

Postscript fonts in T_EX

Phons Bloemen

phons@ei.ele.tue.nl

Abstract

A guide through the jungle of installing and using PostScript and other types of fonts in T_EX. The installation of the font is discussed, and it is shown how to use the fonts in an automatic font generation scheme.

1 Types of fonts

When searching for another set of fonts than the usual Computer Modern, one encounters a variety of other fonts to choose from. Some of them are easy to handle for T_EX, others create more difficulties. The various fonts may be classified according to the format in which they appear.

‘Real’ METAFONT These are fonts created with METAFONT, and exploit its capabilities. In other words, these fonts contain different designs for different sizes. The most prominent example of this kind is Computer Modern.

‘Fake’ METAFONT These fonts are created with METAFONT, but only in a single design size. Other sizes are created by magnifying the design. Examples are Malvern and the German fonts of Yannis Haralambous.

Postscript A large collection of fonts are in PostScript format, or ‘Adobe type 1’ format. A PostScript font is designed for a single design size. When other sizes are needed, magnification takes place. PostScript fonts come in .pfa or .pfb files, which contain the character glyphs. The .pfa files are 7bit ASCII files, they are mainly used on Unix systems. The .pfb file is an 8bit file format, found on MS-DOS machines. Each PostScript font is accompanied by an .afm file, which contains information about the sizes of the characters.

Ghostscript The ‘35 standard PostScript fonts’ are also available in a .gsf format. This format was introduced by Ghostscript. In order to distribute Ghostscript as a useful program, the 35 fonts had to be supplied, without infringing Adobe’s copyrights. This was done by creating them again: the .gsf files. However, the quality of these .gsf fonts is very poor. Ghostscript is able to use real PostScript fonts, when they are available.

HP softfonts Again, a large amount of fonts is distributed in a format suitable for downloading into HP Laserjet printers. They can be converted for use by T_EX, with utilities like `sfptopk`, `hptfm2pl`. How this is done is not discussed here.

Windows TrueType This font format is found on MS-DOS computers running MS Windows, or Macintosh computers. There are currently no tools to convert these for use by T_EX. However, most of the fonts in TrueType format are also available in PostScript format.

Most PostScript fonts are commercial (like the ‘standard 35 PostScript fonts’), and .pfa/.pfb files of them are not to be found on public file servers like CTAN. Some font suppliers have put one of their creations in the public domain, like Bitstream Charter and Adobe Utopia. CD-ROM versions of Corel Draw contain several PostScript fonts. Public domain PostScript fonts, of varying quality, can be found at `ftp.cica.indiana.edu` or its mirror `ftp.nl.net.nl`.

After the choice is made, and the font files are obtained, they must be installed on the system.

- Install the font such that T_EX recognizes it: create the .tfm files.
- Create a style file to use the font in L^AT_EX: create the .sty and .fd files.
- Set up the previewer to display the font: create the .pk files.
- Implement automatic font generation.
- Instruct the printer driver how to process the font.

2 Previewing

Previewing is an important step in creating T_EX documents, and a lot of effort has been put in creating comfortable software to preview .dvi files. Previewers of choice are `xdvi` on Unix platforms, and `dviscr` (from emT_EX) on MS-DOS platforms. `dviscr/xdvi` use .pk font files when previewing a document: there must be .pk files available in different sizes of each font. Usually, they are stored using filenames like `cmr10.300pk` or `300dpi/cmr10.pk`. They may also be stored in a *font library*, a file format (.fli) introduced by emT_EX.

2.1 Automatic font generation

An important option of the previewers is *automatic font generation*. When the previewer can not find a particular

.pk file, it can spawn a subprocess to create it. On Unix, this usually is a shell script called MakeT_EXPK. On MS-DOS, there is a program mfjob.exe which performs this task. The subprocess investigates which size of the missing font is to be generated, and then calls METAFONT with the appropriate parameters. In order to get this scheme working, there must be a .t_fm and a .mf file available for each font.

Using automatic font generation has advantages above the use of a fixed set of fonts or font libraries: there are no problems anymore when some odd size of a rare font is needed. No more ugly substitutions! Another bonus is less use of disk space: It is not necessary to have a full set of all possible *magsteps* of Computer Modern ‘online’. Fonts are created when they are needed, and from time to time the directory tree containing the .pk files can be deleted to free disk space.

2.2 Previewing PostScript fonts

When PostScript fonts are to be previewed, there are four possibilities to get them on the screen.

- d_{vi}drv, x_{dvi} is used. There is no way of converting a PostScript font to a .pk file, and a substitution is used. This solution is clearly not satisfactory.
- The PostScript font is converted to METAFONT. There is a program ps2mf by Erik-Jan Vens [4], which converts a PostScript font in .pfa form to a METAFONT file. The .t_fm file necessary for T_EX and automatic font generation can be created by running METAFONT. The resulting .mf files are rather large: in the order of 300 kb per font. A METAFONT with a high capacity (bigMETAFONT) must be used to process them. The advantage is that the favorite previewer still can be used.
- The use of x_{dvi}/d_{viscr} is abandoned. The .dvi file is converted to a Postscript .ps file by d_{vips}. The PostScript file is then previewed with Ghostscript. The advantage of this scheme is that there is no extra setup required for previewing. Yes, d_{vips} must be installed (and this may cause some problems), but the documents are to be printed on a PostScript printer anyway¹. Almost all versions of Ghostscript have the ability to show the page on the screen (using X11 or VGA graphics), but the user interface of Ghostscript is not very friendly. This problem is solved by g_{hostview} for X11 and g_{sview} for MS Windows. However, these two previewers are not as easy to use as x_{dvi}/d_{viscr}, an extra conversion step by d_{vips} is needed to preview, and the previewing is slower.
- The last option (and IMHO the best) is the use of ps2pk by Piet Tutelaars. ps2pk can convert a PostScript form directly into .pk format. It needs an .afm and a .pfa/.pfb file to do this. There are also options for magnification, slanting and expanding/condensing the font. Above all, it is FAST. Now it is possible to preview PostScript fonts, while us-

ing x_{dvi}/d_{viscr}. The newest version of x_{dvi} (v1.18) is even able to preview .eps pictures, by calling Ghostscript.

Before we discuss how to set up automatic font generation using ps2pk, and how to install the fonts for use by d_{vips}, first something about font names.

3 Font names and encodings

As T_EX runs on virtually every platform, a naming scheme for filenames must be used which runs on every platform. Since every platform has different conventions, the least common denominator must be used: this means there are 8 characters available for a file name, without an extension. The file name is case insensitive, and only letters and numerals may be used: this adds up to 36 possibilities per character position.

3.1 The Berry names

In [1], Karl Berry proposes a naming scheme for font files matching the restrictions above. The eight characters are divided as follows:

S TT W V E DD

- S** represents the source of the font, and is omitted if the source is unknown or irrelevant. Usually, the source is a company selling fonts, called a foundry. Different sources can supply the same typeface.
- TT** represents the typeface name. There is space for $36 * 36 = 1296$ different typefaces in this naming scheme. When the font naming scheme was first presented, this was not judged such a problem. However, there are many more typefaces in the world. See, for instance, [3]² which contains samples of 700 fonts.
- W** represents the weight, or ‘boldness’.
- V** represents the variant, and is omitted if both it and the width are ‘normal’. Many fonts have more than one variant. In this case, Berry proposes to give the variant letters in alphabetical order. Some fonts have so many variants that they exhaust the 8 character name space. The variant representation is the weak point of the naming scheme.
- E** represents the width (‘expansion’), and is omitted if it is ‘normal’.
- DD** represents the design size. One or two numerals are used, so fonts of design sizes between 1 and 99pt can be named. Very big fonts with design sizes above 100pt do exist.... The design size is omitted if the font is linearly scaled from a single .t_fm file, which is the case when using PostScript fonts, and gives us 2 extra positions to put other font characteristics in.

Table 2.2 shows a ‘quick reference card’ for the Berry names. The weight, variant, and width are probably best

¹ Of course, Ghostscript may be used to convert the PostScript file for Laserjet or matrix printers

² Available at ‘De Slegte’

| Berry names quick reference chart | | |
|---------------------------------------|--------------------------------|------------------------------------|
| Format of a Berry name: S TT W V E DD | | |
| Source (S) | Variant/Appearance (V) | Variant/Encoding (V) |
| 9 unknown | 1 semi sans | 3 fractions |
| a Autologic | 4 fax | 9 oldstyle digits |
| b Bitstream | 6 semi serif | c small caps |
| c Compugraphic | b bright | z Cyrillic |
| d DTC | d display, titling, caption | k Greek |
| f freely distributable | e engraved, copperplate | 5 phonetic (IPA) |
| g GNU / FSF | f Fraktur, gothic, blackletter | m math italic |
| h Bigelow & Holmes | g grooved | v math extension |
| i ITC | h shadow | y symbol |
| l Linotype | i text italic, kursiv | 0 Adobe standard encoding |
| m Monotype | j invisible | 2 changed chars, (Adobe's Dfr) |
| n IBM | l outline, open face | 81 ISO Latin 1 (8859/1) |
| o Corel (?) | n informal | 82 ISO Latin 2 (8859/2) |
| p Adobe | o oblique (i.e., slanted) | 85 ISO Latin 5 (8859/5) |
| r 'raw' [obsolete] | p ornament | 8m Macintosh standard |
| s Sun | r normal (roman or sans) | 8w Windows ANSI |
| u URW | s sans serif | 7t old 7-bit T _E X(OT1) |
| x AMS (math) | t typewriter | a Adobe alternate |
| y Y& Y | u unslanted italic | q TeX extended (Cork, DC) |
| z bizarre | w script, handwritten, swash | x Adobe expert |
| Weight (W) | Width (E) | Typeface (TT) (extract) |
| a hairline, thin | o ultra condensed | ag AvantGarde |
| i extra light | u ultra compressed | bk Bookman |
| l light | t thin | cr Courier |
| k book | q extra compressed | hv Helvetica |
| r regular | p compressed | nc NewCenturyS |
| m medium | c condensed (hand) | pl Palatino |
| d demibold | n narrow (auto) | tm Times |
| s semibold | r regular (omitted) | zc ZapfChancery |
| b bold | x extended (hand) | ch Charter |
| x extra bold | e expanded (auto) | ut Utopia |
| h heavy | w wide | nm Nimbus |
| c black | | lc Lucida |
| u ultra | | aq Antiqua |
| p poster | | gs GillSans |
| | Design size (DD) | |
| | Two decimal digits | |

Table 1: A quick reference chart for Berry Names

taken from the original source of the typeface, instead of trying to relate them to some external standard. Examples of Berry names for the 35 standard Postscript fonts are also given in table 2.2

3.2 Problems using Berry names

The Berry naming scheme is not perfect. A drawback is the small amount of possible typeface codes (almost always a lookup table is necessary to find the correct typeface code). The table with typeface codes is updated regularly. Get the newest version from CTAN: /docs/ fontname.

A bigger problem is the representation of font variants. The variants include codes for appearance (like 'grooved', 'italic'), different character sets (Cyrillic, small caps), and different encodings. Especially the large amount of encod-

ings imposes problems. Also, the 'alphabet' rule is broken in a lot of cases (see the examples of the Lucida Bright names: both names are used!). Combined with the fact that the position for 'width' is usually omitted, this may give rise to anomalies like one name being used for two fonts, or one font being given two different names.

The fact that weight and variant are sometimes omitted makes it very hard to 'decode' a Berry name. In some cases, the 'slanted' version of a font can be generated from the 'regular' or 'roman' version. In this case, the Berry name must be decomposed to find out whether slanting is needed.

| | | | | | | |
|--------|---------|--------|---------|----------------------|--------|------------------------|
| pagk | pagko | pagd | pagdo | AvantGarde | pzdr | Zapf-Dingbats |
| pbkl | pbkli | pbkd | pbkdi | Bookman | psyr | Symbol |
| pcrr | pcrro | pcrb | pcrbo | Courier | pzcmri | Zapf-Chancery |
| phvr | phvro | phvb | phvbo | Helvetica | | |
| | | phvrrn | phvron | Helvetica-Narrow | | |
| pncr | pncri | pncb | pncbi | NewCenturySchoolbook | mtmi | MathTimes MathItalic |
| pplr | pplri | pplb | pplbi | Palatino | mtex | MathTimes Extension |
| ptmr | ptmri | ptmb | ptmbi | Times | mtsy | MathTimes Symbols |
| bchr | bchri | bchb | bchbi | Charter | unmr | Nimbus Roman |
| bcrr | bcrrro | bcrb | bcrrbo | Bitstream Courier | unmrs | Nimbus Sans |
| ncrr | ncrrro | ncrb | ncrrbo | IBM Courier | uaqrcc | AntiquaT |
| putr | putri | putb | putbi | Utopia | ugqb | GroteskT |
| hlcrb | hlcrbi | hlcdb | hlcdbi | LucidaBright | hlcrf | Lucida-Blackletter |
| hlcr4 | hlcr4i | hlcd4 | hlcd4i | LucidaFax | hlcriw | Lucida-Calligraphic |
| | hlcri4 | | hlcdi4 | * | hlcrh | Lucida-Handwriting |
| hlcrs | hlcris | hlcds | hlcdis | LucidaSans | hlcrbm | LucMath Mathitalic |
| | hlcrsi | | hlcdsi | * | hlcrbv | LucMath MathExtensions |
| hlcrst | hlcrost | hlcbst | hlcbost | LucidaSansTypewriter | hlcrby | LucMath Symbols |
| | hlcrsto | | hlcbsto | * | hlcrba | LucMath Arrows |

Table 2: Examples of Berry Names

3.3 Font encodings

A lot of the confusion about installing PostScript fonts is caused by encodings. An encoding is the way in which the characters are ordered in a font. For instance, the encoding defines the position ‘65’ to be the letter ‘A’. If there was only one encoding, life would be easy: the ‘A’ would always be found on position 65. The ASCII table is an example of an encoding. However, ASCII, being a 7-bit encoding, did not specify which symbols should be encoded on the positions above 127. When the use of computers grew larger and larger, the need for special (language specific) characters also became greater. Since there was no standard, everyone defined the positions above 128 in his own way (trying to push his encoding as the new ‘standard’, like IBM-ASCII on PC’s). Also, ASCII defined positions 0-31 as ‘control characters’, used to send commands to terminals. These positions were also misused to put all kinds of symbols on. The result: a big mess, hundreds of encodings and a lot of tedious programming to provide mappings between encodings.

In case of T_EX, the situation is not different: T_EX uses his own encodings, derived from the standard ASCII. In fact, there are about 8 different T_EX specific font encodings, the most important are *T_EX text* (also called *OT1*, *Old 7-bit T_EX* encoding), and the *Cork/DC* encoding (also called *T1*). When PostScript fonts are to be used, a mapping must be created, since PostScript fonts use different encodings. The *Adobe Standard* encoding is the most important one. The three encodings mentioned above are shown in table 5.

Virtual fonts are used to map the encoding of PostScript and other fonts to T_EX encoding. The main application of virtual fonts is to incorporate PostScript fonts in T_EX. Two .tfm files are used, a ‘cooked’ file and a ‘raw’ file.

The ‘cooked’ file is used by T_EX and has a T_EX encoding. The ‘raw’ .tfm file is generated from the .afm file of the PostScript font and still has the same encoding as the original font. A virtual font file .vtf with the same name as the ‘cooked’ .tfm file performs the mapping.

In earlier versions of the naming scheme, the ‘raw’ font was recognizable by an ‘r’ prefix. This was a mistake, since the ‘raw’ font is just another encoding. This is admitted in later versions of the naming scheme: the ‘raw’ Adobe-encoded fonts now get the ‘variant’ letter 0 to indicate Adobe encoding. The default encoding is the old 7-bit T_EX encoding (the ‘cooked’ encoding), and this one does not get a variant letter. When the Cork encoding was introduced, the ‘variant’ letter ‘q’ was used.

To make things worse, the latest versions of the naming scheme introduce the ‘7t’ variant letter to indicate the old T_EX encoding. This last step is unnecessary in my opinion: the old 7-bit T_EX encoding is the default encoding of a font in the Berry name scheme. Introducing yet another set of names only increases confusion. To summarize the confusion, using Charter Roman as an example:

- Adobe encoding: rbchr (OBSOLETE!) → bchr0.
- 7-bit OT1 T_EX encoding: bchr → (?) bchr7t.
- 8-bit T1 Cork encoding: bchrq.

The latest releases of the major software packages dealing with PostScript fonts, like dvips and psnfss2e, use the ‘new’ names. The old names with ‘r’ prefix are still widespread, and can be found at sites running older versions of the software.

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|
| 00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? | |
| 40 | @ | 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 | [| \ |] | ^ | _ |
| 60 | ` | 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 | { | | } | ~ | |
| 80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A0 | ı | ç | £ | / | ¥ | f | § | ¤ | ' | " | < | > | fi | fl | - | † | ‡ | • | ¶ | • | , | " | » | ... | % | ¿ | | | | | | |
| C0 | ` | ^ | ~ | - | ˘ | ˙ | ˚ | ˛ | ˜ | ˝ | ˆ | ˜ | ˘ | ˙ | ˚ | | | | | | | | | | | | | | | | | |
| E0 | Æ | ˆ | ˜ | ˘ | ˙ | ˚ | ˛ | ˜ | ˝ | ˆ | ˜ | ˘ | ˙ | ˚ | æ | ı | ı | ø | œ | ß | | | | | | | | | | | | |

Table 3: bchr0 Adobe font encoding

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | Γ | Δ | Θ | Λ | Ξ | Π | Σ | Υ | Φ | Ψ | Ω | ff | fi | fl | ffi | ffl | ı | ■ | ` | ˘ | ˙ | ˚ | ˛ | ˜ | ˝ | ˆ | ˜ | ˘ | ˙ | ˚ | | |
| 20 | ■ | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | ı | = | ¿ | ? |
| 40 | @ | 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 | [| " |] | ^ | ˘ |
| 60 | ` | 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 | - | - | ˘ | ˙ | ˚ |

Table 4: bchr, bchr7t Old 7 bit T_EX encoding (L^AT_EX2.09)

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| 00 | ` | ˘ | ˙ | ˚ | ˛ | ˜ | ˝ | ˆ | ˜ | ˘ | ˙ | ˚ | ˛ | ˜ | ˝ | ˆ | ˜ | ˘ | ˙ | ˚ | ˛ | ˜ | ˝ | ˆ | ˜ | ˘ | ˙ | ˚ | ˛ | ˜ | ˝ | | |
| 20 | ı | ■ | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 40 | @ | 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 | [| \ |] | ^ | _ | |
| 60 | ` | 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 | { | | } | ~ | - | |
| 80 | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | | |
| A0 | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | | |
| C0 | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | | |
| E0 | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | Ǻ | | |

Table 5: bchrq New 8 bit T_EX Cork/DC encoding (L^AT_EX2_ε)

4 Installation of new fonts

In this section, the process of installing new fonts for T_EX is discussed. We will use the Bitstream Charter font as an example, when the installation of PostScript fonts is discussed.

4.1 Setting up the system

In the rest of this article, it is assumed that automatic font generation is used. Here is a short description of how to set this up. First, METAFONT must be available on the system.

On Unix systems, the MakeT_EXPK shell script must be installed. This script is included in the dvips and xdvi distributions. It must be edited to reflect the local T_EX setup (pathnames of directories containing .mf, .tfm, .pk files). Once the script is installed, it will be recognized by xdvi and dvips and used.

For emT_EX, the mfjob.exe program must be installed. The dvidrv previewers must be instructed to use automatic font generation with the commands +metafont-mode, +mfjob-file, +batch-mode. See the dvidrv manuals for more information.

4.2 New Metafont fonts

The installation of a new METAFONT font is not too difficult. The main problem is how to get a .tfm file. Suppose we have a font klinz, which is in METAFONT format.

- Put the file klinz.mf (and the files which are read when it is processed by METAFONT) in a directory which can be found by METAFONT. The most recent versions of METAFONT are able to search subdirectories (Unix versions compiled with the 'kpathsea' library, emMETAFONT version β5 or higher). Use this

feature to keep some order in your METAFONT fonts, and create a subdirectory for klinz.

- Create the .tfm file by calling METAFONT: `mf 'mode=laserjet; mag=1; input klinz'` It will generate `klinz.tfm` and `klinz.300gf`.³
- Put the .tfm file in the directory with other .tfm files (`/usr/local/tex/tfm`, `\emtex\tfm`). The .gf file may be deleted. The font is now ready to use.

4.3 New PostScript fonts

Installing a PostScript font is more complicated. We will use Bitstream Charter (a public domain font, donated by Bitstream) as an example.

4.4 Get the correct filenames

First of all, make sure you have the proper files, and that they have the correct names, according to the Berry naming scheme: `bchr0.afm`, `bchri0.afm`, `bchb0.afm`, `bchbi0.afm`, and the corresponding .pfa/.pfb files. Please rename the fonts if they do not have the trailing '0': this indicates Adobe encoding.

The .afm files can be found on public file servers (if they were not supplied with the font files .pfa/pfb themselves). Look in `CTAN:/fonts/postscript/afm` for .afm files for the standard PostScript fonts. They are already converted to Berry names. More .afm files are in `CTAN:/fonts/postscript/adobe/AFMfiles`.

The installation procedure will create the corresponding \TeX encoded fonts `bchr7t/bchr` and `bchrq` from `bchr0`. The `7t` suffix (indicates old \TeX (OT1) encoding), is omitted. *If you decide to omit it, this decision refers to all occurrences of '7t'*. The new $\LaTeX 2_\epsilon$ packages use the '7t' suffix, so these also must be adapted when you omit the suffix. Install the font files in the directory containing all PostScript fonts (`/usr/local/tex/ps/fonts`, `\emtex\ps\fonts`).

4.5 Create .pl .vpl with fontinst

Use the `fontinst` [2] package to create .pl and .vpl files. This package consists of a set of special \TeX files. We will use \TeX to scan the .afm file of the PostScript font, and to create the .pl, .vpl files. The `fontinst` package is best installed in a separate directory, searched by \TeX . It is easy to use: create a file containing

```
\input fontinst.sty
\latinfamily{bch}{}
\bye
```

and run plain \TeX on it.⁴ It takes some time (about 5 minutes on a 486/50) to process the complete Charter family (4 fonts). After processing, \TeX has created a set of .pl files and .vpl files, which are human-readable versions of the .tfm and .vf files. Small-caps and slanted versions of

the font in all 3 encodings are also available: *Adobe, OT1, T1*.

The `fontinst` package may be slow, it will give the best results, with almost complete T1 encodings of the font. It also contains a bag of tricks for special things like creating new math fonts, and combining two different fonts. See [2] for more details.

4.6 Quick and dirty afm2tfm

A different approach is the use of `afm2tfm`. This method is a little faster, but less precise. The `afm2tfm` program is included in `dvips` distribution. The simplest form is

```
afm2tfm bchr0.afm -v bchr.vpl bchr0.tfm
```

This creates `bchr0.tfm`, which is still Adobe-encoded, and `bchr.vpl`, which has the old \TeX (OT1) encoding. If you are only using $\LaTeX 2.09$ (with NFSS1 and PSNFSS2.4), this is enough.

There are also some tricks, to create small-caps and slanted versions of fonts. The second and third argument of `vp-tovf` may be omitted.

```
afm2tfm bchr0.afm -V bchrc.vpl bchr0.tfm
afm2tfm bchr0.afm -s .167
                    -v bchro.vpl bchr0.tfm
```

When a different encoding is needed, like the *T1* encoding used by $\LaTeX 2_\epsilon$, things get complicated:

```
afm2tfm bchr0.afm -t ec.enc
                    -v bchrq.vpl bchr0.tfm
vpltovpl bchrq.vpl bchr0.afm
```

Here `ec.enc` contains the *encoding vector* of the T1/Cork encoding. Various other .enc files are distributed with `afm2tfm`, more information about the use of encoding files is in the `afm2tfm` manual. The `vpltovpl` program is used to construct combined characters (such as accented characters), and to put them in the new font. The result, however, is not as complete as the results obtained by using `fontinst`.

4.7 Convert .vp .vpl files

Use `pltotf` on every .pl file to create a .tfm file, for example, `bchr0.pl`:

```
pltotf bchr0.pl bchr0.tfm
```

Use `vptovf` on every .vpl file, to create a .tfm and .vf file (same base filename for every argument).

```
vptovf bchr.vpl bchr.vf bchr.tfm
```

The two programs convert the human-readable 'property list' files in \TeX readable 'font metric' files. They are distributed together with METAFONT. The versions of `vp-tovf` and `pltotf` distributed with `emTeX` derive the file names from the first argument. The .pl and .vpl files may be discarded after the conversion.

Now we have a bunch of newly created .tfm files. Put them in the directory where the other .tfm files reside. The same applies to the .vf files. The fonts can now be used by \TeX , but further steps are necessary to preview them.

³When entering this command you may want to use shell escape characters, in order to prevent the Unix shell or MSDOS command.com from 'interpreting' the options

⁴`emTeX` users may have problems running `fontinst`. Add the parameter `-ms 6000` to increase stack size

4.8 Alternative: pre-installed files

For those who do not want to run into trouble when creating the `.tfm` files themselves, there are packages which contain ‘pre-compiled’ `.tfm` and `.vf` files for PostScript fonts. For the 35 standard PostScript fonts, they are grouped together in the `psnfss2e` package. This package also contains style files and `.fd` files for use with L^AT_EX_{2 ϵ} . The metric files in this package (CTAN: `/macros/latex/packages/psnfss`) were created using `fontinst`. Simply extract the `.tfm` and `.vf` files and put them in the correct directories. These metric files also work for L^AT_EX_{2.09} with NSFSSv1 and the PSNFSS 2.4 package of S. Rahtz.

The directory CTAN: `/fonts/metrics` contains metric files and font description files for PostScript fonts, (`.tfm`, `.vf`, `.fd`), grouped together per font family.

Metric files are also supplied in the `dvips` package. However, they still use the obsolete ‘r’ prefix for the raw fonts (even in version 5.58).

4.9 Caution with math symbol fonts!

When installing a PostScript math font like MathTime or LucidaMath, be very careful. Do not try to create the `.tfm` file for the math extension or math symbol font using `afm2tfm`, or weird errors from L^AT_EX will occur. These are caused by the fact that math extension fonts have additional metric information, which is not generated by `afm2tfm`. For MathTimes, take the `.tfm` files from the `mathtime` (CTAN: `/fonts/mathtime`) package. This package also contains L^AT_EX style files to use the news math fonts in T_EX. Metric files for LucidaMath are found in (CTAN: `/fonts/metrics/bigholm`). Style files to use LucidaBright in L^AT_EX_{2 ϵ} are found in the `psnfss2e` package.

4.10 Inform dvips

The newly installed fonts must be made known to `dvips`. The PostScript font files (`.pfa/pfb`) go into a directory specified in `DVIPSHEADERS`, together with the other header files used by `dvips`. There is a file `psfonts.map`, which is used by `dvips` to determine what to do when it encounters a particular font.⁵ For our newly installed Charter font family, the following lines must be added to `psfonts.map`:

```
bchr0 CharterBT-Roman <bchr0.pfb
bchro0 CharterBT-Roman ".167 SlantFont" <bchr0.pfb
bchri0 CharterBT-Italic <bchri0.pfb
bchb0 CharterBT-Bold <bchrb0.pfb
bchbo0 CharterBT-Bold ".167 SlantFont" <bchbo0.pfb
bchbi0 CharterBT-BoldItalic <bchbi0.pfb
```

The `psfonts.map` file is used to determine which file to send to the printer as a header file, when a font is requested.

As can be seen from the example, only the ‘raw’ Berry font names have to be included in the `psfonts.map` file. The second field of each line contains the font name itself. It can be found at the top of the `.afm` file of the font. Transformations like ‘slant’ or ‘enlarge’ are also mentioned. The

generation of ‘small caps’ is not: this is done via the virtual font mapping.

When fonts are resident in the PostScript printer, omit the field `<bchr0.pfb` in the corresponding lines. When Ghostscript is used for further processing of the print jobs, all available fonts are ‘resident’ (at least, if Ghostscript has access to the `.pfa/.pfb` files).

If `afm2tfm` was used to create the `.tfm` files, you may have noticed that `afm2tfm` prints a line on the standard output when it is processing. This line has the exact format for use in `psfonts.map`. It may be captured using output redirection.

4.11 Inform Ghostscript

Ghostscript has also a file `fontmap` which it uses to determine how to handle a font. It has a different format than `psfonts.map`. Ghostscript uses the ‘real’, long font name as an index. It will only encounter this type of names in PostScript files generated by `dvips`, and it does not use any other metric file like `.tfm`, `.vf`. In the case of the new Charter family, following lines must be added:

```
/CharterBT-Roman      (bchr0.pfb) ;
/CharterBT-Italic     (bchri0.pfb);
/CharterBT-Bold       (bchb0.pfb) ;
/CharterBT-BoldItalic (bchbi0.pfb);
```

The `.pfb` fonts files must also be installed in a directory listed in the `GS_LIB` environment variable, otherwise Ghostscript substitutes them with the really ugly *Ugly* font.

The new version 3 of Ghostscript does not need this anymore. It builds the `fontmap` table itself: put all fonts (`.pfb` files) in a directory listed in `GS_FONTPATH`. It only needs `fontmap` to list the `.gsf` fonts in (the public domain surrogates of the 35 standard PostScript fonts). Users of Ghostscript version 3 only need to edit the `fontmap` file if they obtain the ‘real’ standard PostScript fonts, to replace the `.gsf` files. This feature alone is worth the upgrade.

4.12 Using .gsf fonts with T_EX

If you want to use the Ghostscript fonts for previewing, act like the surrogate `.gsf` fonts are the real PostScript fonts. Get the corresponding `.afm` files from a file server like CTAN, or the 4allT_EX CDROM, and execute the procedure described above.

5 Setting up ps2pk and automatic font generation

In order to keep out beloved previewers `xdvi` and `dvips`, some way must be found to convert PostScript font into bitmapped `.pk` files. The package `ps2pk` by Piet Tute-laars will perform this task: it is fast, and the output is of high quality (when good quality `.pfa/.pfb` files are used). The problem is how to incorporate `ps2pk` into an automatic font generation scheme.

⁵The `.map` file is set in the `dvips` configuration file `config.ps`. Additional (local) `.map` files may also be specified

5.1 Only ‘raw’ fonts are needed

If the installation of `.tfm` and `.vfm` files for a PostScript font was done correctly, only ‘raw’ (Adobe encoded) fonts must be generated. When a previewer search for a font, it first looks at the name of the `.tfm` file given in the `.dvi` file. This `.tfm` file was used by T_EX to base its typesetting on. The previewer will now try to find a `.pk` file in the appropriate magnification. If it is not found, it searches for a `.vfm` file for the requested font. In a `.vfm` file, it may get new pointers to other `.tfm` files. If a `.vfm` file is not found either, the automatic font generation mechanism is called, with the name of the ‘not found’ font and the desired sizes as parameters.

In case of a PostScript font, the `.vfm` file maps the T_EX encoding vector onto the Adobe encoding vector, so the `.tfm` files pointed to are ‘raw’ `.tfm` files in Adobe format. The fontnames discussed here are in the Berry format.

Example: `bchrq` or `bchr7t/bchr` is used in a document. There are no `.pk` files with these name, but there are `.vfm` files. They all point to `bchr0.tfm`. For the ‘raw’ fonts, there are no `.vpl` files, so the names of ‘raw’ fonts are passed to the automatic font generator, if the `.pk` files are not there already.

5.2 Mixing Metafont and ps2pk

The automatic font generator is the `MakeTeXPK` script on Unix, and the `dvidrv.exe` program on MSDOS (which calls `mfjob.exe`). The `mfjob.exe` program on DOS is only able to ‘drive’ METAFONT. The `dvidrv.exe` program is very simple, and can easily be replaced by a batch file, which looks like `MakeTeXPK`. The use of 4DOS as command processor on a MSDOS system is recommended, since some arithmetic takes place when calculating the desired sizes.

The `MakeTeXPK` script can be adapted in the following way. Before it passes control to METAFONT it first checks if there is a METAFONT file available to generate the font with. This may be done by a file search operation on the METAFONT font directory. If no `.mf` file is found, ‘it must be a PostScript font’. Then, `ps2pk` enters the arena to generate the `.pk` file. Search the directory containing the `.afm` `.pfa` `.pfb` files to find the correct font file (if nothing is found, give an error message). Run `ps2pk`, and place the `.pk` file in the appropriate directory.

An adapted `MakeTeXPK` script is distributed with `ps2pk`. Examples of how to set this up are also given in [5], both for Unix and MS-DOS. Of course, the mixed use of `ps2pk` and METAFONT is incorporated in 4T_EX.

5.3 Generating the correct font. . .

To create a `.pk` file from a PostScript font at 300dpi, issue:

```
ps2pk -v -X300 -Y300 -P10 -abchr0.afm
      bchr0.pfb bchr0.300pk
```

It is also able to slant PostScript fonts, or to expand them. Note the change in the Berry name of the resulting `.pk` file.

```
ps2pk -v -X300 -Y300 -P10 -abchr0.afm
```

```
-S 0.167 bchr0.pfb bchro0.300pk
ps2pk -v -X300 -Y300 -P10 -abchr0.afm
-E 0.8 bchr0.pfb bchrn0.300pk
```

However, problems arise when PostScript fonts are created by slanting or expanding other fonts. The Berry name must be decomposed, in order to extract the ‘base’ font used for the ‘slanting’ operation. And decomposing Berry names is not easy, see section 3.2.

The following is a simple scheme for decomposing Berry names, which can be implemented in scripts or batch files.

- No *design size* is used.
- Check last position in Berry name. If it is a *width* modifier, use this to set the ‘E’ option, and remove it from the name. The *width* modifier always comes last in a Berry name.
- Again, check the last position. If it is the ‘0’ of Adobe-encoding, strip it from the name.
- Check the last position for other font variants, like ‘slanted’ to set the ‘S’ parameter.
- Bear in mind that the first 4 positions of a Berry name do not contain *variant* symbols.
- The *variant* letters should be in alphabetical order. However, this rule is often broken.

Now, if you still understand how to decompose Berry names, you must admit that it causes many problems. (if you don’t understand it anymore, you know it is problematic, and you wish for another, simpler scheme).

5.4 Use psfonts.map to find the font

A nice solution to the problem of decomposing a Berry name, is to borrow the `psfonts.map` file from `dvips`. This file is modified such that all fonts are considered ‘non-resident’: a field like `<bchr0.pfb` appears on every line. We can now ‘decode’ the lines in `psfonts.map` to construct the command line for `ps2pk`: it list all possible ‘raw’ font names (in Berry format) in the first field on every line (the ‘index’ field), gimmicks like slanting and expanding are clearly documented for use by `dvips`, and the ‘base’ file is also given: the `<bchr0.pfb` like field. See section 4.10 for an example of the `psfonts.map` lines used for the Charter family.

The drawback is that `dvips` will now consider every font being ‘non-resident’ in the printer, and prepend all fonts used in the print job to the PostScript file sent to the printer. Unfortunately, there is no way to instruct `dvips` never to download fonts.

5.5 You only have .gsf files

If you use `.gsf` fonts instead of `.pfa/pfb` fonts, there are two options to preview them without ugly substitutions. However, do not expect top quality, since the `.gsf` fonts are poor.

First, there is a utility `gsftopk` which converts the `.gsf` font into `.pk` format. It is used in the same way as `ps2pk`. It works by internally spawning Ghostscript for rendering the character glyphs. Due to the fact that it calls Ghostscript

with more than 100 parameters, it does not work on MS-DOS machines (even if they have enough memory to accommodate Ghostscript, `gsftopk` and the previewer). On Unix systems it works fine. An adapted version of the MakeT_EXPK script is enclosed in the `gsftopk` package.

Since `gsftopk` does not work on MSDOS systems, some friendly guys created all the necessary `.pk` files, in a reasonable number of magnifications on a Unix system, and put them into emT_EX font libraries. These are on the 4allT_EX CDROM (also on the diskette distribution), they can also be found on CTAN:/fonts/postscript/pk-files in a rather large archive-file. These `.pk` files have the Adobe encoding, so please add a '0' to the font name (and remove the 'r' prefix!). Otherwise, the previewer can not find them: it must have 'raw' font names ending with a '0' which indicates the Adobe encoding. Unfortunately, the font libraries on the CDROM still contain files with old names....

Solving this may require some tedious renaming work: the font libraries (`.fli`) must be extracted into single `.pk` files, the files themselves renamed, and the libraries created again using `fontlib.exe`. After this operation, everything works fine, and previewing can be done using `dvipsr` without touching Ghostscript.

6 Creating style files for L^AT_EX

To facilitate the use of our new font family in L^AT_EX style files and font description files must be created. Examples of how such files look like are given below.

The two examples, `charter.sty` and `otlbch.fd` are computer-generated. I use a batch file or shell script to perform all steps needed to install a PostScript font, as discussed in the article. In the final stage, the batch file creates a style file and `.fd` files, such that the font is ready to use after installing. The `fontinst` package also creates a `.fd` when it processes a font family. With minor adaptations, it is also possible to let it create a `.sty` file as shown in the examples.

6.1 charter.sty

This is an example of a L^AT_EX style file which sets up a PostScript font. This style file can be used by both L^AT_EX2.09 and L^AT_EX2_ε. The L^AT_EX2_ε branch will read the `.fd` files corresponding to the Charter font: an example of such an `.fd` file is given in the next section. The L^AT_EX2.09 branch uses `PSNFSS` to get the appropriate `.tfm` files, in L^AT_EX2_ε, this task is performed by the `.fd` files.

The last part of the style file contains code to make the Charter font the default. The default sans-serif font will be `NimbusSans`, and `BitstreamCourier` will do the typewriter. The latter two fonts are set up by calling the appropriate style file (which look like the one given for Charter).

The style file can be used as an option for `documentstyle` in L^AT_EX2.09, or loaded with `usepackage` in L^AT_EX2_ε. The final effect is the whole document being

typeset in Charter (except the math symbols, which are not supplied in Charter).

6.2 otlbch.fd, t1bch.fd

L^AT_EX2_ε uses `.fd` files to find the names of the `.tfm` files it has to load when a specific font is requested. The file displayed is `otlbch.fd`. The file `t1bch.fd` can be created by 'search and destroy': $\{\text{OT1}\} \rightarrow \{\text{T1}\}$ and $7t \rightarrow q$. The `7t` suffix may be omitted.

7 Summary of the results

After successfully installing the PostScript font 'CharterBT Roman' on your system, the following files will be available, ready to use by L^AT_EX2.09 and L^AT_EX2_ε:

| | |
|--------------------------|-------------------------------------------------------------------|
| <code>bchr0.pfb</code> | character glyphs in Adobe encoding |
| <code>bchr0.afm</code> | font metrics (character sizes) in Adobe format and Adobe encoding |
| <code>bchr0.tfm</code> | font metrics, T _E X format, Adobe encoding |
| <code>bchr.vf</code> | maps Old T _E X encoding into Adobe encoding |
| <code>bchr.tfm</code> | font metrics in old T _E X encoding |
| <code>bchrq.vf</code> | maps Cork encoding into Adobe encoding |
| <code>bchrq.tfm</code> | contains font metrics in Adobe encoding |
| <code>bchr0.pk</code> | bitmapped character (in several sizes) |
| <code>t1bchr.fd</code> | font description file for Cork encoding |
| <code>otlbchr.fd</code> | ditto, for old T _E X encoding |
| <code>charter.sty</code> | style file to use the font in L ^A T _E X |
| <code>psfonts.map</code> | contains line how to create <code>bchr</code> . |

8 Automating this hassle

The procedure for installing PostScript fonts as it is described above can be automated by using batch files or shell scripts. Such a batch file is also incorporated in 4T_EX. In 4T_EX, a new [Makefont] utility menu is created. It automatically installs PostScript font families, and offers a 'sneak preview' for quick font viewing.

You may also take a look on a 'pre-installed T_EX installation', in which all necessary software is already installed. No more difficult installation procedures! Unfortunately, there are only two such installations available at this moment. One of them is the ready to run 4T_EX system on the 4allT_EX CDROM. The other is NT_EX, which is available for Linux systems. NT_EX is part of the Slackware Linux distribution, and comes on 10 HD floppys. It features a complete ready to run T_EX installation, including L^AT_EX2_ε, PostScript font support, and automatic font generation. Unfortunately for non-x86 users, the programs are delivered in binary form. The source code (`web2c` and so on) is not included.

References

- [1] Karl Berry. *Filenames for T_EX fonts*, volume v1.6. TeXINFO document, `karl@cs.umb.edu`, 1993.

```

%%Filename: charter.sty [4allTeX postscript font setup]
%%Created by: tex bch
%%Created using fontinst v1.332
%%This fontinst was hacked by Phons Bloemen
%%Based on PSNFSS 2.4.1 and 3 by S. Rahtz
%%Can be processed by LaTeX209/NFSS1 and LaTeX2e
%%Berry name composition {bch}{r}{b}{i}{}
%%Can be processed by both PSNFSS1 and LaTeX2e
%%Put this file in a TEXINPUT directory
\def\fileversion{3p}
\def\filedate{94/07/28}
\ifx\ProvidesPackage\undefined
\immediate\write\sixt@@n{File: charter.sty NFSS1 4allTeX fontinst (PhB)}
\ifx\selectfont\undefined
\@latexerr{No NFSS1 detected...}\@eha\endinput\fi
\ifx\declare@font\undefined\input{psnfss.sup}\fi
\def\load@charter%
{\postscript@fontdef{charter}{charter}{bchr}{bchri}{bchb}{bchbi}{bchrc}{bchro}}
\ifx\load@nimbussans\undefined\input{nimbussans.sty}\fi
\ifx\load@couribm\undefined\input{couribm.sty}\fi
\romanfont{charter}\sansfont{nimbussans}\typewriterfont{couribm}
\late@extra@def{bch}{}{}
\endinput\fi

\immediate\write\sixt@@n{File: charter.sty LaTeX2e 4allTeX fontinst (PhB)}
\ProvidesPackage{charter}[\filedate\space\fileversion\space
LaTeX2e 4allTeX fontinst (PhB)]
\RequirePackage{couribm}\let\ttdefault=\rmdefault
\let\sfddefault=\rmdefault
\RequirePackage{nimbussans}\let\sfddefault=\rmdefault
\def\rmdefault{bch}
\endinput

```

Figure 1: Example of a L^AT_EX font selection style file

```

%Filename: OT1bch.fd
%Created by: tex bch
%Created using fontinst v1.332
%THIS FILE SHOULD BE PUT IN A TEX INPUTS DIRECTORY
\DeclareFontFamily{OT1}{bch}{}
\DeclareFontShape{OT1}{bch}{b}{n}{<->bchb}{}
\DeclareFontShape{OT1}{bch}{b}{sc}{<->bchbc}{}
\DeclareFontShape{OT1}{bch}{b}{sl}{<->bchbo}{}
\DeclareFontShape{OT1}{bch}{b}{it}{<->bchbit}{}
\DeclareFontShape{OT1}{bch}{m}{n}{<->bchr}{}
\DeclareFontShape{OT1}{bch}{m}{sc}{<->bchrc}{}
\DeclareFontShape{OT1}{bch}{m}{sl}{<->bchro}{}
\DeclareFontShape{OT1}{bch}{m}{it}{<->bchri}{}
\DeclareFontShape{OT1}{bch}{bx}{n}{<->ssub * bch/b/n}{}
\DeclareFontShape{OT1}{bch}{bx}{sc}{<->ssub * bch/b/sc}{}
\DeclareFontShape{OT1}{bch}{bx}{sl}{<->ssub * bch/b/sl}{}
\DeclareFontShape{OT1}{bch}{bx}{it}{<->ssub * bch/b/it}{}
\DeclareFontShape{OT1}{bch}{b}{ui}{<->sub * bch/b/it}{}
\DeclareFontShape{OT1}{bch}{bx}{ui}{<->ssub * bch/b/ui}{}
\DeclareFontShape{OT1}{bch}{m}{ui}{<->sub * bch/m/it}{}
\endinput

```

Figure 2: Example of a NFSS2 font description file

- | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| [2] Alan Jeffrey. <i>The fontinst package</i> , volume 187075803X. alanje@cogs.susx.ac.uk, 1994. | [4] Erik-Jan Vens. Incorporating ps fonts in T _E X. <i>MaPS</i> , 9:81, 1992. |
| [3] Gordon Rookledge and Christopher Perfect. <i>Rookledge's International Typefinder</i> . Sarema Press, Carshalton Beeches, Surrey, GB, 1983. ISBN | [5] Norman Walsh. <i>Making T_EX work</i> . O'Reilly, Sebastopol, CA, USA, 1994. |