DVIxxx - Display TEX DVI Files on Assorted Output Devices

NAME

DVIxxx — TeX DVI to device xxx translator family

SYNOPSIS

dvixxx [-a] [-b] [-c#] [-d#] [-e *VAR=value*] **[-f** *fontsubfile*]
[-l] [-m#] [-o#] [-o#:#] [-o#:#:#] [-p] [-q] [-r#] [-s#] [-v]
[-x# *units*] **[-y#** *units*] **[-z] dvifile1 [dvifile2]** ...
xxx = output device identifier suffix (see below)

DESCRIPTION

Several TEX DVI translators are available. They all expect the name of the DVI file on the command line, and the extension `.dvi` can always be omitted. As illustrated below, they issue a one-line identifier message and, if no command line arguments are given, type a Unix-style[1] *usage* message. Some of them may have additional help files. On case-sensitive file systems, file names may be expected to be entirely in lower case, so you should type `dvialw` instead of `DVIALW`.

For all except DVIBIT (which is intended for interactive display), the output file will be given the name of the `.dvi` file, but extension `.dvi-xxx`, where `xxx` is the three-character mnemonic for the translator program. If long extensions are not supported, then `.xxx` is used. For DVIBIT, output is on `stdout` which defaults to the terminal; it may be redirected in the usual Unix fashion by `>filename` on the command line (e.g. `dvibit foo >foo.out`).

[1] Unix is a trademark of AT&T Bell Laboratories. PC-DOS is a trademark of IBM Corporation. TEX is a trademark of the American Mathematical Society. TOPS-20, VAX, and VMS are trademarks of Digital Equipment Corporation.

As each `.dvi` file is processed, a list of errors is printed on the standard error unit `stderr`; this list is also saved in a file with extension `.dvi-err`, or if long extensions are not supported by the host, then extension `.err` is used. This file is not created if there are no errors. As each page is printed, the physical page number and the TeX page number(s) are printed without a following carriage return; after the last page, the string `[OK]` is printed, followed by a newline. This gives a convenient progress report to the terminal. If it is not wanted, then the error output can be redirected into a file (possibly the null device) (e.g. `dvixxx foo &foo.err`), or the `-q` (quiet) option can be given to suppress it.

These drivers are written in C, and with C preprocessor conditional compilation features, are all derived from one master set of files, so that there is substantial code sharing among them. Host machine and output device dependencies are parametrized to allow easy movement to new hosts and new output devices. Implementations now exist on Gould Unix, Hewlett-Packard Unix, PC-DOS, TOPS-20, VAX Unix, and VAX VMS, with others in progress.

DEVICES SUPPORTED

The available translators are as follows:

DVIALW PostScript (Apple LaserWriter)

DVIBIT Version 3.10 BBN BitGraph terminal

DVICAN Canon LBP-8 A2 laser printer

DVIGD Golden Dawn Golden Laser 100 printer

DVIIMP Imagen imPRESS-language laser printer family

DVIJEP Hewlett-Packard LaserJet Plus

DVIJET Hewlett-Packard LaserJet

DVIL3P DEC LN03 Plus laser printer

DVIL75 DEC LA75 144 dpi printer

DVIM72 Apple Imagewriter 72 dpi printer

DVIMAC Apple Imagewriter 144 dpi printer

DVIMPI MPI Sprinter 72 dpi printer

DVIO72 OKIDATA Pacemark 2410 72 dpi printer

DVIOKI OKIDATA Pacemark 2410 144 dpi printer

DVIPRX Printronix 60h $\times$ 72v dpi printer

DVITOS Toshiba P-1351 180 dpi printer

DVITYP or DVITYPE
DVI Translator for human-readable output

OPTIONS

The order of command options and DVI file names is *not* significant; all switch values apply to all DVI files. DVI files are processed in order from left to right.

Letter case is *ignored* in option switches: `-A` and `-a` are equivalent.

`-a` Implement virtual font caching, if possible. When a font file is opened, a buffer is allocated to contain the entire file, and the file is then read with one system call. This is important primarily on networked file systems, where the many random-access calls in the font file for small amounts of data entail substantial network overhead. With the entire file cached in local memory, this overhead is removed. The additional memory required for the font file buffers amounts to 100K to 200K bytes (assuming the compact `.PK` font file format), which is not excessive. If memory cannot be allocated for a font file, then normal buffering of small blocks is used. A trace option (`-d64`) is provided to monitor the font caching; see below.

`-b` Backwards order printing from the default. For example, laser printers using the Canon LBP-CX print engine normally receive pages in reverse order because they stack printed side up. Some have page handling mechanisms that stack them face down, and in such a case `-b` will ensure that they come out in order $1, 2, \ldots$ instead of $n, n-1, n-2, \ldots$

`-c#` Print # copies of each output page. Page copies are printed consecutively; this does *not* give multiple collated copies of the entire job.

`-d#` Produce debugging output on `stderr` if a non-zero value is given. Multiple `-d` switches may be specified, and one may also add values of the following possible options to obtain the switch value:

1 (DVIJET only) print page bitmap in hexadecimal;

2 display page coordinates and metrics of each output character, and print each character bitmap in hexadecimal;

4 (DVIJEP only) display updated page coordinate of each character after each call to `fixpos()`;

8 print filename and open mode of each *successful* file opening;

16 print filename and open mode of each *unsuccessful* file opening;

32 show discarded off-page text;

64 trace virtual font caching;

128 trace character setting (*lots* of output).

For example, either `-d8 -d16` or `-d24` will trace all attempted file openings.

`-eVAR=value`

Define an environment variable on the command line (see the later section *Environment Variables*). The acceptable values for `VAR` are `DVIHELP`, `FONTLIST`, `TEXFONTS`, and `TEXINPUTS`. Under normal use of the translators, these can be set by TOPS-20 and VAX VMS `define VAR: value` commands, or by Unix CSH `setenv VAR value` or SH `VAR=value` commands. When the translator is invoked by another program, such as a print spooler, on some systems it may not be possible to set a particular value of an environment variable for the subprocess, so this option gets around this limitation. On most Unix systems, it should be possible to use the call `system("VAR=value; dvixxx filename")`.

`-ffontsubfile`

Define an alternate font substitution file which is to be used instead of the default ones (see below).

`-l` Inhibit logging.

`-m#` Reset magnification to #. The default for low resolution printers is `-m603`, corresponding to $1/1.2^5$ magnification of 300-dot/inch fonts. By

TeX conventions, magnification 1000 corresponds to a 200-dot/inch output device. The default magnification is always adjusted according to the output device resolution in order to give a normal page size, so this parameter should rarely be required. Legal values are

$$int((1000 \text{ or } 1440 \text{ or } 1500) \times 1.2^{k/2}) \quad (k = -16 \ldots 16);$$

other values will be set to the nearest in this family. Not all fonts will be available in this wide range, and most installations will probably have only a half dozen or so magnifications.

Magnification values less than 25 are taken to be a TeX magstep parameter which is applied to the standard magnification for that device. For example, `-m-0.5` selects a smaller size, and `-m2` selects a size 1.44 times larger than normal.

`-o#` or `-o#:#` or `-o#:#:#`

Specify a page number, or range of page numbers, to be selected for output. In the third form, the last number is the page number step size; it is normally 1. This option may be specified any number of times. If it is not specified, then all pages will be printed. Pages are numbered in order $1, 2, 3, \ldots$ in the file, but any page number recorded by TeX on the printed page will in general be different. Negative page numbers count backward; -1 is the last page in the document, -2 the second last page, and so on.

As pages are selected for printing, `[#{#}` will be printed on `stderr`, where the first `#` is the page number in the file, and the second `#` is a string of values of the TeX counters `\count0` through `\count9`, separated by dots, with trailing zero counters dropped. `\count0` usually records the printed page number. When the page is completely output, a closing `]` will be printed on `stderr`. Any error messages from processing of that page will therefore occur between the square brackets. For example, `-o1:3 -o12 -o17:23 -o-3:-1` would select pages 1, 2, 3, 12, 17, 18, 19, 20, 21, 22, and 23, plus the last three pages.

Pages are processed in the order found in the DVI file; there is intentionally no attempt made to sort them according the `\count0` values, since different macro packages may use this counter for different purposes, and in the case of floating tables and figures, the pages may not be in order anyway.

Pages will always be printed in an order appropriate for the device so that the first document page occurs first face up in the document stack; the `-b` option can be used to reverse this order. For example, some Hewlett-Packard LaserJet Plus printers are equipped with a page flipper which stacks output face down; for these, the `-b` option will ensure that the pages come out in the expected order.

Specification of a page number step size is useful for producing duplex (two-sided) printing. For example, with laser printers using the Canon LBP-CX engine, the first run could specify `-o1:9999:2`, which would stack output face up, beginning with the last page, and ending with page 1 on top. The printed pages can then be reinserted in the input tray *face up*, page 1 on the top, exactly as they were found in the output tray, with the top of the page in the tray closest to the end which is inserted first into the printer. A second run with `-b -o2:9999:2` would then print pages 2, 4, ..., on the backs of pages 1, 3, ...; note the `-b` option to get backwards order on the second run.

There is a bug in Microsoft C's `sscanf()` on the IBM PC; it does not correctly parse input on the format `"%d:%d:%d"` in `option()` for the page number switch. It correctly returns the numbers, but instead of returning the number of such items parsed, it returns -1, which should only happen if none are parsed. A work around seems to be to supply a trailing colon on the switch, so that you write `-o17:` instead of `-o17`.

`-p` Inhibit font preloading. This may produce output a few seconds earlier when all pages are output, but should have negligible effect on the execution time, and consequently, should normally not be specified. When individual pages are being printed with the `-o#` option, preloading is necessary (and will be forced) to ensure that all fonts are defined before they are referenced.

`-q` Quiet mode. Status displays to `stderr` are suppressed, unless warning or error messages are issued. For interactive devices (DVIBIT), warning messages are suppressed.

`-r#` (Device = HP LaserJet only). Specify the Laser Jet output resolution in dots per inch. `#` must be one of 75, 100, 150, or 300. The actual plot file is identical in each case; only the size on the output page is changed, because the resolution change is effected by printing 1×1, 2×2, 3×3, or 4×4 pixel blocks.

`-r` (Device = Golden Laser 100 only). Select run-length encoding of the output file. This reduces disk space typically by 10% to 40%, but increases host CPU time for the preparation of the output file.

`-r` (Device = Apple ImageWriter only). Select run-length encoding of the output file.

`-r` (Device = Toshiba P-1351 only). Select run-length encoding of the output file. This reduces disk space typically by 10% to 40%, but increases host CPU time for the preparation of the output file, and because of poor logic in the printer, may double the print time! The print quality is also substantially worse, so this option is generally *not* recommended.

`-s#` (Device = Apple LaserWriter only). Force characters larger than # pixels wide or high to be reloaded each time they are required. The Version 23.0 PostScript interpreter has a bug which manifests itself in fatal `VM error` messages when large characters are sent. A reasonable default value has been set for this which should normally avoid the problem. Specifying `-s0` will cause reloading of every character each time it is used.

`-v` (Device = Apple LaserWriter only). Force reloading of all required fonts at start of each page.

`-x#`*units*

The `-x` options specify the left margin of the TEX page on the output page in any of several units. Letter case is not significant in the units field, which must *not* be separated from the number by any space. `#` may be fractional. For example, `-x1.0in`, `-x2.54cm`, `-x72.27pt`, and `-x6.0225pc` all specify a one-inch left margin. Negative values are permissible, and may be used to shift the output page left (possibly truncating it on the left) in order to display a wide TEX page.

The units field is mandatory, and may be one of

`bp` big point ($1in = 72bp$)

`cc` cicero ($1cc = 12dd$)

`cm` centimeter ($1in = 2.54cm$)

`dd` didot point ($1157dd = 1238pt$)

`in` inch

- `mm` millimeter ($10mm = 1cm$)
- `pc` pica ($1pc = 12pt$)
- `pt` point ($72.27pt = 1in$)
- `sp` scaled point ($65536sp = 1pt$)

`-y#`*units*

The `-y` options specify the top margin of the TEX page on the output page in any of the indicated units. Letter case is not significant in the unit field, which must *not* be separated from the number by any space. `#` may be fractional. For example, `-y1.0in`, `-y2.54cm`, `-y72.27pt`, and `-y6.0225pc` all specify a one-inch top margin. Negative values are permissible, and may be used to shift the output page up (possibly truncating it on the top) in order to display a long TEX page.

By decree of the Stanford TEX Project, the default TEX page origin is always 1 inch over and down from the top-left page corner, even when non-American paper sizes are used. This corresponds to the switch settings `-x1in -y1in`; these values are assumed unless overridden.

`-z` (TOPS-20 or 4.xBSD Unix only). For each DVI file processed, type in an EXEC command `DVISPOOL: dvifilename` (on Unix, `DVISPOOL dvifilename`) followed by a newline; the user may then define `DVISPOOL:` (or `DVISPOOL`) to be a program or shell script which sends the translation of the DVI file to the appropriate output spooler.

SAMPLE EXECUTION

Here is a sample execution of LATEX and DVIALW extracted from a TOPS-20 PHOTO log:

```
@latex biblio.ltx
This is TeX, Tops-20 Version 1.1 (preloaded format=lplain 84.9.29)
(APS:<BEEBE.PLOT79.DOCUMENTATION>BIBLIO.LTX.28
LaTeX Version 2.06a - Release 7 July 84
(APS:<TEX.LATEX>REPORT.STY.2
Document Style 'report'. Version 0.91 - released 25 June 1984
(APS:<TEX.LATEX>REP11.STY.2))
(APS:<BEEBE.PLOT79.DOCUMENTATION>MYBIBLIO.STY.1 Mybibliography
environment style - Version 0.0 - 15-May-86)
(APS:<BEEBE.PLOT79.DOCUMENTATION>BIBLIO.AUX.12) [0]
(APS:<BEEBE.PLOT79.DOCUMENTATION>BIBLIO1.LTX.3 [1] [2] [3] [4]
```

```
[5]) [6]
(APS:<BEEBE.PLOT79.DOCUMENTATION>BIBLIO.AUX.13)
(see the transcript file for additional information)
Output written on APS:<BEEBE.PLOT79.DOCUMENTATION>BIBLIO.DVI.1
(7 pages, 13960 bytes).
Transcript written on APS:<BEEBE.PLOT79.DOCUMENTATION>BIBLIO.LST.1.

@dvialw -x0.3in -y0.2in biblio bt:example
[TeX82 DVI Translator Version 2.0 for PostScript [Apple LaserWriter
laser printer]]
[Input from DVI file biblio.dvi]
[Output on file biblio.dvi-alw]
[7 pages]
[1500 magnification]
[7{6}] [6{5}] [5{4}] [4{3}] [3{2}] [2{1}] [1{0}]  [OK]
[Input from DVI file bt:example.dvi]
[Output on file bt:example.dvi-alw]
[1 pages]
[1500 magnification]
[1{1}]  [OK]
```

When the TOPS-20 version of TEX finishes execution, it normally simulates terminal input of a line of the form

```
TeXspool: dvifile
```

without supplying a final carriage return. The default value of the logical name `TeXspool:` points to a dummy program which does nothing, so if you just type a carriage return yourself, the line is effectively ignored. This is reasonable in that it usually takes several trips through TEX before you have a `.dvi` file worth printing. If you like, you can redefine `TeXspool:` to point to your favorite DVI translator, for example,

```
define TeXspool: sys:dvialw.exe
```

Then when you type a carriage return when TEX finishes, it will run the translator immediately, saving you a line of typing. If you do not want the translator to run, just cancel the line by typing `CTL-U` or `CTL-C`.

A sample invocation of DVITYPE is as follows:

```
@dvitype
DVIFILE    : story.dvi
OUTPUT     : tty:
```

```
This is DVItype, Tops-20 Version 2.8
Output level (default=3, ? for help):
Starting page (default=*):
Maximum number of pages (default=1000000):
Assumed device resolution in pixels per inch (default=300/1):
New magnification (default=0 to keep the old one):
Options selected:
  Starting page = *
  Maximum number of pages = 1000000
  Output level = 3 (the works)
  Resolution =  300.00000000 pixels per inch
numerator/denominator=25400000/473628672
magnification=1000;       0.00006334 pixels per DVI unit
' TeX output 1986.06.20:1039'
Postamble starts at byte 569.
maxv=43725786, maxh=30785863, maxstackdepth=3, totalpages=1
Font 33: amsl10---loaded at size 655360 DVI units
Font 23: ambx10---loaded at size 655360 DVI units
...and so on...
```

FONT SUBSTITUTION

If no `-ffontsubfile` option is given, and font substitution is required, if the current DVI file is `foo.dvi`, then the files `foo.sub`, `texfonts.sub`, and `texinputs:texfonts.sub` will be tried in order. The first two will be found on the current directory, and the last is the system default. This gives the option of document-specific, user-specific, and system-specific substitutions, and the `-f` option allows all of these to be overridden.

Font substitution lines have the form:

```
% comment
oldname.oldmag  ->      subname.submag  % comment
oldname oldmag  ->      subname submag  % comment
oldname         ->      subname         % comment
```

Examples are:

```
% These provide replacements for some LaTeX invisible fonts:
iamr10 1500     ->      amr10 1500      % comment
iamr10.1500     ->      amr10.1500      % comment
iamssb8         ->      amssb8          % comment
```

The first two forms request substitution of a particular font and magnification. The third form substitutes an entire font family; the closest available magnification to the required one will be used. Any dots in the non-comment portion will be converted to spaces, and therefore, cannot be part of a name field.

The first matching substitution will be selected, so magnification-specific substitutions should be given first, before family substitutions.

Comments are introduced by percent and continue to end-of-line, just as for TeX. One whitespace character is equivalent to any amount of whitespace. Whitespace and comments are optional.

SCREEN CONTROL

At present, DVIBIT is the only family member which supports interactive viewing of the TeX output. The following description therefore applies only to it, but the functionality should be adhered to in any new interactive device drivers.

All switches, including the page selection (`-o`) and page origin (`-x` and `-y`) switches, work normally. In order to avoid unnecessary waste of screen space, you probably will want to specify `-x0in` and `-y0in` to remove the default one-inch left and top margins. The `-q` option is probably also advisable to avoid warning messages, such as from font substitutions.

At beginning of page, a command and status menu is displayed at the top of the screen. When the end-of-page command is reached in the DVI file, or as soon as keyboard input is available, the driver will enter the end-of-page routine. Any keyboard input command is then read and acted upon; unrecognized input is discarded with a warning beep. The advantage of checking for keyboard input during the main DVI reading loop is that unwanted display can be avoided. This is valuable if you are repositioning the page, or skimming a document. The EMACS text editor uses much the same display algorithm—do nothing more to the screen if a user command will probably invalidate it anyway.

The input can select

- redisplay of the current page, possibly shifting it up, down, left, or right, to see more of it, or to restore a display trashed by an unexpected system message or transmission line error;
- continuation to the next page in the page list selected by default or by the `-o` option;

- backing up to the previous page (useful if you overshoot);
- display of an arbitrary page by typing its sequence number;
- termination of execution.

Although the menu on the top line displays only a subset of the possible commands, a number of synonyms are provided for user convenience. In particular, arrow keys in VT52 and VT100 modes are recognized, as are EMACS control-character commands to move the cursor or page display. Commands are provided for both coarse and fine adjustment of page position.

Here is the current command list. Input is immediate; no terminating carriage return is necessary. Consequently, typing error correction is supported only for the digit string command; it ends at the first non-digit typed.

D Move the display down by 1/8 of screen size.

U Move the display up by 1/8 of screen size.

L Move the display left by 1/8 of screen size.

R Move the display right by 1/8 of screen size.

d or **Ctl-N** or **down-arrow**

Move the display down by 1/64 of screen size.

u or **Ctl-P** or **up-arrow**

Move the display up by 1/64 of screen size.

l or **Ctl-B** or **left-arrow**

Move the display left by 1/64 of screen size.

r or **Ctl-F** or **right-arrow**

Move the display right by 1/64 of screen size.

. or **Ctl-L**

Redisplay current page.

@ Redisplay current page with *startup* page positioning.

CARET or **BACKSPACE**

Redisplay previous page.

nnn

nnn is a digit string; DELETE/RUBOUT and BACKSPACE keys correct typing errors in it. Move to nnn^{th} page, where document pages are numbered 1, 2, ... The TEX page numbers are displayed in the status window. This is a recursive display; if you respond at end-of-page with a *next-page* command, display will revert to the page sequence you were viewing when you first issued the *nnn* command.

SPACE or **RETURN** or **Ctl-V**

Display next page.

Q or **q** or **X** or **x**

Quit or exit. The screen will be cleared and the terminal restored to its normal font and emulation mode.

Z Zoom up one magstep (1.2 times larger) from current size.

z Zoom down one magstep (1.2 times smaller) from current size.

It is likely that some font magnifications will be unavailable for zooming, so do not be alarmed if some characters are displayed as blanks when you do this. You can use the font substitution mechanism (`-f` option above) to work around this, or you can ask your font administrator to generate the required magnifications. When font substitution happens because of an unavailable magnification, characters of an incorrect size are used with the spacing required for the font which TEX used, so output is likely to look peculiar.

To avoid exhausting the terminal's font memory, larger characters as sent as raster bitmaps each time they are used, rather than as downloaded fonts, making the screen display much slower. The size limit is large enough that this should not be necessary except at large magnifications.

SPECIALS

The TEX `\special{}` command is intended to allow the specification in a `.tex` file of a request to the DVI driver, usually for the insertion of graphical material at that point in the document. It is currently implemented only for DVIALW; other drivers will simply issue a warning message.

The TEX `\special{}` command is expected to look like one of the following:

```
\special{overlay filename}          % absolute positioning
\special{include filename}          % relative positioning
\special{insert filename}           % relative positioning
```

In the first case, the PostScript file to be included will be mapped onto the page at precisely the coordinates it specifies. In the other two cases, the upper-left corner of the bounding box will be placed at the current point. The PostScript file must then contain (usually near the start) a comment of the form

```
%%BoundingBox: llx lly urx ury
```

specifying the bounding box lower-left and upper-right coordinates in standard PostScript units (1/72 inch). Alternatively, if the comment

```
%%BoundingBox: (atend)
```

is found in the file, the last 1000 characters of the file will be searched to find a comment of the form:

```
%%BoundingBox: llx lly urx ury
```

If the PostScript file cannot be opened, or the `\special{}` command string cannot be recognized, or for relative positioning, the bounding box cannot be determined, a warning message is issued and the `\special` command is ignored.

Otherwise, the section of the PostScript file between the comment lines

```
%begin(plot)
%end(plot)
```

is copied to the output file surrounded by

```
save
300 72 div 300 72 div scale % revert to standard 1/72 inch units
% if relative positioning, then
% (xcp(in 1/72in)-llx) (ycp(in 1/72in)-ury) translate
...PostScript file contents...
restore
```

Plot files produced by <PLOT79 > have all the expected commands in them to allow their use in TEX `\special{}` commands. The two <PLOT79 > parameters which influence the size of the plot are

- the device size specified in the call to SETSZ(); it defaults to $11in$ if SETSZ is not called.
- the device space specified in the call to SETDS2() or SETDS3(); it defaults in the CORE system to the unit square, but if the <PLOT79 > framing routines are called, they will reset the device space to a horizontal or vertical frame in proportions of the local standard paper size ($1 : 8.5/11$ in the USA).

For example, if a device size of $5in$ is specified for a standard horizontal frame, the bounding box will be declared to be $5in$ wide and $(8.5/11) \times 5in = 3.8636in$ high, so a TEX manuscript requiring the plot could have the following commands at the start of a new paragraph:

```
\special{include plotfilename}
\vspace*{3.9in}
```

ENVIRONMENT VARIABLES

The behavior of the DVI translators can be influenced by definition of logical names on TOPS-20 and VAX VMS, or environment variables in Unix and PC-DOS. Compiled-in internal defaults will be provided for any of these which are not defined. They *must* be entirely in upper-case, since that is conventional on Unix systems. The names currently recognized are as follows:

DVIHELP This variable defines an alternate help string which is typed when the user makes an input error. It should direct the user to additional documentation. For example, on TOPS-20, it might be `try HELP DVI or XINFO DVI`.

FONTLIST Normally, the drivers are prepared to search first for `.pk`, then `.gf`, then `.pxl` font files. This variable can be used to change this search order, or remove one or more of the possibilities. It is expected to contain at least one of the strings `PK`, `GF`, or `PXL`, possibly separated by arbitrary punctuation and other text. This flexibility is necessary because some operating systems expect environment variables to conform to some syntax, such as that of a file name. Letter case is *not* significant. Some acceptable strings are `PXL-then-PK-then-GF`, `pk.gf`, `use-only-PXL-fonts`, and `PXL/GF/PK`.

TERM This variable is used only for DVIBIT; if it does not evaluate to either `bitgraph` or `bg`, DVIBIT will refuse to run. On Unix, this is the conventional way of defining terminal types with the TERMCAP or TERMINFO systems. This variable is ignored on VAX VMS, since the VMS C library sets it to a value which can never be `bitgraph` or `bg`.

TEXFONTS This defines the directory path for finding font files. Its value is *prepended* to the name of a TEX font to get a full file specification. A typical value in Unix for `TEXFONTS` would be `/usr/local/lib/tex/fonts/`. On TOPS-20, font *cmr10* on a 300-dot/inch device might correspond to the files `texfonts:-cmr10.300gf`, `texfonts:cmr10.300pk`, or `texfonts:cmr10.-1500pxl`.

TEXINPUTS This defines the directory path for finding files which are not in the current working directory. It is *prepended* to file names. A typical value in Unix would be `/usr/local/lib-/tex/macros/`.

IBM PC Caveats

The latest version of the drivers has been compiled with Microsoft C Version 4.0. With Version 3.0, some `.dvi` files experienced a fatal "*floating-point stack overflow*" error both with and without a floating-point coprocessor; this can only be due to code generation errors, and it disappeared with Version 4.0.

PC-DOS by default has only a small number of available open files, and this number is not adequate for the drivers with the value of five for `MAXOPEN` set in `machdefs.h`. You need to increase the limits by entering the lines

```
FILES=10
BUFFERS=10
```

in the `config.sys` file in the boot directory, then reboot the system to have the new values take effect. Larger values are of course possible, though

`FILES=20` is the limit with current versions of PC-DOS. Run-time performance can be quite sensitive to these settings, so you may wish to experiment.

If there is no `config.sys` file, or the settings of `FILES` and `BUFFERS` are too small, you will find the disk whirring madly while the driver attempts to open font files with neighboring magnifications, and then it will finally die with a message "*unable to open .err file*". Use of the `-d24` option may be useful in tracking how many files can successfully be opened.

The drivers have been loaded with the default Microsoft floating-point library; the compiler generates calls to library routines which test a flag initialized at startup time which indicates the presence or absence of the floating-point coprocessor chip. If it is available, the library routines will automatically use it. You can force the chip to be ignored by defining an arbitrary non-empty string value for the environment variable `NO87`, for example

```
set NO87=no-8087-available
```

When the DVI translator runs, the value of this variable should be typed on the screen as the first output line. On a Leading Edge PC, this typeout does not appear, for unknown reasons. On a 4.77MHz PC XT, the translators run twice (!) as slowly when `NO87` is defined.

The reason that you might need to know this is that the method employed by the library routines for detecting the presence or absence of an 8087 (or 80287) chip is not infallible, thanks to design flaws of some PC's and possibly also the Intel chips. It is conceivable that the library might think a coprocessor chip is present, when in fact it is not, and the first floating-point instruction executed would hang the machine.

FILES

The values of `texinputs:` and `texfonts:` below are system-dependent. On Unix systems, typical values are `/usr/local/lib/tex/macros/` and `/usr/-local/lib/tex/fonts/`.

`*.dvi`	TeX DeVice Independent output file
`*.dvi-err`	TeX DVIxxx translator error log
`*.err`	TeX DVIxxx translator error log when long extensions are not available

`*.dvi-xxx`	TeX DVIxxx translator output file
`*.xxx`	TeX DVIxxx translator output file when long extensions are not available
`*.sub`	DVI file-specific font substitution file
`DVISPOOL`	Environment variable (4.xBSD Unix only) defining program or shell script which sends translation of DVI file to the appropriate output spooler.
`DVISPOOL:`	Logical name (TOPS-20 only) defining program which sends translation of DVI file to the appropriate output spooler.
`texfonts.sub`	Job-wide font substitution file
`texfonts:*.*pxl`	TeX default font rasters
`texfonts:*.*gf`	TeX default font rasters
`texfonts:*.*pk`	TeX default font rasters
`texinputs:dvialw.ps`	PostScript header file containing standard macro definitions prefixed to PostScript output from DVIALW
`texinputs:texfonts.sub`	System-wide font substitution file

SEE ALSO

dvitype(1), latex(1), tex(1), tr2tex(1), *Local LaTeX Guide*, *A TeX DVI Driver Family.*

BUGS

Bugs in either the software or its documentation should be reported by electronic or postal mail to

Nelson H.F. Beebe
Center for Scientific Computation
220 South Physics Building

University of Utah
Salt Lake City, UT 84112
USA

Tel: (801) 581-5254
EMAIL: Beebe@Science.Utah.Edu (Internet)

An active electronic mailing list for news about the DVI driver family development is maintained by the author at the above net address. Send requests there if you wish to be on it.

AUTHORS

David Fuchs at Stanford University wrote DVITYPE in WEB and defined the DVI file format.

Mark Senn at Purdue University wrote a preliminary version of the BBN BitGraph driver in C, using DVITYPE as a model.

Stephan v. Bechtolsheim and Bob Brown at Purdue, Robert Wells at BBN, and Jim Schaad and Richard Furuta at the University of Washington, improved it.

Contributions for PostScript devices came from Neal Holtz at Carleton University. Simon Barnes of Schlumberger Cambridge Research Ltd., and Robin Rohlicek at BBN provided useful additions to the BBN BitGraph driver which have been generalized and incorporated in Version 2.07.

The transformation to about a dozen other device drivers, the massive code rearrangement for many new features as well as easy identification of host- and device-dependent sections, plus support for `.pk` and `.gf` compact font files, was carried out at the University of Utah by Nelson H.F. Beebe. He also wrote the documents *A TEX DVI Driver Family* and *Using* LATEX *at the University of Utah College of Science DEC-20.* The first describes all of these drivers in detail, and the second is the *Local* LATEX *Guide*.

Lon Willett at Utah adapted DVIJEP to make DVIIMP for the Imagen laser printer family.

John Sauter adapted one of the low-resolution printer drivers to produce DVIL75 for the DEC LA75 printer, and DVIL3P for the DEC LN03 Plus laser printer

Norman Naugle and colleagues at Texas A&M implemented the family on several new systems.