Output Devices

The Ideal TEX Driver

Robert W. McGaffey
Martin Marietta Energy Systems, Inc.
Oak Ridge, Tennessee

It is intended that this article encourage discussion of features of output drivers with a view towards both selecting the best features currently available and labelling those as standards to be met by driver writers. TEX has been available long enough and output drivers have matured to the point that it is possible and reasonable to decide upon a set of minimal features. (This is not yet true of TEX previewers.) We hope that there will be enough interest in setting standards to result in a session on output standardization at the TUG meeting this summer.

A driver should provide page-level control over:

1. Page orientation. In particular, both portrait mode and landscape mode must be supported. In addition, devices which support PostScript should be able to orient a page at any angle desired. Zero degrees should be equivalent to portrait mode, 90° equivalent to landscape, 45° halfway between, 180° upside down, and so forth (assuming increasing angles measured counter-clockwise).
2. Magnify a page to any magnification desired. We realize that the availability of fonts will affect some devices, but others such as the APSμ5 and the FR80 should be able to handle all magnifications within a given range. TEX does not restrict users to certain `\magnifications` and neither should drivers unless the fonts are missing. We recognize that this directly contradicts the restriction in TEX that there be only one document-level `\magnification` per TEX file and that it will cause trouble when `true` dimensions are used.
3. Modify the margins. TEX's `\output` routine creates a `box` and then calls upon `\shipout` to send the `box` to the `.dvi` file. The only standard that *has* been adopted (that we know of) concerns the margins surrounding the `\vbox` that TEX ships out. In TUGboat, Volume 5, No. 1, David Fuchs states in an article (page 22): "I would urge everyone to adopt the convention used at Stanford: all `DVI`-reading programs allow the user to specify an extra top or left margin, but these values default to 1 inch if not explicitly specified." We second the motion. The user should be able to replace these values with his own values.
4. Exceed the page boundaries without wrap-around, overstriking, or some other catastrophe. Stated positively, any material which will be printed off the page should be indicated in an error message and then not printed. Ideally, if part of a figure, a rule, or even a character would fit on the page, it would be good if that part of it were included. But we realize that doing this in some cases would cause severe loss in computer efficiency.

With respect to fonts, drivers should be able to:

1. Handle all of the `CMR` fonts. (Not the `AMR` fonts as a substitute.)
2. Allow user specifiable control over handling unavailable fonts. Users should be able to select from the following list:
   a. Replace the missing characters with white space exactly matching the metric dimensions of the characters.
   b. Replace the missing characters with black ink precisely matching the metric dimensions of the characters.
   c. Substitute a smaller or larger font if the difference in size is less than 0.5 pixels on the output device and inform the user that this was done. For example, a ten point font on a 300 dot/in laser printer uses approximately 41.511 pixels for its character heights. (10 points divided by 72.27 points/in multiplied by 300 dots/in.) If we assume that the magnification for this font was 1000, then 41.011 pixels would indicate a magnification of 987.955 and 42.011 pixels comes out to 1012.045. Thus, if this font were requested at any magnification in the range 988–1012 inclusive, we can be sure that using the font at magnification 1000 would at most produce an error of one pixel in either direction. We include this feature because once (in trying to calculate magnifications) we requested a font at magnification 1096 only to be turned away. So we shifted the numerator in our calculation and got turned away for asking for 1094. There was no simple way to get 1095 and yet that was what we needed and wanted. If this feature were implemented,

we would have been in business.* In no case should a font be replaced with another font whose topology is different (such as CMB10 for CMR10).

d. "Jiggle" the pixels of the font closest in magnification to make it smaller or larger until it is the right size. This method would actually be useful for draft copies if the final output device is capable of "real" magnification. This technique would also allow pixel-bound devices to support orientations of any angle.

3. Refer to pixel files with device-dependent names. On our VAX we have an LN03 driver which refers to the pixel files using the name TEX$PXLDIR. This causes difficulties if our QMS driver uses the same name. Yet if each driver used a name like TEXLN03_PXLDIR, there would be no problem.

With respect to font loading:

1. Drivers should download only the characters specifically needed to produce the document. This will save both device memory and CPU time for both the device and the computer which drives the device.
2. Drivers should be written to be executed in three stages. The first procedure should write a file giving the font and character information needed by the document as well as the character positions. The second procedure should download the fonts and the third must set the characters on the page. The reason for this approach is to enable a font manager to replace the second procedure. This font manager could keep track of what fonts and characters are already loaded in the device and thus enable more efficient processing.

With respect to rules:

1. All \hrules with exact heights (plus depths) must span the same number of pixels on the paper, assuming devices of the same resolution. (We have seen this not occur. The results are distracting at best, and have caused us to make some macros device-dependent.) This can be accomplished by calculating the rule thickness in pixels, rounding to the nearest number of pixels, and finally, using one pixel if the result turns out to be zero. Of course, the horizontal movement should depend upon sps and not pixels.
2. Similar arguments apply to \vrules and their widths.

With respect to outputting pages:

1. There should be at least three ways to order the output:
   a. The pages may be output in the same order they were shipped out. A starting page and number of pages should be allowed here.
   b. The pages may be output in reverse order. A starting page and number of pages should be allowed here also.
   c. The pages may be output in page number order. In this mode \count0 should be more significant than \count1 which should be more significant than \count2 etc. Thus page 4.6 should be printed before page 4.7 and page 4.5 before page 5.4. Pages numbered with roman numerals should also come out in their real order (i, ii, iii, ...). In this mode drivers should allow ranges of pages such as (-2.4,3.2) for pages ii.4 through 3.2. If there should be two or more pages with the same set of \counts and that page number is specified, all of those pages should be printed.

With respect to the \special command drivers should be able to:

1. Pull in a plot from a plot file which is written in some standard format which the output device can read. This may be a .tkf file for a Tektronix plotter, a Quic file for a QMS laser printer, etc.
2. Since the above is often impractical, every driver should call a subroutine named SPECIAL and send it the character string contained in the .dvi file when it does not understand the string itself. This would allow users of the driver to expand the capabilities of their driver without knowing or having the source code available. This subroutine would be supplied as a null routine, and of course the driver would have to be sent as object code rather than executable code so that the subroutine special could be linked to the driver.

With respect to the driver parameters a driver should:

1. Be able to read an initialization file which can be made to contain the instructions and

* Editor's note: TEX itself can generate such an "inexact" magnification: for a font at \magstep4 with a document magnification of 1200, a magnification of 2489 will be requested instead of the 2488 defined by plain for \magstep5.

parameters which are usually used by the end users. Thus, if a user commonly uses landscape mode he/she may create such a file and save the effort of typing in the landscape command every time.
2. In addition to the initialization file, a driver should be able to write the commands it receives from a user into a file which can be read by the driver and executed. Thus if a user is debugging a particular set of pages, he/she need not reenter the set of commands but may instead refer the driver to the file containing the needed set of commands.
3. Drivers should also be able to run from batch mode as well as interactive mode.

With respect to METAFONT, drivers should be able to do either of the following:

1. Have the ability to download fonts produced by METAFONT.
2. Come with a program able to convert standard METAFONT output into a form needed by the driver. This would enable users to create different sizes of the CMR fonts as well as logos, special symbols, etc.

Editor's note: Robert McGaffey has agreed to chair a committee to define standards for TEX output drivers, and will be holding an organizational meeting at the Seattle TUG meeting. Persons interested in participating in this effort should write to Robert, with a copy of the letter to Bart Childs, to assist in planning.

TEX Output Devices

Don Hosek

The device tables on the following pages list all the TEX device drivers currently known to TUG. Some of the drivers indicated in the tables are considered proprietary. Most are not on the standard distribution tapes; those drivers which are on the distribution tapes are indicated in the listing of sources below. To obtain information regarding an interface, if it is supposed to be included in a standard distribution, first try the appropriate site coordinator or distributor; otherwise request information directly from the sites listed.

The codes used in the charts are interpreted below, with a person's name given for a site when that information could be obtained and verified. If a contact's name appears in the current TUG membership list, only a phone number or network address is given. If the contact is not a current TUG member, the full address and its source are shown. When information on the drivers is available, it is included below.

Screen previewers for multi-user computers are listed in the section entitled "Screen Previewers". If a source has been listed previously under "Sources", then a reference is made to that section for names of contacts, etc.

Corrections, updates, and new information for the list are welcome; send them to Don Hosek, Bitnet DHOSEK@HMCVAX (postal address on page 99).

Sources

ACC Advanced Computer Communications, Diane Cast, 720 Santa Barbara Street, Santa Barbara, CA 93101, 805-963-9431 (DECUS, May '85)

Adelaide Adelaide University, Australia

The programs listed under Adelaide have been submitted to the standard distributions for the appropriate computers. The PostScript driver permits inclusion of PostScript files in a TEX file. The driver is described in *TUGboat, Vol. 8, No. 1.*

AMS American Mathematical Society, Barbara Beeton, 401-272-9500 Arpanet: BNB@XX.LCS.MIT.Edu

Arbor ArborText, Inc., Bruce Baker, 313-996-3566, Arpanet: bwb%arbortext@umix.cc.umich.edu

ArborText's software is proprietary and ranges in price from $150 to $3000. The drivers for PostScript printers, the HP LaserJet Plus, the QMS Lasergrafix, and Imagen printers are part of their DVILASER series. The drivers all support graphics and include other special features such as use of resident fonts or landscape printing when supported by the individual printers.

Printing on the Autologic APS-5 and μ-5 phototypesetters with DVIAPS includes support of Autologic standard library fonts and Logo processing.

A-W Addison-Wesley, Brian Skidmore, 617-944-3700, ext. 2253

Addison-Wesley supports graphics on all Macintosh software, and on Imagen, PostScript, and QMS laser printers on the IBM PC.

Bochum Ruhr Universität Bochum, Norbert Schwarz, 49 234 700-4014

Caltech1 California Institute of Technology, Glen Gribble, 818-356-6988

Caltech2 California Institute of Technology, Chuck Lane, Bitnet: CEL@CITHEX