notably some fonts with incorrect BlueScale or FamilyBlues/FamilyOtherBlues arrays
- Also correcting any path direction/construction issues

Kerning efficiency

- Remove kern pairs with values of zero
- Remove kern pairs between glyphs of different scripts
- Only affects compiled font if a class becomes entirely unused on one side
 - entire row or column of class kerning table eliminated

Update TM/patent/copyright



Duplicate small caps, ligatures

- A.sc for 'c2sc' and a.sc for 'smcp'
- dotless i-sc

Eliminate glyph double-mapping

- For each glyph, only one Unicode
- Reduces problems in mapping from glyph names to Unicode
 - space
 - periodcentered
 - hyphen
 - fraction
 - macron
 - Tcommaaccent
 - tcommaaccent

Added stylistic sets

- Shelley
 - 2 sets of alt caps
- Bulmer
 - K, R
- Kigali & Sassafras
 - 3 different ascender/descender lengths

Add languagesystem coverage

- DFLT script and dflt language
- Should make features work without specifying all the scripts and languages
- May not be widely supported by apps/OSes yet

Language tags fixes

- Change TUR > TRK
- Add AZE and CRT to TRK
- Cleaner code for handling ligatures vs TRK/AZE/CRT
 - use 'locl' to handle things upstream rather than separately in other features
 - see my blog for details

Style linking / PANOSE

- Adjusting a few OS/2.usWeightClass values to work better with Windows
- Obvious mismatches between OS/2.usWeightClass and PANOSE
- Some fonts did not have PANOSE numbers or had partial numbers

Probably Adobe-specific changes

- fsType update
- glyph name corrections
- bold bit “on” incorrectly
- Incorrect ‘size’ feature construction
- Kerning fixes

fsType update

- 12 > 8 per OT 1.4 change
- Correcting a couple of fonts to allow “editable” instead of “preview & print”

Glyph name corrections

- Won't change anything in WinANSI/MacRoman
- Changes to make relationship of glyph to codepoint more clear, with variants
 - Asmall > A.sc

Bold bit “on” incorrectly

- Only “on” when font is style-linked bold of some lighter weight
- Not the same as ForceBold in T1/CFE
 - Hardly anything uses ForceBold any more

Incorrect 'size' construction

- makeOTF change
- incorrect definitions of ranges
 - not contiguous

Kerning fixes

- Fixing outright bugs where glyphs were in wrong classes (f-ligs, ae, oe, dotlessi)

Resources

- <http://blogs.adobe.com/typblography>
- Typophile “build” forum



Revolutionizing
how the world engages
with ideas and information

