Output Devices

TeX Output Devices

Don Hosek

Ordinarily this column includes an exhaustive listing of names and addresses for sources of output device drivers, along with several pages of charts showing what is available. The installment in this issue is much shorter, as you see. There are two reasons for this: the information contained in the charts and the list of sources is being installed in a database and is temporarily in a relatively unprocessable state, and also, there has been no news worth mentioning since the last issue appeared.

Let me therefore take this opportunity to describe the proposed new policy for this column. Effective immediately, the complete list and charts will appear in only issue number 1 of the volume year. The remaining issues will contain only updates. Your comments on this proposal are invited; please send them to both the author and the Editor, whose addresses can be found in the address list starting on page 145.

Report from the DVI Driver Standards Committee

Tom Reid
Don Hosek

The first few months of 1989 have shown a healthy increase in the DVI driver standards discussion. For those people with network access, much has been done to provide for the dissemination of the information which has come through our hands.

The group has a LISTSERV discussion group, DRIV-L, which is the primary means of communication between its members. The list is set up so that anyone who wants to contribute ideas may do so by sending mail to DRIV-L@TAMVM1 (Bitnet) or DRIV-L@TAMVM1.TAMU.EDU (Internet). These notes will be automatically distributed to the membership of the group.

Archives of past discussions as well as papers on the topic and the current versions of standards documentation, programs, and macros are stored on the Clarkson archive in the dvi-standard group. Individuals with FTP access may obtain the files from sun.soe.clarkson.edu in the directory pub/dvi-standard. Those without FTP access may still obtain the files via e-mail using the same mechanism as is used by the LaTeX style collection, substituting dvi-standard for latex-style where appropriate. For example, to obtain the file driv-l.log8809 and a list of other files, one might send a message to archive-server@sun.soe.clarkson.edu which looks like:

```
path fschwartz%hmcvax.bitnet@clvm.clarkson.edu
get dvi-standard driv-l.log8809
index dvi-standard
```

By the TUG meeting in August, we hope to have much of the proposed standard documented and available from the archive.

Bitnet users may also obtain log files from Listserv@tamvm1 by sending the command

```
get driv-l log yymm
```

to Listserv@tamvm1 where *yy* is the last two digits of the year and *mm* is the month, expressed as a two digit number. For example, to obtain the log from September, 1988, one would send the command get driv-l log8809 to Listserv. Listserv commands should be sent either as the first line of a single-line mail message or as an interactive message (TELL on CMS, SEND on VMS).

For those without network access, the files may be obtained on a floppy disk from John Radel for his usual fees (see the article, "Free" TeX software for IBM PCs," page 202, in this issue for information on obtaining these files).

The remainder of this article outlines some preliminary results of the committee's work. Persons interested in implementing portions of this standard should check the Clarkson archive or contact Robert McGaffey, address on page 145, to obtain the most recent information on the standard.

5 \special commands

The committee has decided that the \special commands defined to date will be labeled as "experimental" and later classified as "production" after they've undergone sufficient testing to justify the reclassification. Experimental \special commands are distinguished by the prefix X_.

Further work on the precise syntactical rules for \special are under development.

5.1 Interface

One of the early decisions of the committee was that \special will be treated as a primitive command which the end user should never need to type. Instead, \special should be accessed through a high level macro set. This has the additional advantage that users at beta test sites will usually not be affected by changes to the syntax or names of \special commands. This is important since when a \special changes status from "experimental" to "production", its name will change as noted above.

The committee is developing macros for both plain TeX and LaTeX to interface with the developing standard. At the present time, only preliminary versions of these macros have been written, but a full macro set for both plain TeX and LaTeX should be be available by the publication time of this article.

5.2 Scope

\special commands have been broken down into six classes depending on what portion of the DVI output they would affect.

Global These \special commands affect the entire document. Examples of this class of \special include commands for selecting duplex printing or setting the printing orientation (portrait, landscape, etc.).

Page These \special commands affect only the page on which they are printed. Examples of this class include requests for feeding of special paper from an auxiliary tray (*e.g.,* for a cover sheet) or a single-page change in orientation.

Box These \special commands affect a block of output that is enclosed in a TeX box (and thus is, by necessity, on a single page). For example, a command to rotate a block of text would fall under this class.

Delimited These \special commands are those that affect a block of output which is not necessarily enclosed by a TeX box or contained entirely on a single page. For example, a \special command to set color would fall into this class.

Output generating These \special commands are those which generate self-contained output of some sort. For example, the X_vec \special of Section 5.3 falls into this class.

Attribute setting These \special commands modify the next output generating command which appears on the current page. If no output generating command follows an output modifying command, the command is ignored and the DVI driver program should issue a warning. An example of this class of commands would be the X_linewidth \special described in Section 5.3.

The remainder of this section will consist of additional notes on those classes of \special commands which need additional comment.

Global specials

Global specials, it has been decided, will be required to appear on the first page of the document. They will either be identified with a prefix (X_global:), delimited by a pair of \special commands (X_begin_globals ... X_end_globals) or some similar scheme.

One issue that has not been decided is whether the first page containing the global \special commands should be the first page of text or a special page on its own. Having global options specified as part of the actual first page of text minimizes the impact on existing drivers. However, it does present some problems with existing macro packages in regard to ensuring that the options are output at the right place. This problem stems from the fact that the \special commands used to convey the options to the drivers are normally placed in the body of the document. Macro packages which place headline text or entirely separate title pages prior to writing the first part of the "body" of the document will cause text to appear in the DVI file before the global options. Headline text may or may not have any impact upon the global options, but separate title pages will prevent the global options from being on the first page of the DVI file. To get around this problem, the mechanism used for passing global information will need to "cooperate" with the output routine within the macro package.

Requiring an entirely separate page at the start of the DVI file avoids the need for special interaction with the output routines of various macro packages. Instead of placing \special commands in the body of the first page, a separate macro is used which issues a separate \shipout containing the \special commands. This approach makes things easier for programs which sort or otherwise reorganize a DVI file since no culling of global options from the first text page is necessary. However, the separate page technique has an undesired effect: it produces a blank page on existing drivers which do not understand the options page.

Box specials

A box \special command, since it will always be entirely typeset on a single page, will be enclosed in a TeX box (\hbox or \vbox). In the DVI output, box structure is reflected by surrounding *push* and *pop* commands. For example, the TeX commands:

```
normal
```

```
\hbox{\special{abc} special}
text
```

generate the following DVI code:

```
    "normal"
    push
    right
    xxx "abc"
    "special"
    pop
    right
    "text"
```

A DVI driver can exploit this for a command such as X_rotate by maintaining on the DVI stack, values for items such as *rotation_angle*.

Delimited specials

The committee has not found an effective way to deal with open block \special commands yet. They will probably need to be issued in cooperation with the output routine, to insure that every delimited command is broken down into matching pairs of \special commands on each page within its bounds.

This approach is necessary for two reasons:

- If pages are reordered for any reason (*e.g.*, reverse ordering for laser printers which stack output face up) the driver should not need to have to scan the entire file to insure that it does not inadvertently break up a pair of \special commands producing a delimited command.
- Without special care being taken, a delimited command which spans pages may inadvertently affect page headers and footers which are typeset between the beginning and ending blocks.

5.3 Graphics commands

Three techniques for including graphics have been discussed. These are:

1. Make graphics entirely with TeX primitives.
2. Use METAFONT to build a graphic as a font.
3. Allow the driver to include a device-specific graphic.

Graphics by TeX

Handling graphics entirely with TeX macros and primitives which use dots or characters from a special graphics font is a technique which has been in use for some time. The LaTeX picture environment and PiCTeX work in this way with the former assembling characters from a graphic font and the latter using closely spaced dots.

In TUGboat **10**(1),[1] David F. Rogers proposed a series of TeX macros to provide plotting primitives; these macros would generally be used by TeX input generated by some graphics package. The macros which were proposed created graphics by closely spacing dots along each line in the same manner as PiCTeX.

The problem posed by creating graphics in this manner is that TeX must store all of the graphic elements in memory at once for an entire page, possibly exceeding TeX's capacity.

To calculate the memory needs, the technique for positioning each dot was specified as:

```
\kern\DX \raise\Y \hbox{\DOT}%
```

where \DX is a dimension register giving the displacement in the "x" direction from the previous point and \Y is a dimension register giving the displacement in the "y" direction from the reference point of the graph. \DOT defines the plotting symbol and \DX accounts for the width of this symbol.

In memory, TeX saves \kern\DX in a *kern* node, the raised hbox in an *hlist* node, and the plotting symbol in a *char_node*. These take two words, seven words, and one word of memory, respectively, for a total of ten words per dot. A normal-size implementation of TeX with 64 k-words of memory allows about 6000 dots to be positioned before it runs out of memory (assuming that no other macros are loaded and neglecting other text on the page). Spacing the dots at 100 per inch, this gives about 60 inches, which is not sufficient for many graphs.

To enhance the capacity of this graphics technique, we decided to use a \special to add a vector drawing capability to TeX and DVI drivers and use the \special instead of closely-spaced dots. This changes the TeX command sequence to:

```
\kern\DX \raise\Y
\hbox{\special{X_vec \number\XC
               \space \number\YC}}%
```

where \XC and \YC are dimension registers giving the components of the vector. Component values in scaled points are likely to be six-digit numbers with an additional minus sign for negative numbers. Thus, an average length for the \special string is likely to be around 18 characters. In memory, a \special is saved in a two-word *whatsit* node which points to the \special string. Thus the total memory needs, counting the *kern* and *hlist* nodes, will average 29 words per vector which allows roughly 2000 vectors. This may be sufficient for many graphs, but falls somewhat short for complex three-dimensional

[1] This article also appeared in TeXhax **89**(7).

surface plots. (One sample 3D surface plot consisted of 13,000 vectors.)

Two `\special` commands have been defined for graphics of this sort (and specialized commands for more complicated graphic elements will be defined in the future). The commands defined are:

`X_linewidth` n Specify that the following vector is to be drawn with a line width of n DVI units (scaled points for TEX). Vectors are normally 1 point in width. If no vector follows the `X_linewidth \special` on this page, the command is ignored and the DVI driver program should issue a warning.

`X_vec` Δx Δy Draw a diagonal line from the current point to the point which is offset by Δx and Δy from the current point. Δx and Δy are specified in terms of DVI units.

Graphics by METAFONT

A different approach to graphics inclusion is to use METAFONT to produce the graphic as a character of a font and position it using TEX's normal character positioning capabilities. The advantage of this technique is that the graphic is in a format which many drivers will already accept.

METAPLOT by Pat Wilcox[2] is one example of a package which takes this approach.

However, the technique has a number of drawbacks: Graphic fonts are resolution-dependent; a separate graphic font is needed for different resolution devices. METAFONT records changes in pixel values across a scan line when it builds a character. Thus, the memory needs depend upon the complexity of the graphic in addition to the size and resolution of the device. To circumvent this limitation, it is necessary to break the whole graphic into smaller pieces. It is important to ensure that the heights and widths of each piece are integral numbers of pixels to allow them to be reassembled without the alignment problems which occur for letters within words.

Including device-dependent files

With this approach, the DVI driver processes a special Graphics Description File (GDF) which, among other things, indicates the names and formats of separate graphic files in device-dependent format. A driver searches this list to find a file in a format appropriate for the device it supports. This allows a greatly simplified graphic files to be defined for previewing purposes while a detailed, higher resolution version is used when the DVI file is printed.

GDF files are processed both by TEX and by the DVI driver. TEX `\input`s the file and executes code at the start of the file. This code sets some dimension and box registers giving the size of the graphic then terminates with an `\endinput` to return control to the macro which did the `\input`. The portion of the GDF file following the `\endinput` is processed by the driver.

The driver section of the file consists of a series of keywords which identify lines that apply to a particular graphics format, rotation, etc. The driver scans these lines searching for a format which it understands. Depending on the driver and the graphics format, additional lines may have to be searched for other attributes such as rotation. Eventually, the name of the graphics file to be included will be found and the driver will incorporate it into the output file.

In "A portable graphics inclusion" (TUGboat **10**(1)), Bart Childs, Alan Stolleis, and Don Berryman suggested another scheme for using `\special` to include device-dependent graphics files.

6 Additional reference material

In addition to the works mentioned in the Editor's note at the end of our last report, the following may also be of interest:

- Guntermann, Klaus and Joachim Schrod. "High quality DVI drivers". Available from the Clarkson archive as the file `schrod-guntermann1.tex`.
- Hosek, Don. "Proposed DVI `\special` command standard". Available from the Clarkson archive as the file `hosek1.tex`.

In addition, anyone interested in implementing any portion of the developing standard should read the logs available from the Clarkson archive or `Listserv@tamvm1`.

◇ Tom Reid
Computing Services Center
Texas A&M University
College Station, TX 77843
Bitnet: X066TR@TAMVM1

◇ Don Hosek
3916 Elmwood
Stickney, IL 60402
Internet: u33297@uicvm.uic.edu
Bitnet: u33297@uicvm
Bitnet: dhosek@ymir
UUNet: dhosek@jarthur.claremont.edu
JANET: u33297%uicvm.uic.edu@uk.ac.earn-relay

[2] See page 179 of this issue of TUGboat for information about this package; see also the AmigaTEX notes of March 12, 1989 or TEXMAG **3**(3).