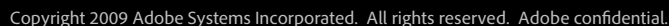# Glyph Sets
for Layout Features

David Lemon

Senior Manager,
Type Development
Adobe Systems, Inc.

19 November 2009

Why set guidelines for glyph sets in layout features?

- Character sets can be grouped in two categories:

- Arbitrary character sets

  - Generally extensions to a font's functionality

- Logically-defined character sets

  - Natural result of defining a font's functional requirements

Examples of arbitrary character sets

- A modern font for the European market must include €, £ – and probably $

- But it may or may not include non-European currency characters like ¥, Pts, or ₡

- …or older characters like ₣, £, or ₠

Examples of logical character sets

- Many languages do not use every letter of the Latin alphabet

  - Hawaiʻian: a b c d e f g h i j k l m n o p q r s t u v w x y z

  - But even Hawaiʻians need to set words in other languages, so they include the full alphabet

- It makes no sense to include only some of the characters needed for a language the font should support

  - If the font supports Icelandic, it needs both ð and þ

- Some of this approaches the realm of glyph sets

  - To support lowercase Greek, you need both forms of sigma: σ and ς

Glyph sets behave like character sets in this respect

- Layout features use alternate glyphs

  - And glyphs can be grouped according to the features that use them

- Some layout features are inherently arbitrary

  - Their alternate glyphs tend to be expressions of imagination and variety

- Other layout features have a natural relationship to specific characters

  - The set of characters depends on the font's design, and language support

  - But always reflects a clear & functional use

Layout features with arbitrary glyph sets

- Any situation where the lack of one or more alternates doesn't create a visual problem

    - Swash, discretionary ligatures, beginning/ending forms (in Western fonts), stylistic alternates

- No standard user expectation

    - Purely the designer's judgement

Layout features with logical glyph sets

- These features operate on functional sets of characters, so users expect complete coverage

  - Figure styles must cover at least 0-9

  - Small capitals should match the font's set of capitals

0123456789 0123456789 0123456789

Borðum því sem Morgunbladið

Alexander is spelled
"Αλέξανδρος" in Greek

Adobe

It all seems so obvious—but there's a problem

- Many fonts break the implicit agreement for logical glyph sets

  - A few support Greek and Cyrillic, but have small capitals only for Latin

  - Includes Adobe's Minion Pro, which we'll be fixing…

- The main problem area is with features like superiors

  - The old Latin-1 set had only ^{1} ^{2} ^{3} as superior figures

  - "Expert" fonts had only the "French" alphabetic superiors $^{a\,b\,d\,e\,l\,i\,m\,n\,o\,r\,s\,t}$

Fonts aren't used only by typographers: case 1

- Users get confused and upset when fonts don't behave as expected.

  - This should be a concern for font developers

  - It is a concern for layout applications

- Many publishing formats (e.g. academic papers) require things like small caps and superiors, but aren't set in InDesign

  - The app's have met this demand for many years by scaling glyphs

  - Now they could use properly designed glyphs, but the fonts' layout features may not provide the full set needed

 - This makes the app's hesitant to add support for OpenType layout features

Fonts aren't used only by typographers: case 2

- A significant amount of text today is set algorithmically

  - Automated layout is now the norm for many environments

  - There is no way a human can check the results before they're published

  - If the layout engines can't have confidence the fonts will provide what's needed, their only choice is to keep the typography crude & basic

- Some automated-layout environments include:

  - Text-oriented applications (e.g. news, eBooks) on smart phones

  - Dedicated eBook devices

  - Specialized news readers

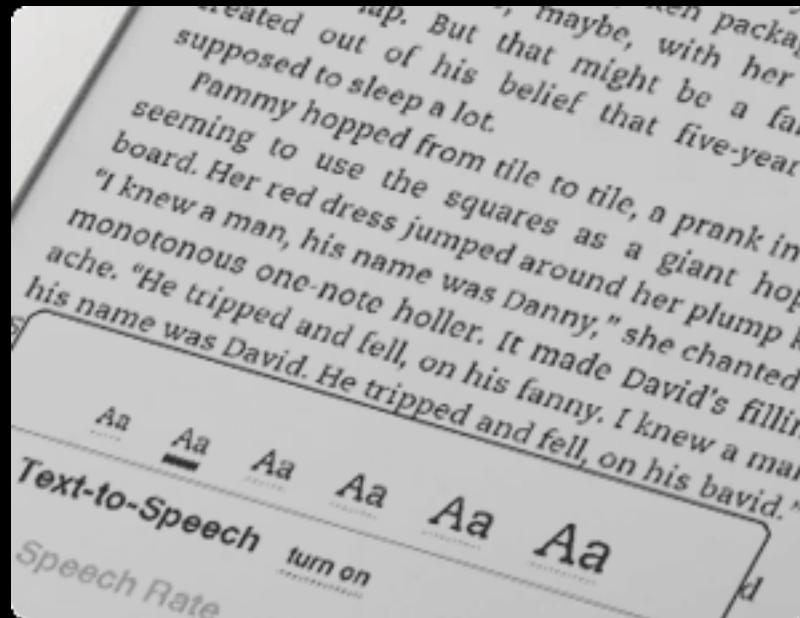 - Statements (e.g. phone bill, bank account history, college transcripts, …)

Adobe

Smart-phone text

Smart-phone text
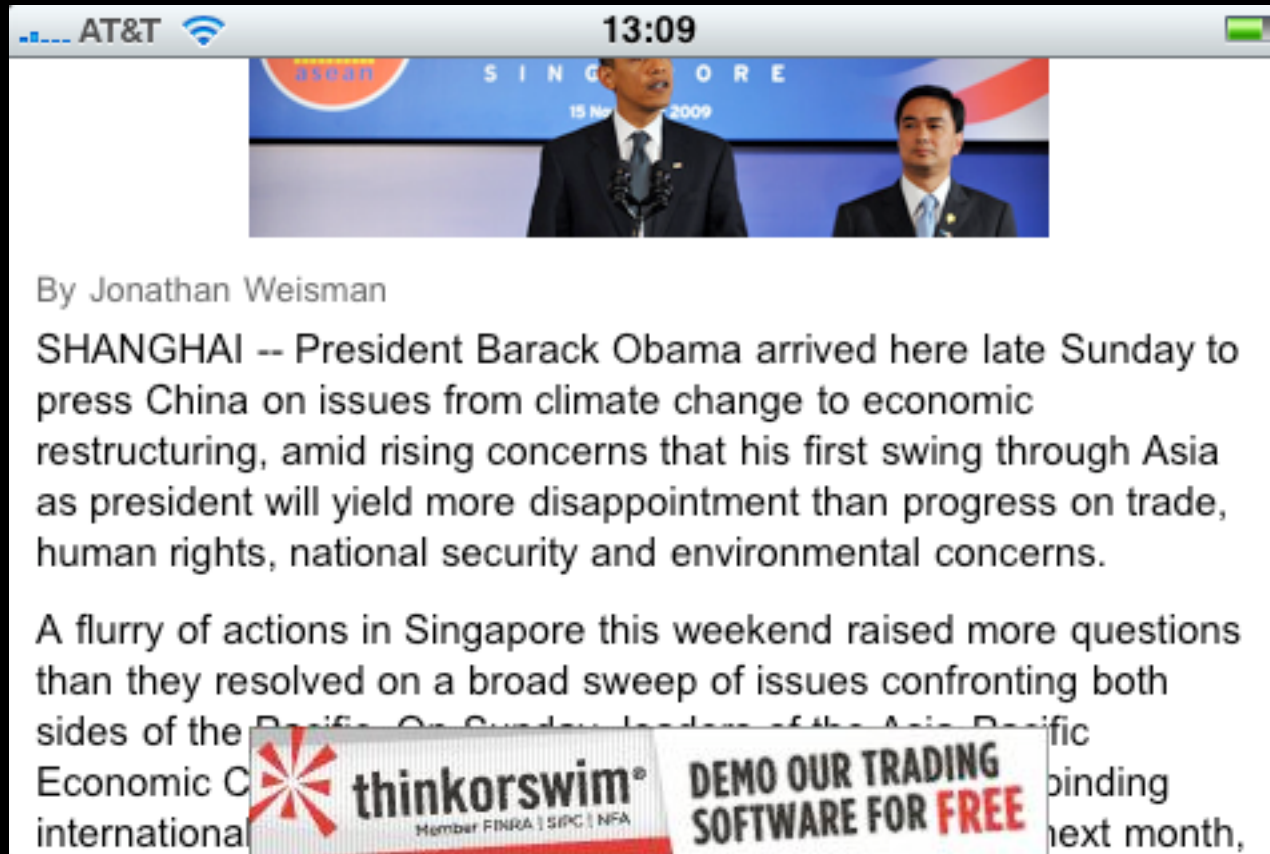
eBook devices

News readers (demo)

Three-part solution

- ## Clarify the layout feature descriptions in the OpenType spec

  - Identify the features that operate on logical (non-arbitrary) character sets

  - Specify the set of alternate glyphs required for basic coverage

- ## Encourage font developers to build compliant fonts

  - Extend old fonts to fix any problems

  - Don't release new fonts unless they work properly

- ## Encourage more applications to use the layout features

  - App's can be more confident that fonts will work correctly

  - Clarified spec means problems with incomplete glyph sets are a font issue, not an application issue

Three-part solution

- Example of guidance to add to 'c2sc' description

  - If this feature is implemented in a font, it should provide small-cap alternate forms for all alphabetic capital characters present in the font, regardless of script. If the font contains figures and currency symbols designed for use with the small capitals, these should also be included in the feature. Small-cap forms of full-height punctuation characters (e.g. exclam, question, braces, brackets and parentheses, and the ampersand) are also recommended in this feature, as it will be used for settings in which all the text is set in small capitals.

We can help improve typography

- **More reliable layout features**

  - Broader use, with fewer problems

- **Better layout by default**

  - Improve the quality of typesetting all around us

- **Happier users**

  - A vital requirement for the font market

- **Questions?**

  - lemon@adobe.com