The PDF Font Aquarium

Thomas Merz

PDFlib GmbH
München, Germany
www.pdflib.com

Bethesda (MD), June 2003

Timetable 1: prehistoric Era

- 1985: Adobe publishes the PostScript and Type 3 specification
- 1986: Adobe starts distribution of Type 1 fonts (proprietary format)
- 1987: Apples starts work on the TrueType font format
- 1990: Adobe publishes ATM for Mac and the Type 1 font format specification
- 1991: Unicode 1.0
- 1991: Apple supports TrueType fonts in System 6
- 1992: Adobe ships Type 1 Multiple Master (MM) fonts
- 1992: Microsoft supports TrueType fonts in Windows 3.1
- 1993: Adobe publishes the Type 42 font format for wrapping TrueType font data in PostScript
- 1993: Windows NT 3.5 converts Type 1 fonts to TrueType (sort of...)

Timetable 2: modern Times

- 1993: Acrobat 1.0 (PDF 1.0) supports Type 1, Type 3, and TrueType fonts
- 1995: Adobe ships PostScript CID fonts and CMaps for Asian text
- 1996: Adobe and Microsoft jointly announce the OpenType font format
- 1996: Acrobat 3 (PDF 1.2) improves Unicode support with ToUnicode CMap
- 2000: Windows 2000 supports OpenType and Type 1 fonts (but not MM!)
- 2001: Mac OS X supports OpenType and Type 1 (but not MM!)
- 2001: Acrobat 5 (PDF 1.4): Tagged PDFs are fully Unicode-compliant
- 2003: Adobe and other font foundries ship thousands of OpenType fonts
- 2003: Unicode 4.0

PostScript Type 1 Fonts

- Developed by Adobe and part of all PostScript versions
- Glyphs are identified by name; glyph names are arranged into encoding vector
- 8-bit addressing: a maximum of 256 glyphs can be used at a time
- Fonts may contain an arbitrary number of glyphs
- Historic relicts in the font and file format:
  - multiple layers of encryption in order to obscure the inner workings
  - ASCII wrapper for the actual font data
  - syntactic restrictions because of PostScript and ATM requirements
- Deploying Type 1 Fonts
  - Adobe Type Manager ATM
  - Windows 2000/XP, Mac OS X
  - Acrobat 1 (PDF 1.0) and above
- Type 1 fonts have a very high reputation, and are the publishing workhorse still today

TrueType Fonts

- Developed by Microsoft and Apple
- Thousands of glyphs can be used at a time
- Metrics information contained in the font file (no additional metrics file)
- Deploying TrueType Fonts:
  - Windows 3.1 and above
  - Mac OS System 6 and above
  - Acrobat 1 (PDF 1.0) and above
- TrueType fonts had a touch of low-quality and office use, but are increasingly used in publishing, too

OpenType Fonts

- Finally the holy grail of font formats...
  - unites TrueType and PostScript outline descriptions
  - unites Mac and Windows file formats
  - unites font outline and metrics data
- OpenType is based on the TrueType file format (*.ttf), but adds:
  - support for Type 1 outlines (*.otf)
  - extended typographic features (e.g. ligatures and small caps)
- Deploying OpenType Fonts (full support; ttf flavor supported earlier):
  - Windows 2000 and above
  - Mac OS X and above
  - Acrobat 3 (PDF 1.2) and above
  - Adobe Applications really take advantage of the new font features
- OpenType will be the font format of the next decades:
  - supported by major vendors and systems
  - full Unicode support
  - extended typographic features

CID Fonts

- CID fonts are initially targeted at Chinese, Japanese, and Korean (CJK) text
  - CJK languages contain thousands of characters
  - character collections summarize all required characters for a region
  - CID (Character Identifier) are used to address these characters
  - CMap (character maps) defines the mapping of CID for various encodings
- CMaps provide for highly flexible mapping of characters
- CID fonts are no longer restricted to CJK text
  - fully supported in Acrobat 3 (PDF 1.2) and other applications
  - modern PDF creators generate PDF output with CID fonts
- CID fonts in PDF are usually the result of converted fonts:
  - CIDFontTypeo (»Type 1 CID«): result of CID PostScript fonts and OpenType fonts with PostScript outlines
  - CIDFontType2 (»TrueType CID«): result of TrueType fonts and OpenType fonts with TrueType outlines

Font Embedding

- Embedding includes the font data in the PDF file for viewing and printing
- All font formats can be embedded in PDF, but some are converted
- Embedding increases the PDF file size
- TrueType, OpenType, and CID fonts may contain a flag which controls embedding
  - some versions of Fontographer generate undefined embedding flag
  - many fonts specify embedding restrictions although the designer didn't mean it
  - handling of embedding flag relaxed from Distiller 4.0 to 4.05
- Embedding ensures proper text display, and should be avoided only for good reasons (trading quality for small file size)

Font Embedding

- Embedding depends on various factors:
  - Distiller settings controlled by the user
  - embedding flag in the font
  - character set and encoding
- Distiller ignores embedding settings in several situations where successful font substitution cannot be guaranteed:
  - Type 1 Symbol fonts
  - Type 1 fonts with less than 115 or more than 229 characters
  - Type 3 fonts will always be embedded
  - CID fonts with an unknown character collections
  - TrueType and OpenType fonts with certain encoding schemes

Font Subsetting

- Subsetting embeds only those characters which are actually used in a document; advantages:
  - reduce file size
  - make it harder to pirate the font
- Subset fonts have a prefix of six random characters in their name
- Subsetting of Latin Type 1 fonts can be controlled via the Distiller UI
- Distiller ignores subsetting controls, and always embeds a subset in various situations:
  - Non-Latin Type 1 fonts
  - TrueType and OpenType fonts
- Subsetting can happen earlier in the process, e.g. in the printer driver

Font Substitution

- Acrobat tries to substitute required fonts which are not embedded
  - Font descriptor describes the missing font
  - Latin characters: Multiple Master fonts AdobeSansMM and AdobeSerifMM simulate missing fonts
  - Acrobat 3.0.1 and above: CJK text with standard character collections (using the standard CJK fonts from the Asian font pack)
- Compare original and substitute fonts via View, Use Local Fonts
- The following can not be displayed unless they are installed on the system:
  - symbol (pi) fonts
  - text fonts with characters outside the Adobe standard Latin character set (e.g. some Polish characters)
  - CJK fonts with non-standard character collections (very rare, though)
- Problems:
  - limited character set in substitution fonts: some characters work, some don't
  - some text fonts are encoded as symbol fonts, and cannot be substituted

PDF Core Fonts (Base 14)

- Core fonts as defined in the PDF reference are guaranteed to be always present, and need not be embedded in the PDF file:
  - Courier, Courier-Bold, Courier-Oblique, Courier-BoldOblique
  - Helvetica, Helvetica-Bold, Helvetica-Oblique, Helvetica-BoldOblique
  - Times-Roman, Times-Bold, Times-Italic, Times-BoldItalic
  - Symbol
  - ZapfDingbats
- The actual appearance may vary from one instance of Helvetica to another
- In Acrobat 4 and above core fonts may be embedded in PDF, and are allowed to override the standard core font metrics (important for prepress)

The Core Font Mess

- Alternate names are allowed, and occur (excerpt for Helvetica-Oblique):
  - Helvetica-Italic = Helvetica,Italic = Arial-Italic = Arial,Italic, Arial-ItalicMT
  - Acrobat maps these names accordingly
  - PDFWriter is notorious for creating PDF with alternate core font names
- Acrobat versions ship with different core font sets and map these:

1, 2, 3	*Courier*	*Helvetica*	*Times-Roman*	*Symbol*	*ZapfDingbats*
4, 5	*Courier*	*ArialMT*	*TimesNew-RomanPSMT*	*Symbol*	*ZapfDingbats*
6	*CourierStd*	–	–	–	*AdobePiStd*

- Implications of the modified Acrobat 6 core font set
  - Helvetica and Times are no longer available in the Acrobat installation
  - text display in these fonts is subject to font availability on the system
  - should better embed these in order to avoid problems
  - CourierStd contains 374 glyphs, many more than in previous versions

Acrobat does a good Text Extraction Job

- Takes into account all available information
  - encodings and CMaps
  - Unicode mapping tables
  - decomposes ligatures into individual characters
  - OpenType font information about stylistic glyph variants (GPOS table inversion)
- Acrobat is pretty smart when it comes to text extraction, but it may fail for certain PDFs

Q & A