The TeX Font Panel

Nelson H. F. Beebe (chair)
Center for Scientific Computing
University of Utah
Department of Mathematics, 322 INSCC
155 S 1400 E RM 233
Salt Lake City, UT 84112-0090
USA
Email: beebe@math.utah.edu, beebe@acm.org, beebe@computer.org, beebe@ieee.org (Internet)
WWW URL: http://www.math.utah.edu/~beebe
Telephone: +1 801 581 5254
FAX: +1 801 585 1640, +1 801 581 4148
Internet: beebe@math.utah.edu

Introduction

The TUG'2001 Font Panel convened on Thursday, August 16, 2001, with members William Adams, Nelson H. F. Beebe (chair), Barbara Beeton, Hans Hagen, Alan Hoenig, and Ross Moore, with active participation by several attendees in the audience. The list of topics that was projected on the screen makes up the sectional headings in what follows, and the topics are largely independent.

Any errors or omissions in this article are solely the fault of the panel chair.

Unicode

The work of the Unicode Consortium, begun in 1988, and first reported on for the TeX community in a *TUGboat* article [9], has reached version 3.0 of the Unicode Standard [29]. Version 3.1 appeared about the time of the TUG'2001 conference, and version 3.1.1 shortly thereafter. Unicode is a proper subset of the ISO/IEC 10646 Universal Character Set Standard [14], but publication of the latter lags.

Unicode defines a character set that is intended ultimately to cover all of the world's writing systems. Its first 128 entries are identical to the ASCII character set (dating from 1964) used by most of the world's computers.

There is a very active Unicode technical discussion e-mail list: send subscription requests to unicode-request@unicode.org. The list is archived at http://www.unicode.org/mail-arch/.

Unicode conferences are held twice a year, with the twentieth in late January 2002; see http://www.math.utah.edu/pub/tex/bib/index-table-u.html#unicode for a bibliography of publications about Unicode.

Since most programming languages, operating systems, file systems, and even computer I/O and CPU chips, have character knowledge designed into them, changing the character set has huge ramifications for the computing industry and for worldwide business data processing, data exchange, and record keeping.

Fortunately, a particular encoding scheme called UTF-8 makes it possible for files encoded in pure ASCII to also be Unicode in UTF-8 encoding, easing the transition to the new character set.

Up to version 2.0 in 1996, the Unicode character repertoire could be fit into a table of $2^{16} = 65\,536$ entries. Version 3.0 in 2000 increased the count to over a million, although just under 50 000 are assigned and tabulated in the book. Version 3.2 in 2002 has just over 95 000 assigned. Consortium members hold the view that 20 or 21 bits per character (just over two million) may ultimately be necessary by the time all historical scripts have been covered.

Despite the Consortium's warning that the collection was expected to grow, several vendors did not pay attention, and prematurely adopted 16-bit entities to hold Unicode characters.

Thus, the C language data type, wchar_t, introduced in 1989 Standard C [7, 13, 28], is implemented as a 16-bit unsigned integer in many C and C++ compilers, with a companion function library that also has this limitation.

Even worse, the popular Java programming language is defined in terms of an underlying virtual machine [23, 24], already implemented in hardware, whose instructions are permanently designed for 16-bit characters.

These 16-bit limitations can be overcome by representation of Unicode values with variable numbers of bytes, as was done with the UTF-8 encoding. Unfortunately, the opportunity to simplify character processing significantly by having fixed-size units is tragically lost.

In the panel chair's view, these design errors will rank with the infamous ASCII/EBCDIC split in 1964, with IBM System/360 adopting EBCDIC, and everyone else (by about 1980) adopting ASCII, with enormous economic costs, and user confusion, that lasted for decades.

Newer operating systems are already designed to use Unicode as the native character set, and vendors of older ones are migrating in that direction through UTF-8 encoding.

Of course, jumping from a 256-character set to one with potentially millions of characters poses an almost impossible problem for font vendors. It will be a very long time before the Unicode font repertoire is adequate. Current systems with native Unicode support generally provide only a subset of characters, and then sometimes only in low-resolution screen bitmaps. Bitstream for a while offered their Cyberbit Unicode font, but in July 2001, withdrew it without explanation.

Thanks to fine work by fellow TUG members Yannis Haralambous and John Plaice [26], TEX has been extended to fully support Unicode. Their system is known as Ω (Omega), and it has been available on the annual TEX Live CD-ROM distributions since at least version 5 in 2000. Development has not been as rapid as end users might like, but it must be understood that this is a hugely complex problem, and the Ω designers have been proceeding very carefully, cognizant of other TEX developments such as PDFTEX, ε-TEX, and $\mathcal{NTS}$, in addition to the evolution of the Unicode Standards.

Mathematics fonts

Fonts for mathematics are a substantial problem, because, among the more than twenty thousand fonts on the market, only a handful have a remotely adequate repertoire of mathematical glyphs. These fonts are almost the only choices: *Computer Concrete*, *Computer Modern*, *Informal Math*, *Lucida*, *MathTime*, *PA Math*, *PX*, *Palatino Math*, *Pandora*, and *TX*.

While it is, of course, possible to use an existing mathematics font with any other text font, the results are rarely visually successful. For some careful studies of this, see Hoenig's book [11, Chapter 10].

Font subsetting

DVI drivers for virtually all devices, other than PostScript, subset the fonts that they include in their output streams: descriptions of unused characters are simply omitted.

Doing this for PostScript TYPE 1 outline fonts has proved considerably more troublesome. These fonts are generally encrypted, but Adobe has published the encryption algorithm and keys, so software like T1DISASM (from Lee Hetherington's and Eddie Kohler's T1UTILS package, available at `ftp://ctan.tug.org/tex-archive/fonts/utilities/t1utils`) can readily disassemble a font.

Disassembly reveals essentially a table of *numbered* (not named) subroutines, SUBRS, each containing positioning commands, and calls to other subroutines, plus a table of character definitions, CHARSTRINGS, indexed by character name. Each entry of CHARSTRINGS also consists of positioning and drawing commands, and calls to the numbered subroutines.

Because subroutine numbers could be constructed dynamically, it is in general not possible to identify which of the numbered subroutines can be omitted, but a DVI driver could drop unused entries from the CHARSTRINGS table. This is transparent to font rendering software, since the entries are named, rather than numbered.

It was reported by a reviewer that computation of subroutine numbers is in practice not done in existing TYPE 1 and TYPE 2 COMPACT FONT FORMAT (CFF) fonts, so perhaps it is safe to drop subroutines that are not explicitly called.

Recent versions of Tom Rokicki's DVIPS driver are capable of subsetting PostScript TYPE 1 outline fonts, as can Adobe Acrobat DISTILLER and GHOSTSCRIPT's PS2PDF.

However, this subsetting introduces new problems. What if the DVI file also included PostScript figures which themselves used fonts? Subsetting might remove characters needed by those figures.

It is infeasible, or unreliable, for the DVI driver to attempt to examine an included figure file to determine its font requirements, because far too many PostScript producing programs fail to conform to Adobe's Document Structuring Conventions that would otherwise clearly, and simply, record the file's font needs. Those conventions are clearly described in the first two editions of the PostScript Language Reference Manual [1, Appendix C] [3, Appendix G], but were ominously dropped from the third edition [6]. They are, however, documented at the Adobe Web site among the technical notes collected at `http://partners.adobe.com/asn/developer/technotes/postscript.html`, in the file `http://partners.adobe.com/asn/developer/pdfs/tn/5001.DSC_Spec.pdf`.

Each TYPE 1 font contains a special 24-bit (0 . . . 16 777 215) unsigned number, the UNIQUEID, which is intended to allow printing devices to cache bitmaps of rendered fonts between jobs. A million of these numbers are reserved for private use, and the rest are allocated to font vendors on request. A subsetted font is a *different* font, because it lacks some characters, and so must be assigned a UniqueID from the private use area. A random choice from this area would mean a one-in-a-million chance of confusion between fonts in a printer.

Regrettably, several versions of Adobe's own Acrobat Distiller, and most versions of ghostscript (until the panel chair, who is a long-time ghostscript beta tester, reported the problem) use a fixed UniqueID and fixed name for subsetted fonts! Happily, versions of ghostscript released in 2001 no longer have this problem.

This non-UniqueID and fixed fontname problem fouls up more than just printers. It has been a huge headache in the U.S. National Science Foundation FastLane grant proposal project, started about 1998 to speed up, and regularize, proposal submission.

FastLane is a clear case of technology being adopted before its time. Had NSF required submission of a single PDF file for the entire proposal, or not refused to accept documents without font subsetting, the UniqueID and fontname issue would never have been noticed.

Unfortunately, they instead require submission of multiple PDF files, with subsetted and embedded fonts. These files are then merged into one PDF file for the entire proposal, then distributed back to the submitter for printing and verification, and sent electronically to proposal reviewers.

Because of the non-UniqueIDs and fixed fontnames, the software that does the merging gets hopelessly confused, and fails to produce correct output. The proposal submitter is then held to be at fault. The panel chair has spent an inordinate amount of time working with colleagues, and remotely, with NSF staff and administrators, to overcome these problems. After three years of user complaints, NSF has finally relaxed their draconian requirements, and now accepts DVI and PostScript files as well.

Font substitution

A common problem when documents are distributed is that the required fonts may be missing at the end-user site, and because of font licensing, it may not be possible to include the fonts with the documents.

World-Wide Web browsers usually just ignore requests for missing fonts, falling back to a default font. Adobe Acrobat Reader goes further: it uses the original font metrics embedded in the PDF file, and then substitutes the missing font with another. TEX DVI drivers usually complain about missing fonts, but some will then provide a substitute, and some may even support a user-defined font substitution file.

The PANOSE system [8] is a font classification system that assigns numeric values in 0...15 for ten font attributes (family, serif style, weight, proportion, ...). The system is further described at `http://www.w3.org/Fonts/Panose/pan2.html`, `http://www.w3.org/Printing/stevahn.html`, and `http://www.agfamonotype.com/print_manu/pan1.htm`

Hewlett-Packard now owns the PANOSE technology, but has proposed it for open international standardization. They recommend that font files be augmented with a PANOSE number that can be used by matching software to find the closest available font. In practice, this classification seems not to have been done: in a scan of more than 8700 font files on my system, I found only two that had an embedded PANOSE number.

Nevertheless, since a lot of work has been done in the PANOSE system to identify characteristic properties of fonts, those features should be carefully considered when new font software is written.

Font licensing

Noted font designer Chuck Bigelow thoroughly discussed the issue of typeface copyright issues in an article reprinted in *TUGboat* several years ago [10].

With widespread sharing of PostScript and PDF files on the Internet, it is difficult for font vendors to enforce their licenses. Some vendors, including Adobe, take the view that a subsetted font is sufficiently crippled that it is unlikely to be of interest to font miners, and explicitly permit distribution of documents containing subsetted fonts, while forbidding distribution without subsetting. Others are more restrictive: Bitstream does not permit free distribution of their font metric files, and historically, Autologic would not even reveal font metrics to its own font licensees!

Users can always achieve font subsetting for PostScript files from any source by the simple conversion path PostScript → PDF → PostScript, *provided* that Distiller or ps2pdf options have been chosen to turn on subsetting. Regrettably, few Web site owners have the sophistication, or conscience, to do this, and the conversion programs do not subset fonts by default. Versions of ps2pdf after May 2000 subset by default.

More troublesome is the issue of patents on algorithms and file formats, and the U.S. Patent

Office, in particular, continues to issue software patents on ideas that are utterly obvious, even to people who have never used a computer. The X Window System backing store patent held by AT&T essentially says "if you cannot store data here, put it there instead, but only if you license the right to do so from us first". The European Union, followed by the U.S. Government, has retroactively extended patent and copyright lifetimes, exacerbating the problem.

Adobe has always been very open about publishing specifications of PostScript, PDF, and font formats, allowing anyone to implement them. They copyright their tradenames and license their software (see `http://partners.adobe.com/asn/developer/legalnotices.html`), and by virtue of being both specifiers and implementors, have a head start on their competition. However, they do not interfere with, or discourage, competitors. As a result, several printer vendors have shipped models with non-Adobe PostScript interpreters, and Aladdin and GNU GHOSTSCRIPT have helped to make PostScript and PDF support almost universally accessible.

TRUETYPE, and possibly OPENTYPE, are covered by Apple patents that restrict what rendering software can do to display the fonts. This is highly unfortunate, and some people may react by refusing to use such fonts. Their widespread use on common desktop operating systems makes it difficult to avoid them, however. For further discussion, see the FreeType Web site, `http://www.freetype.org/patents.html`.

Font index

Tens of thousands of fonts are available commercially, and hundreds are typically installed on each desktop computer. Sadly, it sometimes takes a expert to hunt down a particular font to make it usable in a particular application.

Font files often have filenames that are not obviously related to font names. For example, BaskervilleMT-BoldItalic might be found in files with names containing `mbvbi8a` or `basbi___`. Designers of deficient file systems with drastic filename length limitations are partly responsible, but some platforms complicate things by concealing fonts in 'resources' or 'registries', or by converting standard font formats to proprietary internal ones (without offering the reverse conversion).

The problem of mapping font names to filenames has led software designers to invent several different, and mutually incompatible, map file formats, adding to system administrator burdens. Later on the same day that this was written, traffic on the GHOSTSCRIPT developers list discussed precisely this problem, with respect to new Apple operating systems; it evidently remains an inadequately solved problem!

Font licensing means that few font files are visible to Internet Web search engines, so your hunt for VanDijkBoldPlain might be almost fruitless.[1] While this sentence was being written, a search on `http://www.dogpile.com/`, a site that searches several other search engines, found only one hit from twelve of them. That hit was to the font index of more than 20 000 font names created by, and maintained by, the panel chair at `http://www.math.utah.edu/~beebe/fonts`.

That font index is vendor neutral: each listed font has a vendor name, linked to a vendor page at the index site that gives company name, address, other contact information, and links to the vendor Web site.

The index is maintained as rigorously validated and prettyprinted HTML files, all in a standard format, and all derived from tables, catalogs, CD-ROM contents, and other resources. The index maintainer has no commercial interest in the font industry, and the index may be mirrored by anyone.

Multiple Master fonts

In 1992, Adobe introduced the concept of 'Multiple Master fonts' [4, 5]. These are fonts with one to four user-adjustable parameters that can be tweaked to change character shapes.

Knuth's METAFONT system is much more general: the Computer Modern fonts have 62 parameters [20, pp. 10–11] whose variation can lead from fixed-width `typewriter` fonts to proportionally-spaced **bold**, *italic*, roman, sans serif, *slanted*, SMALL CAPS, ... The design of such 'meta' fonts is truly a landmark of human ingenuity.

Sadly, it appears that Adobe has withdrawn support for Multiple Master fonts: their documentation has been removed (fortunately, the panel chair has an archived copy), and the fonts may no longer be available, although several are still listed at `http://www.adobe.com/type/browser/C/C_4e.html`. It has also been reported that the metric files for these fonts were not included in font packages sold by Adobe, and are hard to find.

In the hands of expert programmers, the capabilities of Multiple Master fonts could have been made easily available to the masses, but except for one experiment with ILLUSTRATOR [12], the opportunity was never taken advantage of, and has now

[1] Curiously, if you search for the logical conjunction of the four parts separately, you may be luckier!

been removed. Such technological shortsightedness, probably influenced by quarterly bean-counting reports, is deplorable.

Perhaps the OpenType effort (see `http://www.opentype.org/`) will revive this possibility. In the meantime, some clever programmers could do wonderful things for the font world by creating a fancy interface to METAFONT!

The t1utils package mentioned above can handle Multiple Master fonts, and according to `http://developer.apple.com/technotes/tn/tn2029.html`, Apple MacOS X has added new printer support for such fonts.

Eddie Kohler's mminstance package, available at `http://www.lcdf.org/type/`, includes mmafm, for creating an Adobe Font Metric (AFM) file, and mmpfb, for creating a normal Type 1 font file, from a particular instance of a Multiple Master font. These tools have been ported to Microsoft Windows, and are expected to be in the next edition of the TEX Live CD-ROM.

Font file handling problems

The format of Type 1 font files is described in a small book [2]. Unfortunately, software implementors have not always followed that specification.

Adobe's own Type Manager (ATM), which renders Type 1 fonts for screen display on single-user desktop operating systems, is known to be sensitive to the precise formatting of font files, and fails on some third-party fonts that otherwise conform to the specification, and are handled correctly by PostScript interpreters in printers. Adobe Illustrator may exhibit similar problems.

Recent Adobe products, such as Illustrator 7.0 and later, use a new font-rendering engine, called CoolType, instead of Adobe Type Manager. CoolType tries to reduce font menu clutter, and ends up treating cmr10 and cmr12 as the same font. It also has trouble with subsetted fonts with encoding arrays.

The problems with ATM and CoolType are clear cases where Adobe software developers are not adhering to their own published font specifications, and users should complain bitterly. Fortunately, clever people, notably Tom Kacvinski, in the TEX community were able to react quickly and find workarounds.

Some desktop software products fail to handle Type 1 fonts that contain characters in slots 0 . . . 31 or 128 . . . 159. The lower 32 slots are vacant in all of Adobe's standard encodings defined in Appendix E of the PostScript Language Reference Manuals [3, 6]; the other 32 are partly used in advanced font encodings, but not in the default StandardEncoding used by most Type 1 fonts. This problem affects most TEX fonts, which tend to contain either 128 or 256 characters.

Type 1 fonts are based on a named collection of characters, and those names are then assembled into a 256-element 'encoding vector' to provide character access using 8-bit bytes as indexes into the encoding vector. Many fonts have more than 256 characters, as shown in Figure 1, although 8-bit bytes in character strings limit access to 256 in any one instantiation of the font.

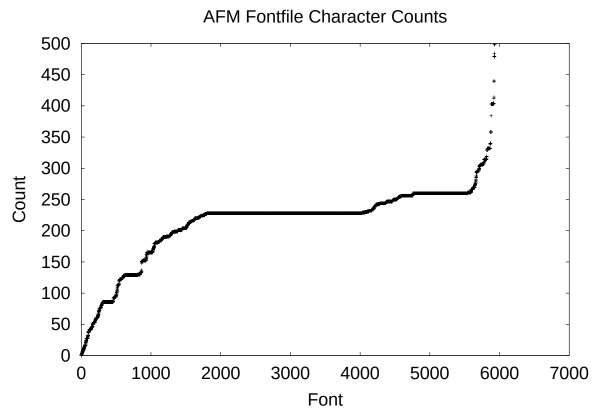

Figure 1: Character counts in 6021 Type 1 fonts, ordered by increasing counts. The largest font in this collection contained 26,304 Chinese characters. A total of 1269 fonts (21%), have more than 256 characters, but only 39 (0.6%) have more than 1000.

Recently, we have encountered Type 1 fonts designed by people who have completely failed to understand the importance of encoding vectors and named characters. These fonts have an encoding vector with the Adobe StandardEncoding names, but the characters with those names have shapes that bear no relation to their names! This makes constructing an encoding vector to use them exceedingly painful and error prone.

The Apple MacOS operating system does not directly use fonts in Type 1 format; instead, it converts them to a proprietary internal format. While this conversion is quite well debugged, it is another possible point of failure. It means that software ported from other platforms must be rewritten to use the MacOS font format. It may also be nontrivial to export fonts from MacOS to other operating systems. Observation of MacOS since its introduction in 1984 clearly shows that its design discourages software developers from porting to it. This may change rapidly once the new UNIX-based MacOS X becomes widely deployed.

GNU Project and Aladdin font-related activities

Work on the GNU font utilities (ftp://ftp.gnu.org/pub/gnu/fontutils) ceased from 1993 to 1998, but a lot of changes were logged for 2000. Much more could be done; can you volunteer?

The already-noted FreeType Project is not part of the GNU Project, but some of the FreeType people involved work in both, and the goals are much the same: freely-distributable software of wide utility.

Although only delayed releases of Aladdin Software's GHOSTSCRIPT fall into the GNU archive, it is worth noting here that in the last year, GHOSTSCRIPT development has moved to the bazaar model [27], but an expert chief architect maintains final control over quality, and the developers list is quite active.

Just a month before the TUG'2001 Conference, the first release of GHOSTPCL was made (see http://www.artifex.com/downloads/). This builds on the GHOSTSCRIPT source tree, but provides instead, for the first time, a portable and distributable interpreter for Hewlett-Packard's Printer Command Language, PCL. Many low-cost desktop printers, and most laser printers, recognize PCL, even if they lack PostScript support.

The lack of PCL screen display, until GHOSTPCL was released, has hindered software development of PCL tools. As GHOSTPCL matures from its current shaky state, PCL may become of more interest in the TeX community, and that is especially important because PCL is available to those on a low budget. Although GHOSTSCRIPT can create page bitmaps for many different non-PostScript printers, the files are large, and slow to print.

Other font developments

Some unattributed Polish programmers produced the TTF2PF package for converting TRUETYPE fonts to TYPE 1 format. It is available at ftp://ftp.gust.org.pl/TeX/GUST/contrib/BachoTeX98/ttf2pf/

The CTAN archives in ftp://ctan.tug.org/tex-archive/fonts/utilities/ collect more than 40 packages for dealing with font conversions.

Oleg Motygin's TTF2MF package, available at ftp://ctan.tug.org/tex-archive/support/ttf2mf, converts TRUETYPE fonts to METAFONT source code. It does so by invoking an operating system function on Microsoft WINDOWS, so it runs only on that platform. Nevertheless, in announcements this year on the TEX-EURO mailing list, Daniel Taupin has demonstrated that TTF2MF can be used quite effectively to convert such fonts, making them accessible to the majority of TeX DVI drivers, which are incapable of handling the TRUETYPE format. There are, of course, thorny license issues that must be dealt with, and TTF2MF does not handle hinting, so the METAFONT programs are not expected to produce good results at low resolution. Despite the lack of hinting, I have found that screen display of these fonts is quite acceptable.

A month after the TUG'2001 conference, Vladimir Volovich announced the release of his CM-Super font package, available in ftp://ctan.tug.org/tex-archive/fonts/ps-type1/cm-super. It contains a large collection (376) Type 1 representations of all the EC, TC, and LH fonts, including Cyrillic. These fonts contain characters needed for a half-dozen common LaTeX font encodings, and for Adobe's STANDARDENCODING. This greatly extends the set of TeX fonts that are available in Type 1 form, making them usable in many other software tools, and importantly, for labelling in illustrations.

The TeX Live 6 CD-ROM contains Scott Pakin's MF2PT1 tool for converting a subset of METAFONT to TYPE 1 format.

The EuroTeX 2001 proceedings are expected to contain an article by Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk about their work on METATYPE1, a METAPOST-based engine for generating TYPE 1 fonts. Their software is available at ftp://bop.eps.gda.pl/pub/metatype1.

My personal hope is that software will eventually be developed, and made freely available across all major platforms, for automatic and reliable conversion between any of the common outline font formats. Not only would this be of considerable convenience for users, but font designers could then take advantage of features offered by a particular format, such as METAFONT's shaped pens.

Further reading and tool pointers

Tracy's book [30] is an interesting history of type design.

Karow's books [15, 16, 17, 18] are among the few devoted entirely to practical aspects of font design, and font file data representation.

Level, Newman and Newman [21, 22] provide a convenient catalog of thousands of fonts, with samples of each.

Moye [25] describes the commercial Fontographer package (http://www.macromedia.com/software/fontographer/), which is one of the more widely used desktop publishing systems for font design, and font manipulation on desktop platforms.

The commercial FontLab system (http://www.fontlab.com/) provides tools for font design and manipulation. It began as a program on the Atari ST, and was later ported to Microsoft WINDOWS and Apple MACOS.

Adobe's TYPE 1 Font Format is described in [2]; that reference contains pointers to an electronic version of the book, and a supplement that covers later developments.

Hoenig's book (in the biased view of the panel chair) is an excellent one for LATEX and TEX users, because it is all about using fonts with TEX.

The online notes for David Kindersley's Workshop [19] make interesting reading about font design and legibility issues.

There are extensive bibliographies on fonts and related issues in the TEX Users Group bibliography archive at http://www.math.utah.edu/pub/tex/bib.

Acknowledgements

Thanks go to my fellow panel members, my wife Margaret, and L. Peter Deutsch of Aladdin Software, for many helpful comments on, and historical background for, drafts of this article.

References

[1] Adobe Systems Incorporated. *PostScript Language Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985. ISBN 0-201-10174-2. ix + 321 pp. LCCN QA76.73.P67 A33 1985.

[2] Adobe Systems Incorporated. *Adobe Type 1 Font Format—Version 1.1*. Addison-Wesley, Reading, MA, USA, August 1990. ISBN 0-201-57044-0. iii + 103 pp. LCCN QA76.73.P67 A36 1990. US$14.95. URL http://partners.adobe.com/asn/developer/pdfs/tn/T1_SPEC.PDF. See also supplement [5].

[3] Adobe Systems Incorporated. *PostScript Language Reference Manual*. Addison-Wesley, Reading, MA, USA, second edition, 1990. ISBN 0-201-18127-4. viii + 764 pp. LCCN QA76.73.P67 P67 1990.

[4] Adobe Systems Incorporated. Adobe Type 1 font format Multiple Master extensions. Technical note 5086, Adobe Systems Incorporated, 1585 Charleston Road, P. O. Box 7900, Mountain View, CA 94039-7900, USA, Tel: (415) 961-4400, February 14, 1992. Withdrawn and no longer available: superceded by [5].

[5] Adobe Systems Incorporated. Type 1 font format supplement. Technical note 5015, Adobe Systems Incorporated, 1585 Charleston Road, P. O. Box 7900, Mountain View, CA 94039-7900, USA, Tel: (415) 961-4400, May 15, 1994. URL http://partners.adobe.com:80/asn/developer/pdfs/tn/5015.Type1_Supp.pdf.

[6] Adobe Systems Incorporated. *PostScript Language Reference*. Addison-Wesley, Reading, MA, USA, third edition, 1999. ISBN 0-201-37922-8. xii + 897 pp. LCCN QA76.73.P67 P67 1999. US$49.95, CDN$74.95. URL http://partners.adobe.com/supportservice/devrelations/PDFS/TN/PLRM.pdf;http://partners.adobe.com/asn/developer/PDFS/TN/PLRM.pdf. This new edition defines PostScript Language Level 3. An electronic version of the book is available at the Adobe Web site, and is also included in a CD-ROM attached to the book.

[7] American National Standards Institute and Computer and Business Equipment Manufacturers Association and Secretariat. *American National Standard for information systems: programming language: C: ANSI X3.159-1989*. American National Standards Institute, 1430 Broadway, New York, NY 10018, USA, December 14, 1989. 219 + vi + 119 pp.

[8] Benjamin Bauermeister. *A manual of comparative typography: the PANOSE system*. Van Nostrand Reinhold, New York, NY, USA, 1988. ISBN 0-442-21187-2. xxvii + 257 pp. LCCN Z250 .B34 1988.

[9] Nelson Beebe. Character set encoding. *TUGboat*, 11(2):171–175, June 1990. ISSN 0896-3207.

[10] Charles Bigelow. Notes on typeface protection. *TUGboat*, 7(3):146–151, October 1986. ISSN 0896-3207.

[11] Alan Hoenig. *TEX Unbound: LATEX and TEX Strategies for Fonts, Graphics, & More*. Oxford University Press, Walton Street, Oxford OX2 6DP, UK, 1998. ISBN 0-19-509686-X (paperback), 0-19-509685-1 (hardcover). ix + 580 pp. LCCN Z253.4.L38H64 1997. US$60.00 (hardcover), US$35.00 (paperback). URL http://www.oup-usa.org/gcdocs/gc_0195096851.html.

[12] Joe Holt. Notes from a slave to Multiple Master fonts. *Adobe Developers Association News*, 3(2), February 1994. URL http://partners.adobe.com/asn/developer/techjournal/archived/Vol3no02.pdf.

[13] International Organization for Standardization. *ISO/IEC 9899:1990: Programming languages*

— *C*. International Organization for Standardization, Geneva, Switzerland, 1990. URL `http://www.iso.ch/cate/d17782.html`.

[14] International Organization for Standardization. *ISO/IEC 10646-1:1993, Information technology. Universal Multiple-Octet Coded Character Set (UCS). Part 1: Architecture and Basic Multilingual Plane*. International Organization for Standardization, Geneva, Switzerland, 1993. 754 pp. CHF 360. URL `http://www.iso.ch/cate/d18741.html`. About two dozen corrections and amendments to this Standard have been published.

[15] Peter Karow. *Digital Formats for Typefaces*. URW Verlag, Hamburg, Germany, 1987. ISBN 3-926515-01-5. 400 pp. LCCN Z253.3 .K371 1987.

[16] Peter Karow. *Digitale Schriften, Darstellung und Formate, Geleitwort von Hermann Zapf (Digital Fonts, Representation and Formats, Forward by Hermann Zapf)*. Springer-Verlag and URW Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc. and Hamburg, Germany, 1992. xiii + 457, with 230 illustrations pp. English translation in [17].

[17] Peter Karow. *Digital typefaces: description and formats*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1994. ISBN 0-387-56509-4. xiii + 448 pp. LCCN Z250.7 .K37 1994. Foreword by Hermann Zapf. Translated by Bill Horton from German edition [16].

[18] Peter Karow. *Font Technology: Description and Tools*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1994. ISBN 0-387-57223-6. xvi + 460 pp. US$79.00.

[19] David Kindersley and Lida Lopes Cardozo. David Kindersley's workshop. World-Wide Web document, 1987. URL `http://www.icce.rug.nl/erikjan/bluefuzz/David-Kindersley/David-Kindersley/David-Kindersley.html`.

[20] Donald E. Knuth. *Computer Modern Typefaces*, volume E of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986. ISBN 0-201-13446-2. xv + 588 pp. LCCN Z250.8.M46 K574 1986.

[21] Jeff Level, Bruce Newman, and Brenda Newman. *Precision type font reference guide, version 5.0*. Precision Type, Commack, NY, USA, 1995. ISBN 0-9646252-0-2. xxv + 653 pp. LCCN Z250.7 .P74 1995.

[22] Jeff Level, Bruce Newman, and Brenda Newman. *Precision type font reference guide, version 5.0*. Hartley & Marks Publishers, Point Roberts, WA, USA and Vancouver, BC, Canada, 2000. ISBN 0-88179-182-2. xxv + 653 pp. US$39.95, CDN$59.95. Republication of [21].

[23] Tim Lindholm and Frank Yellin. *The Java Virtual Machine Specification*. Addison-Wesley, Reading, MA, USA, second edition, 1999. ISBN 0-201-43294-3. xv + 473 pp. LCCN QA76.73.J38L56 1999. US$42.95.

[24] Jon Meyer and Troy Downing. *Java Virtual Machine*. O'Reilly & Associates, Inc., 103a Morris Street, Sebastopol, CA 95472, USA, Tel: +1 707 829 0515, and 90 Sherman Street, Cambridge, MA 02140, USA, Tel: +1 617 354 5800, February 1997. ISBN 1-56592-194-1. xxiv + 426 pp. LCCN QA76.73.J38M49 1997. US$32.95. URL `http://www.oreilly.com/catalog/javavm;http://www.ora.com/www/item/javavm.html`.

[25] Stephen Moye. *Fontographer: type by design*. MIS:Press, New York, NY, USA, 1995. ISBN 1-55828-447-8. xi + 275 pp. LCCN 250.8.F65 M69 1995.

[26] John Plaice and Yannis Haralambous. The latest developments in Ω. *TUGboat*, 17(2):181–183, June 1996. ISSN 0896-3207.

[27] Eric S. Raymond. *The cathedral and the bazaar: musings on Linux and open source by an accidental revolutionary*. O'Reilly & Associates, Inc., 103a Morris Street, Sebastopol, CA 95472, USA, Tel: +1 707 829 0515, and 90 Sherman Street, Cambridge, MA 02140, USA, Tel: +1 617 354 5800, revised edition, 2001. ISBN 0-596-00131-2, 0-596-00108-8 (paperback). xiv + 241 pp. LCCN QA76.76.O63 R397 2001.

[28] Herbert Schildt, American National Standards Institute, International Organization for Standardization, International Electrotechnical Commission, and ISO/IEC JTC 1. *The annotated ANSI C standard: American National Standard for Programming Languages C: ANSI/ISO 9899-1990*. Osborne/McGraw-Hill, Berkeley, CA, USA, 1990. ISBN 0-07-881952-0. xvi + 219 pp. LCCN QA76.73.C15S356 1990. US$39.95.

[29] The Unicode Consortium. *The Unicode Standard, Version 3.0*. Addison-Wesley, Reading, MA, USA, 2000. ISBN 0-201-61633-5. xxx + 1040 pp. LCCN QA268.U56 1996. US$50.00. URL `http://www.unicode.org/`

`unicode/standard/versions/`. Includes CD-ROM. The principal authors and editors of *The Unicode Standard, Version 3.0* are Joan Aliprand, Julie Allen, Joe Becker, Mark Davis, Michael Everson, Asmus Freytag, John Jenkins, Mike Ksar, Rick McGowan, Lisa Moore, Michel Suignard, and Ken Whistler.

[30] Walter Tracy. *Letters of credit: a view of type design*. David R. Godine, Publisher and G. Fraser, Boston, MA, USA and London, UK, 1986. ISBN 0-87923-636-1 (Godine), 0-86092-085-2 (Fraser). 219 pp. LCCN Z250.A2 T73 1986. US$27.50.