
Partial Font Embedding Utilities for PostScript Type 1 Fonts

Basil K. Malyshev and Michel Goossens

Abstract

This article describes the `fload` utility, that reads a PostScript file and selectively embeds only those characters from Type 1 PostScript fonts that are actually used inside the PostScript file. This can result in substantial savings in the size of PostScript files, especially for larger font sets (like the 256-character DC family) or when only a few characters of each font are used inside the PostScript file. Also memory requirements for PostScript interpreters dealing with these files can thus be reduced.

1 Introduction

Documents often use a lot of fonts and thus require printers with a sufficient amount of memory, especially if the fonts are embedded as Type 1 sources inside the output PostScript file. However, in most cases only a limited number of characters of each font is actually used, and PostScript files with embedded Type 1 fonts can thus be substantially shortened if only those characters are included.

The tool `fload` uses the `ghostscript` program (Aladdin Software) to examine the input PostScript file and extract the necessary information; then it embeds the needed characters in the output PostScript file.

Partial font embedding is implemented in three steps, as follows:

1. Creation of a PostScript file using any publishing system. *No* fonts should be embedded in this file; the `fload` procedure assumes that the Type 1 images of the fonts are available somewhere on the system.
2. The PostScript file is processed by the program `ghostscript` where some character loading commands, such as `show`, `awidthshow`, etc., have been redefined to construct the list of all characters used for each font. When the complete input file is read these font usage statistics are written into an intermediate file (with extension `fstat`).
3. The latter file is then read by the `subfont` program which embeds the needed characters for each font into the output file.

This approach can be applied to PostScript output generated by a variety of applications—it has been tested with `dvips`, `FrameMaker`, and MS Windows PostScript driver output. Moreover it handles fonts inside embedded pictures correctly. Its only

drawback is that `fload` needs some time to scan the input file to prepare the list of characters used and then to embed the (partial) fonts into the output file.

2 Setting up the `fload` system

2.1 The distribution set

The `fload` distribution includes the following:

1. A command file (a C-shell script `fload` for Unix, `fload.bat` for MS-DOS, and `fload.com` for VMS) that first invokes `ghostscript` with the `psfstat.ps` procedure and then executes the `subfont` program to embed the needed characters (see section 3.2).
2. A PostScript procedure `psfstat.ps` which redefines several important PostScript commands and collects information about the characters used in each font.
3. The `subfont` program (written in C) which extracts only the required characters from any ATM compatible Adobe Type 1 font and creates a new smaller font (see appendix A).

2.2 Installation

To make installation of the programs easy for Unix systems a C-shell script `INSTALL` is included. Before running the script, one must specify where certain programs and files reside, by editing the file `install.dirs` (see below).

```
# File: install.dirs
# System dependent executable files
# e.g. the subfont program
setenv FBIN /usr/local/bin
# System independent scripts
# e.g. the fload C-shell script
setenv FSHR /usr/local/bin
# Library files
# e.g. psfstat.ps, Fontmap, SubFont.map, *.FS
setenv FLIB /usr/local/lib/fload
# man pages (optional)
setenv FMAN /usr/local/man/man1
# font path
setenv FPATH myfonts:/usr/local/lib/texmf/fonts
# End of file 'install.dirs'
```

The installation then proceeds as follows:

1. Compilation of the `subfont` program.
2. The `subfont` program shares a single fontmap file (called `Fontmap.t1` in the `$FLIB` directory) with the `ghostscript` program. It is read by the `psfstat.ps` script in addition to the standard `Fontmap` file required by `ghostscript` to indicate where the font sources are located on the system; the format of this file is identical to a `ghostscript` map file and includes a list

of the font names and the associated file names where they reside, i.e., it is of the form:

```
/Type1_fontname (filename_on_system) ;
```

A Unix script `mkmap` (`mkmap.bat` on MS-DOS) is provided to create a map of the PostScript Type 1 files available in the specified directory tree, as shown in the following example:

```
[1] mkmap /usr/local/lib/texmf/fonts
/usr/local/lib/texmf/fonts
Total of 1 files are processed.
font /cmb10/ in file ./cm/type1/cmb10.pfb/
Total of 1 files are processed.
font /cmr10/ in file ./cm/type1/cmr10.pfb/
Total of 1 files are processed.
...
```

This procedure generates the file `Fontmap.t1`, that starts with (compare with the lines above):

```
/cmb10 (./cm/type1/cmb10.pfb) ;
/cmr10 (./cm/type1/cmr10.pfb) ;
...
```

3. All files from the `lib` distribution subdirectory are copied into the `$FLIB` library directory. The `INSTALL` script as distributed defines `$FLIB` as the directory `/usr/local/lib/fload`. In principle, `$FLIB` could also be defined as `/usr/local/lib/ghostscript` (the directory used by `ghostscript`). However, to minimize possible future name clashes between file names for `fload` and `ghostscript`, it is advisable to keep these directories distinct.
4. The names of fonts that are *not* to be included in the output PostScript file are the ones specified via the `-p` option of `fload`. By default these are the thirteen “standard” fonts, as specified in the file `Standard.FS` that comes with the distribution. Another distributed file, `35.FS` contains the names of the 35 “generally available” PostScript fonts present in most of the present-day PostScript printers. At the end of installation, users can add their own supplementary `FS` files specifying the names of fonts that are resident in their PostScript printer, and that thus need not be embedded in the generated output PostScript file. For instance, in Russia Cyrillic fonts could always be resident, or Kanji fonts in Japan or China. To save considerable amounts of disk space these fonts will not generally need to be embedded in the PostScript files, and their names must then be declared in a file with extension `FS`, and specified with the `-p` option of the `fload` command (see section 3.2). Of course, when these files have to be exported to sites where these fonts are not

available on the system, they will have to be embedded in the PostScript files. `FS` files reside in the `$FLIB` directory.

3 Using the fload program

This section shows how to set up `dvips` to generate output usable with `fload` and then introduces the syntax of the `fload` command itself.

3.1 Preparing PostScript output with dvips

To use `fload` with the `dvips` program (Rokicki) one must use a map file for `dvips` that specifies that PostScript fonts are *not* to be embedded into the PostScript output file. In particular, the map file must include only font names, and *no* file names, i.e., entries should be of the form (file `cmfontsr.map`):

```
cmr10 cmr10
and not (file cmfontsr.map):
cmr10 cmr10 <cmr10.pfb
```

since the latter specifies that the font should be input from the file `cmr10.pfb`. Note, however, that `subfont` must know where it can find the fonts it needs, so that, in this case, `cmr10` must be declared in the `ghostscript` font map (see step 2 in section 2.2).

3.2 Syntax of the fload command

`fload` is a C-shell script that parses (via the procedure `psfstat.ps` and the `ghostscript` interpreter) a legal PostScript file, determines which characters in which fonts are required but *not preloaded* in the file,¹ embeds (“loads”) the required characters from the fonts in question (using the `subfont` program), and writes them together with the input PostScript source file to the output file.

```
fload [-p fset] <PSf-in> [<PSf-out>]
```

`-p fset`

This option selects the font set that is resident in the printer, and from which the characters will be taken. The standard configuration comes with two font sets, namely `Standard` (the default) and `35`. The user can easily add more font sets (see section 2.2 step 4).

The list of all font sets installed at a given installation can be obtained by calling `fload` without parameters, as shown below.

```
[1] fload
Usage: (of the fload de Basil)
fload [-p fontSet] <ps-file> [<out-ps-file>]
```

¹ Partial font downloading with `fload` only works when the font files in question are *not* already embedded in the input file `<PSf-in>`.

where following fontSets are available
35
Standard

<PSf-in>

Input PostScript file that has to be scanned to define the characters to be embedded.

<PSf-out>

Output PostScript file containing the Type 1 definitions of the characters in the fonts referenced in the document. If <PSf-out> is not given the file is written to standard output.

If one has an invalid PostScript file, or one of the required fonts is not available, then a file with extension `flog` can be examined; it contains the transcript of parsing the source by `ghostscript`.

The `fload` procedure has been tested with PostScript output generated by `dvips` (see Appendices B to F, and figures 1 and 2), `FrameBuilder` (on Unix), and the MS Windows PostScript driver. Also, several kinds of fonts have been embedded, such as ATM compatible Type 1 fonts, Multiple master fonts,² and fonts with shareable sets of CharStrings. Note, however, that every new font must be tested to ensure that partial font embedding functions correctly.

4 Examples

As explained above to partially load only characters actually used in their document, (L^A)T_EX users should proceed in two stages:

1. run `dvips` on the `dvi` file using a map-file that does *not* embed the fonts inside in the PostScript output file (see section 3.1);
2. run `fload` on that PostScript file to embed the characters used inside the documents, so that the file becomes portable (see section 3.2).

Below we give several explicit examples.

4.1 CM fonts only

We start (Appendix B) by treating a simple one-page document (the French document described on page 333 of *The L^AT_EX Companion* (Goossens et al.), and reproduced also in figure 1). After running L^AT_EX we generate the PostScript file with `dvips` and include the header file `draft.ps` (via the `-h` command option) to put the word “DRAFT” across the page in the “funny” `cmff10` font (see figure 1 on page 77 for more details). In fact, we run `dvips` three times to show different ways of embedding the fonts. We start by using the default setup that

² Multiple master fonts are treated correctly only with `ghostscript` version 3.12 or higher.

includes the `pk` version of the fonts [1].³ Next [2], we embed the complete font inside the output file using the `-P` option (using the configuration file `config.CM` that includes `cmfonts.map`), and finally [3], we run with the “resident” version of the config file (`config.CMr` and its associated map file `cmfontsr.map`), that does not embed fonts (see section 3.1 for more about these `dvips` map files).

The listfile (`ls`) command [4] shows that the Type 1 fonts take up more than 90% of the file (that is normal for a one-page document, and not a representative number for “normal” situations). On the other hand, the `pk` versions of the fonts (at the default resolution of 300 dpi) take up about half the size of the file (31kb out of 59kb).

Now we are ready to extract the characters explicitly referenced from the Type 1 font files and embed them by running the `fload` utility [5], and another listfile [6] tells us that the Type 1 fonts now take up about 3/4 of the file.

Finally we have a look inside the intermediate file `babelfrer.fstat`, that the `fload` program writes (with the help of `ghostscript`) to see which characters in which fonts are used in the input PostScript file `babelfrer.ps`. One observes not only the presence of the “normal” L^AT_EX fonts, but also the `cmff10` font that was included via the header file `draft.ps`, and for which the five letters (D, F, A, R, and T) are all present in the list.

4.2 Using DC fonts

Appendix C deals with a L^AT_EX file similar to the one in appendix B, but we now use 8-bit characters in the input, the T1-encoding, and DC-fonts. The numbering of the commands corresponds to that in Appendix B. It is seen that, at the given resolution, the `pk` file and “reduced” Type 1 version are only marginally larger than with the CM fonts, while the full font embedded version is almost twice as large as its CM equivalent, which is as expected since a DC font contains about twice as many characters as its CM equivalent. The embedded characters are shown in the file `babelfre8r.fstat`. When one compares the entry for a CM font like `cmr10` in the file `babelfrer` in Appendix B with its DC equivalent `dcr10` in the file `babelfre8r.fstat` in Appendix C, one notes the occurrence of the pre-composed Eacute, agrave, eacute, etc., characters, that are absent from CM fonts, where they are constructed using the `\accent` primitive.

³ The numbers between square brackets in this section refer to the command numbers in the appendices.

4.3 DC fonts and some maths

The example in appendix D (it corresponds to figure 11.9 in *The L^AT_EX Companion*, German edition) also uses 8-bit input and DC-fonts, but contains also a lot of mathematics (see figure 2). In this case it is especially interesting to note the huge difference [3] between the three files without (`mathexar.ps`), with partial (`mathexadcr.ps`), and with complete font embedding (`mathexa.ps`). The contents of the file `mathexar.fstat` shows the actual characters that were embedded.

4.4 Real-life examples

To have a better idea of the usefulness of the `fload` procedure in real life, we have generated PostScript output files under the various conditions discussed earlier, but generating for a Linotype typesetter (at 1270 dpi), which corresponds more to a professional environment where the advantages of the approach proposed in this article are more evident.

Appendix E shows how a version of the present article's dvi-file (`fload.dvi`) was first generated with the `pk` fonts [1], embedding the full Type 1 fonts [2], and embedding no fonts [3]. The latter file was treated by `fload` [4], and the sizes of the various files can be seen by comparing the result of the `listfile` command [5]. It is seen that in this case the sizes of the `pk` and Type 1 version are more or less comparable, while the file treated with the `fload` procedure is more than 30% smaller than both of these. This means that partial font-embedding is especially important size-wise for medium sized documents, using a few characters in many fonts, and at professional resolutions. Note, however, that the (partial font loading) Type 1 solution is resolution-independent, so that the relative advantages should not be limited to the size-parameter only.

As a final test, we had a look at the $\mathcal{A}\mathcal{M}\mathcal{S}$ -distribution (American Mathematical Society), and treated two of their documents (`amsl doc.tex` and `testmath.tex`, see Appendix F). The output generated by the `dvips` command [1] shows the 29 CM fonts that are included. The `listfont` command [2] shows the difference between including complete Type 1 fonts (file `amsl doc.ps`) and embedding only the Type 1 code for the characters actually used (file `amsl docr.ps`). It is seen that the reduction is of the order of two in three. The next line [3] is even more informative, since it corresponds to a real article in the field of mathematics. The PostScript code for the article, without the fonts (file `testmathnone.ps`) is about 181 kbytes. When we add complete Type 1 fonts (file `testmath.ps`),

we end up with almost 1.6 Mbytes, while with the `fload` program, this is reduced to just under half a Mbyte (file `testmathr.ps`). This has to be compared to the situation using `pk`-fonts at a resolution of 1270 dpi (file `testmath1270pk.ps`), as normally used for $\mathcal{A}\mathcal{M}\mathcal{S}$ journals, where one needs just over 2 Mbytes.

5 Conclusion

From all of these examples, especially from the real-life ones in section 4.4, it is evident that substantial gains in file size can be obtained by using the `fload` system. This is in particular the case for high resolution typesetting, where `pk` font bitmaps become rather large, while the size of the (reduced) Type 1 fonts remain identical (apart from small resolution-dependent differences in positioning introduced by the `dvips` program). This reduction in size is also important if one wants to export the PostScript file for use in other applications, like Adobe Acrobat, where Type 1 fonts are gracefully handled, while support for Type 3 fonts is not very good. It would be helpful if this kind of partial font embedding could become somehow more closely integrated with the dvi driver itself.

References

- Aladdin Software. *GhostScript version 3.33 interpreter/previewer*. Electronically distributed document, 1994.
- American Mathematical Society. *$\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX Version 1.2 User's Guide and Sample Paper for the `amsmath` Package*. Electronically distributed documents, 1995.
- Michel Goossens, Frank Mittelbach and Alexander Samarin. *The L^AT_EX Companion*. Addison Wesley, 1994.
- Tomas Rokicki. *Dvips: A DVI-to-PostScript Translator*. Electronically distributed document, 1994.

Appendices

A Syntax of the subfont program

The `subfont` program parses an Adobe Type 1 font program (in `pfa` (hexadecimal) or `pfb` (binary) format), extracts the specified characters, and writes a new font containing only those characters. When working with a single font it outputs a font in a `pfa` or `pfb` `atm` compatible format. When working in multi-font mode (the default or chosen with the `-f` option) the output is written in a form that can only be used by a complete PostScript interpreter (not by `atm`). Moreover, in multi-font mode, fonts with the same encoding share one encoding vector,

the `eexec` PostScript operator is not used in the output stream, and several other optimizations are performed.

The syntax of the `subfont` program is:

```
subfont command-list <file>
```

If instead of a file name `<file>`, one uses a hyphen (-), then the input Type 1 font is read on standard input. This is useful for using `subfont` in command pipes.

`-v` Run in *verbose* mode, i.e., output information during the parsing stage.

`-o<file>`

Write the output to file `<file>`. By default, the output is written to `stdout`, from where it can be easily redirected.

`-a<file>`

Similar to `-o`, but the output file is opened in append mode. This can be useful for adding a prolog.

`-b<file>`

Write the output file in binary mode. This only works in single-font mode, where it writes a `pfb` file. If used in multi-font mode, the program will stop when one tries to load the second font.

`-c<file>`

Copy the file from input to output (as-is). Can be useful on VMS.

`-n/./././`

This option specifies the list of characters to be included. If it is absent, all characters are included. The result will be to convert a `pfb` to a `pfa` file, or the reverse (see an example below).

`-f <file>`

Reads the list of the required characters and fonts from the file specified and processes each font in turn. This option activates multi-font mode. When fonts have an identical encoding they will share their encoding vectors. The `eexec` operator will not be used in the output stream and other optimizations will be performed. Every line of this file must contain a specification of the type:

```
font-name/char-1/char-2/.../char.../
```

Examples can be seen in appendices B, C, and D, in the listings of the files with extension `fstat`.

`-m<map-file>`

Specifies the name of the file containing the mapping between the font names, as used in the file processed by the `-f` option, and the files containing the fonts in question. The format of a map file is described in section 2.2, point 2.

`-I<path>`

The font search path used for searching the font files (used with the `-f` option).

`-u<file>`

The name of the file that will contain the list of fonts that are undefined. This file will be created only in case unknown fonts exist (used with `-f` option).

`-e#` Touch encoding flag. In multi-font mode identical font encoding vectors are loaded only once. To do this, every encoding vector is loaded out of the font body and must have a unique name and every font must only refer to an already loaded encoding vector.

To turn off this feature one should use the option `-e-1`. When the program prepares several files without the `-f` option but using the `-n` option and `<file>`, the encoding vector will be loaded as-is.

To force removing identical encodings one can say `-e1`.

`-x#` `eexec` applying flag. In multi-font mode fonts will be output without the `eexec` instruction. This feature makes output files smaller.

To suppress this feature in multi-font mode one can use the option `-x2`. When the program prepares several files without the `-f` option but using the `-n` option and `<file>`, then `eexec` will be applied. In this case, to remove `eexec` one can use `-x0`.

`-z <file1> ...`

Read every file in the list `<file1> ...` and for each line with the pattern:

```
/FontName /<font-name> def
```

output a line like:

```
/<fontname> (<file-name>);
```

This facility is useful for making font map files and is actually used by the `mkmap` utility described in section 2.2 (point 2).

As an example, one can create a font containing only the characters “T”, “E”, and “X” from the Courier font, by specifying:

```
subfont -b TEX.pfb -n/T/E/X/ cour.pfb
```

where the compressed binary `pfb` format is used.

One can select the alphanumeric output format by typing:

```
subfont -o TEX.pfa -n/T/E/X/ cour.pfb
```

If one wants a representation where the character outline code is not encoded with `eexec`, one can use the `-x0` option:

```
subfont -x0 -o TEX.pfa -n/T/E/X/ cour.pfb
```

B Example of using fload with the CM fonts

```
[1] dvips -hdraft.ps babelfre -o babelfrecm.ps
This is dvipsk 5.58e Copyright 1986, 1994 Radical Eye Software
' TeX output 1995.01.29:1928' -> babelfrecm.ps
<draft.ps><tex.pro><special.pro>. [1<colorcir.eps><tac2dim.eps>]
[2] dvips -hdraft.ps -PCM babelfre -o babelfre.ps
This is dvipsk 5.58e Copyright 1986, 1994 Radical Eye Software
' TeX output 1995.01.29:1928' -> babelfre.ps
<draft.ps><tex.pro><cmr12.pfb><cmbx12.pfb><cmbx10.pfb><cmr10.pfb><cmmi10.pfb>
<cmtt10.pfb><cmr7.pfb><cmti10.pfb><texp.s.pro><special.pro>. [1<colorcir.eps>
<tac2dim.eps>]
[3] dvips -hdraft.ps -PCMr babelfre -o babelfrer.ps
This is dvipsk 5.58e Copyright 1986, 1994 Radical Eye Software
' TeX output 1995.01.29:1928' -> babelfrer.ps
<draft.ps><tex.pro><texp.s.pro><special.pro>. [1<colorcir.eps><tac2dim.eps>]
[4] ls -l babelfre*.ps
-rw-r--r--  1 goossens et          408759 Jan 29 19:37 babelfre.ps
-rw-r--r--  1 goossens et          58818 Jan 29 19:37 babelfrecm.ps
-rw-r--r--  1 goossens et          28209 Jan 29 19:35 babelfrer.ps
[5] fload babelfrer.ps > babelfrerr.ps
Process PS file babelfrer.ps.
Write font using statistic to babelfrer.fstat
Process font 'cmti10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmti10.pfb'
Process font 'cmff10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmff10.pfb'
Process font 'cmr10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmr10.pfb'
Process font 'cmbx10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmbx10.pfb'
Process font 'cmr12' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmr12.pfb'
Process font 'cmtt10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmtt10.pfb'
Process font 'cmmi10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmmi10.pfb'
Process font 'cmr7' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmr7.pfb'
Process font 'cmbx12' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmbx12.pfb'
Total of 10 files are processed.
[6] ls -l babelfrerr.ps
-rw-r--r--  1 goossens et          104071 Jan 29 19:35 babelfrerr.ps

----- FILE babelfrer.fstat -----
cmti10/parenright/t/i/S/m/o/d/parenleft/f/s/u/comma/P/l/a/n/two/c/p/e/r/acute/
cmff10/D/F/A/R/T/
cmr10/parenright/bracketright/g/t/i/v/D/Q/colon/x/F/S/m/one/fi/o/b/three/d/q/
grave/parenleft/L/f/s/A/u/comma/C/P/nine/period/E/l/T/a/n/two/I/c/p/
quoteright/four/bracketleft/X/e/r/acute/
cmbx10/g/t/x/S/m/one/U/fi/b/d/u/P/E/l/a/n/two/p/quoteright/e/r/
cmr12/t/i/v/x/m/cedilla/one/d/five/f/s/u/nine/j/E/l/a/n/two/c/p/quoteright/e/r/
cmtt10/g/t/i/braceleft/braceright/m/b/o/three/d/q/backslash/f/s/h/u/comma/w/
period/l/equal/a/n/c/p/e/r/
cmmi10/period/
cmr7/o/A/e/
cmbx12/g/t/i/x/S/m/one/U/b/fi/d/grave/L/f/s/u/P/E/R/l/T/a/n/two/I/c/p/quoteright/e/r/acute/
```

C Example of using fload with the DC fonts

```
[1] dvips -hdraft.ps babelfre8 -o
This is dvipsk 5.58e Copyright 1986, 1994 Radical Eye Software
' TeX output 1995.01.29:1941' -> babelfre8.ps
<draft.ps><tex.pro><special.pro>. [1<colorcir.eps><tac2dim.eps>]
[2] dvips -hdraft.ps -PDC babelfre8 -o babelfre8dc.ps
This is dvipsk 5.58e Copyright 1986, 1994 Radical Eye Software
' TeX output 1995.01.29:1941' -> babelfre8dc.ps
<draft.ps><tex.pro><dcr12.pfb><dcbx12.pfb><dcbx10.pfb><dcr10.pfb><cmmi10.pfb>
```

```

<dctt10.pfb><dcr7.pfb><cmr10.pfb><cmr7.pfb><dcti10.pfb><texps.pro>
<special.pro>. [1<colorcir.eps><tac2dim.eps>]
[3] dvips -hdraft.ps -PDCr babelfre8 -o babelfre8r.ps
This is dvipsk 5.58e Copyright 1986, 1994 Radical Eye Software
' TeX output 1995.01.29:1941' -> babelfre8r.ps
<draft.ps><tex.pro><texps.pro><special.pro>. [1<colorcir.eps><tac2dim.eps>]
[4] ls -l babelfre8*.ps
-rw-r--r--  1 goossens et          60161 Jan 29 19:41 babelfre8.ps
-rw-r--r--  1 goossens et        720063 Jan 29 19:41 babelfre8dc.ps
-rw-r--r--  1 goossens et         28151 Jan 29 19:41 babelfre8r.ps
[5] fload babelfre8r.ps > babelfre8dcr.ps
Process PS file babelfre8r.ps.
Write font using statistic to babelfre8r.fstat
Process font 'dcr10' from file '/usr/local/lib/texmf/fonts/./dc/type1/dcr10.pfb'
Process font 'dcbx12' from file '/usr/local/lib/texmf/fonts/./dc/type1/dcbx12.pfb'
Process font 'dcti10' from file '/usr/local/lib/texmf/fonts/./dc/type1/dcti10.pfb'
Process font 'cmff10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmff10.pfb'
Process font 'cmr10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmr10.pfb'
Process font 'dctt10' from file '/usr/local/lib/texmf/fonts/./dc/type1/dctt10.pfb'
Process font 'dcbx10' from file '/usr/local/lib/texmf/fonts/./dc/type1/dcbx10.pfb'
Process font 'dcr12' from file '/usr/local/lib/texmf/fonts/./dc/type1/dcr12.pfb'
Process font 'dcr7' from file '/usr/local/lib/texmf/fonts/./dc/type1/dcr7.pfb'
Process font 'cmmi10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmmi10.pfb'
Process font 'cmr7' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmr7.pfb'
Total of 12 files are processed.
[6] ls -l babelfre8dcr.ps
-rw-r--r--  1 goossens et        114515 Jan 29 19:42 babelfre8dcr.ps

----- FILE babelfre8r.fstat -----
dcr10/Eacute/agrave/eacute/parenright/bracketright/g/t/i/v/D/Q/colon/x/F/S/m/
  one/fi/o/b/d/q/parenleft/L/f/s/A/u/comma/C/P/nine/period/E/l/T/a/n/two/
  I/c/p/quoteright/bracketleft/X/e/r/
dcbx12/egrave/eacute/g/t/i/x/S/m/one/U/b/fi/d/L/f/s/u/P/E/R/l/T/a/n/two/I/c/p/quoteright/e/r/
dcti10/eacute/parenright/t/i/S/m/o/d/parenleft/f/s/u/comma/P/l/a/n/two/c/p/e/r/
cmff10/D/F/A/R/T/
cmr10/one/three/two/four/
dctt10/g/t/i/braceleft/braceright/m/b/o/three/d/q/backslash/f/s/h/u/comma/w/
  period/l/equal/a/n/c/p/e/r/
dcbx10/g/t/x/S/m/one/U/fi/b/d/u/P/E/l/a/n/two/p/quoteright/e/r/
dcr12/ccedilla/t/i/v/x/m/one/d/five/f/s/u/nine/j/E/l/a/n/two/c/p/quoteright/e/r/
dcr7/A/
cmmi10/period/
cmr7/o/e/

```

D Example of using fload with a lot of mathematics

```

[21] fload mathexar.ps > mathexadcr.ps
Process PS file mathexar.ps.
Write font using statistic to mathexar.fstat
Process font 'cmmi7' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmmi7.pfb'
Process font 'dcr10' from file '/usr/local/lib/texmf/fonts/./dc/type1/dcr10.pfb'
Process font 'dcbx12' from file '/usr/local/lib/texmf/fonts/./dc/type1/dcbx12.pfb'
Process font 'cmr10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmr10.pfb'
Process font 'cmsy7' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmsy7.pfb'
Process font 'dctt10' from file '/usr/local/lib/texmf/fonts/./dc/type1/dctt10.pfb'
Process font 'cmex10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmex10.pfb'
Process font 'cmsy10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmsy10.pfb'
Process font 'cmmi10' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmmi10.pfb'
Process font 'cmr7' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmr7.pfb'

```

Process font 'cmmi5' from file '/usr/local/lib/texmf/fonts/./cm/type1/cmmi5.pfb'

Total of 12 files are processed.

```
[3] ls -l mathexa*.ps
-rw-r--r--  1 goossens et          640749 Jan 29 19:12 mathexa.ps
-rw-r--r--  1 goossens et          87363 Jan 29 20:07 mathexadcr.ps
-rw-r--r--  1 goossens et          8748 Jan 29 19:13 mathexar.ps
```

----- FILE matheaxr.fstat -----

```
cmmi7/gamma/k/A/l/n/
dcr10/udieresis/adieresis/germandbls/parenright/Z/g/t/B/i/v/hyphen/D/k/x/F/m/z/one/
  U/b/o/three/d/parenleft/f/s/A/N/h/u/comma/w/period/l/T/a/n/two/I/c/four/K/e/r/
dcbx12/udieresis/M/g/t/B/i/hyphen/k/F/S/m/one/o/three/d/f/s/A/h/u/l/a/n/two/c/p/e/r/
cmr10/parenright/six/plus/one/parenleft/equal/two/four/
cmsy7/minus/element/
dctt10/parenright/bracketright/g/t/i/m/b/o/d/q/parenleft/backslash/f/s/h/u/w/l/zero/
  a/n/c/p/bracketleft/e/r/
cmex10/parenrightbig/parenleftbigg/integraltext/summationdisplay/parenleftbig/
  integraldisplay/parenrightbigg/
cmsy10/minus/periodcentered/
cmmi10/k/x/z/b/d/f/comma/w/l/y/a/I/
cmr7/Gamma/one/three/two/four/
cmmi5/C/
```

E Using fload on the present article

```
[1] dvips -D1270 fload -ofload1270pk.ps
[2] dvips -D1270 -PCM fload -ofload1270cm.ps
[3] dvips -D1270 -PCMr fload -ofload1270cmr.ps
[4] fload fload1270cmr.ps fload1270cmp.ps
[5] ls fload1270*.ps
-rw-r--r--  1 goossens et          878948 Feb 12 16:26 fload1270cm.ps
-rw-r--r--  1 goossens et          501927 Feb 12 16:26 fload1270cmp.ps
-rw-r--r--  1 goossens et          262556 Feb 12 16:26 fload1270cmr.ps
-rw-r--r--  1 goossens et          838147 Feb 12 16:25 fload1270pk.ps
```

F Using fload with some $\mathcal{A}\mathcal{M}\mathcal{S}$ documents

```
[1] dvips -PCM -PAM amsldoc -o
This is dvipsk 5.58e Copyright 1986, 1994 Radical Eye Software
' TeX output 1995.02.16:1008' -> amsldoc.ps
<tex.pro><cmsy10.pfb><cmr17.pfb><cmr12.pfb><cmr10.pfb><cmr8.pfb><cmbx12.pfb>
<cmbx10.pfb><cmbsy10.pfb><cmbx7.pfb><cmr7.pfb><cmmi10.pfb><cmtt10.pfb>
<cmsy8.pfb><cmr6.pfb><cmbx8.pfb><cmti10.pfb><cmex10.pfb><cmtt8.pfb>
<cmtt12.pfb><cmmi7.pfb><cmsy7.pfb><cmr5.pfb><cmmi5.pfb><cmti8.pfb><cmsy5.pfb>
<cmmbi10.pfb><cmbsy7.pfb><cmti7.pfb><cmmi8.pfb><texps.pro>. [1] [2] [3] [4]
[5] [6] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]
[16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30]
[31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43]
[2] ls -l amsldoc*.ps
-rw-r--r--  1 goossens et          488335 Feb 16 10:12 amsldocr.ps
-rw-r--r--  1 goossens et          1639085 Feb 16 10:13 amsldoc.ps
[3] ls -l testmath*.ps
-rw-r--r--  1 goossens et          1578903 Feb 16 10:29 testmath.ps
-rw-r--r--  1 goossens et          2032195 Feb 16 10:32 testmath1270pk.ps
-rw-r--r--  1 goossens et          181349 Feb 16 10:26 testmathnone.ps
-rw-r--r--  1 goossens et          480525 Feb 16 10:27 testmathr.ps
```


Exemple d'un article en français
29 janvier 1995

Table des matières

1 Une figure EPS	1
2 Exemple d'un tableau	1

Liste des figures

1 Deux images EPS	1
-------------------	---

Liste des tableaux

1 Quelques commandes de l'option french de babel	1
--	---

1 Une figure EPS

Cette section montre comment inclure une figure PostScript[1] dans un document L^AT_EX. La figure 1 est insérée dans le texte à l'aide de la commande `\epsfig{file=colcircr.eps,width=3cm}`.

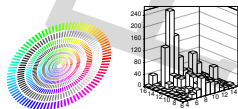


Figure 1: Deux images EPS

2 Exemple d'un tableau

Le tableau 1 à la page 1 montre l'utilisation de l'environnement `table`.

```
\primo 1° \secundo 2° \tertio 3° \quatro 4° 2ème 2°
```

Tableau 1: Quelques commandes de l'option french de babel

Références

[1] Adobe Inc. *PostScript, manuel de référence (2^e édition)* InterEditions (France), 1992

Index

figure, 1	PostScript, 1	section, 1
index, 1	références, 1	tableau, 1

1

Figure 1: Example of page in French

In the figure above the word “DRAFT” was produced by using the following `draft.ps` header file with `dvips`:

```
% Prints "DRAFT" diagonally across the page
/bop-hook{gsave
  200 30 translate 65 rotate
  /cmff10 findfont 250 scalefont
  setfont 0 0 moveto
  0.85 setgray (DRAFT) show
  grestore} userdict begin def end
```

Note the use of the font `cmff10`. The presence of the five characters (corresponding to the word “DRAFT”) of this font is picked up correctly by the font analysis step, as seen in Appendix B in the listing of the contents of the file `babelref.stat`:

```
cmff10/D/F/A/R/T/
```

1 Mehrfachintegrale

`\iiint` und `\iiiiint` erzeugen drei und vier Integralzeichen mit gleichmäßigem Zwischenraum, und zwar sowohl im normalen Text, wie z.B. $\iiint_A f(x, y, z) dx dy dz$ und $\iiiiint_A f(w, x, y, z) dw dx dy dz$, als auch in abgesetzten Formeln.

$$\iiint_A f(x, y, z) dx dy dz \quad (1)$$

$$\iiiiint_A f(w, x, y, z) dw dx dy dz \quad (2)$$

2 Binomische Ausdrücke

Für binomische Ausdrücke, wie etwa $\binom{n}{k}$ gibt es die Befehle `\binom`, `\dbinom` und `\tbinom`. `\binom` ist eine Kurzform für `\fracwithdelimsO{Opt}`.

$$\begin{aligned} \sum_{\gamma \in \Gamma^c} I_\gamma &= 2^k - \binom{k}{1} 2^{k-1} + \binom{k}{2} 2^{k-2} \\ &+ \dots + (-1)^l \binom{k}{l} 2^{k-l} + \dots + (-1)^k \\ &= (2-1)^k = 1 \end{aligned} \quad (3)$$

3 Split-Formeln

Die `split`-Umgebung erzeugt keine Numerierung, da sie nur innerhalb anderer abgesetzter Formelumgebungen verwendet werden kann, wie etwa `equation`, `align` oder `gather`. Die Formelnummer wird dann von der äußeren Umgebung erzeugt.

$$\begin{aligned} (a+b)^4 &= (a+b)^2(a+b)^2 \\ &= (a^2+2ab+b^2)(a^2+2ab+b^2) \\ &= a^4+4a^3b+6a^2b^2+4ab^3+b^4 \end{aligned} \quad (4)$$

Figure 2: `dvips` output (DC and CM math fonts)

Note added in Proof

Sergey Lesenko has implemented partial embedding of Type 1 font sources into PostScript files within the framework of Tom Rokicki’s `dvips` program. This work is presented in an article in the proceedings of the 1995 TUG Conference in St. Petersburg (*TUGboat* 16(3), to be published). Lysenko is working closely with Tom Rokicki to integrate his code into the next major release of `dvips`, so that it is anticipated that for T_EX users, partial font embedding will be a trivial task in the near future.

- ◇ Basil K. Malyshev
IHEP, Protvino, Russia
Email: goossens@cern.ch
- ◇ Michel Goossens
CERN, Geneva, Switzerland
Email: malyshev@desert.ihep.su